On Learning Powers of Poisson Binomial Distributions and Graph Binomial Distributions

Dimitris Fotakis

Yahoo Research NY and National Technical University of Athens

Joint work with **Vasilis Kontonis** (NTU Athens),
Piotr Krysta (Liverpool) and **Paul Spirakis** (Liverpool and Patras)

Distribution Learning

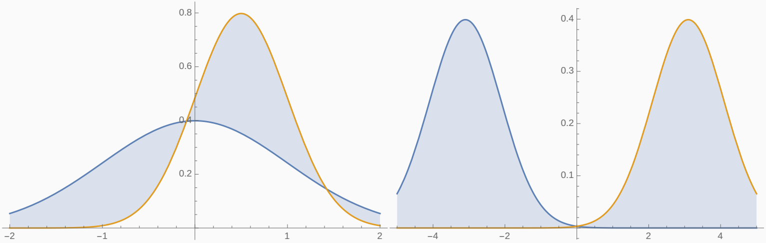
- Draw samples from unknown distribution P (e.g., # copies of NYT sold on different days).
- Output distribution Q that ε-approximates the density function of P with probability $\geqslant 1 - \delta$.
- Goal is to optimize # samples(ε, δ) (computational efficiency also desirable).

Distribution Learning

- Draw samples from unknown distribution P (e.g., # copies of NYT sold on different days).
- Output distribution Q that ε-approximates the density function of P with probability $\geqslant 1-\delta$.
- Goal is to optimize # samples(ε, δ) (computational efficiency also desirable).

Total Variation Distance

$$d_{\mathrm{tv}}(P, Q) = \frac{1}{2}\int_{\Omega} |p(x) - q(x)| \,\mathrm{d}x$$



Distribution Learning: (Small) Sample of Previous Work

- Learning any unimodal distirbution with $O(\log N/\varepsilon^3)$ samples [Birgé, 1983]
- Sparse cover for Poisson Binomial Distributions (PBDs), developed for PTAS for Nash equilibria in anonymous games [Daskalakis, Papadimitriou, 2009]
- Learning PBDs [Daskalakis, Diakonikolas, Servedio, 2011] and sums of independent integer random variables [Dask., Diakon., O'Donnell, Serv. Tan, 2013]
- Poisson multinomial distributions [Daskalakis, Kamath, Tzamos, 2015], [Dask., De, Kamath, Tzamos, 2016], [Diakonikolas, Kane, Stewart, 2016]
- Estimating the support and the entropy with $O(N/\log N)$ samples [Valiant, Valiant, 2011]

Warm-up: Learning a Binomial Distribution $\mathrm{Bin}(n, p)$

Find $\hat{p}$ s.t. $|pn - \hat{p}n| \leqslant \varepsilon\sqrt{p(1-p)n}$, or equivalently:

$$|p - \hat{p}| \leqslant \varepsilon\sqrt{\frac{p(1-p)}{n}} = \mathrm{err}(n, p, \varepsilon)$$

Then, $d_{\mathrm{tv}}(B(n, p), B(n, \hat{p})) \leqslant \varepsilon$

Warm-up: Learning a Binomial Distribution $\mathrm{Bin}(n, p)$

Find $\hat{p}$ s.t. $|pn - \hat{p}n| \leqslant \varepsilon\sqrt{p(1-p)n}$, or equivalently:

$$|p - \hat{p}| \leqslant \varepsilon\sqrt{\frac{p(1-p)}{n}} = \mathrm{err}(n, p, \varepsilon)$$

Then, $d_{\mathrm{tv}}(B(n,p), B(n,\hat{p})) \leqslant \varepsilon$

Estimating Parameter p

- Estimator: $\hat{p} = \left(\sum_{i=1}^{N} s_i\right)/(Nn)$
- If $N = O\big(\ln(1/\delta)/\varepsilon^2\big)$, Chernoff bound implies

$$\mathbb{P}[|p - \hat{p}| \leqslant \mathrm{err}(n, p, \varepsilon)] \geqslant 1 - \delta$$

Poisson Binomial Distributions (PBDs)

- Each X_i is an independent 0/1 Bernoulli trial with $\mathbb{E}[X_i] = p_i$.
- $X = \sum_{i=1}^{n} X_i$ is a PBD with probability vector $\boldsymbol{p} = (p_1, \ldots p_n)$.
- X is close to (discretized) normal distribution (assuming known mean μ and variance σ^2).
- If mean is small, X is close to Poisson distribution with $\lambda = \sum_{i=1}^{n} p_i$.

Learning Poisson Binomial Distributions

Birgé's algorithm for unimodal distributions: $O\big(\log n/\varepsilon^3\big)$ samples.

Learning Poisson Binomial Distributions

Birgé's algorithm for unimodal distributions: $O\left(\log n/\varepsilon^3\right)$ samples.

Distinguish "Heavy" and "Sparse" Cases [DaskDiakServ 11]

- Heavy case, $\sigma^2 \geqslant \Omega(1/\varepsilon^2)$:
    - Estimate variance mean $\hat{\mu}$ and $\hat{\sigma}^2$ of X using $O(\ln(1/\delta)/\varepsilon^2)$ samples.
    - (Discretized) $\mathrm{Normal}(\hat{\mu}, \hat{\sigma}^2)$ is ε-close to X.

Learning Poisson Binomial Distributions

Birgé's algorithm for unimodal distributions: $O\big(\log n/\varepsilon^3\big)$ samples.

Distinguish "Heavy" and "Sparse" Cases [DaskDiakServ 11]

- Heavy case, $\sigma^2 \geqslant \Omega(1/\varepsilon^2)$:
  - Estimate variance mean $\hat{\mu}$ and $\hat{\sigma}^2$ of X using $O(\ln(1/\delta)/\varepsilon^2)$ samples.
  - (Discretized) $\mathrm{Normal}(\hat{\mu}, \hat{\sigma}^2)$ is ε-close to X.
- Sparse case, variance is small:
  - Estimate support: using $O(\ln(1/\delta)/\varepsilon^2)$ samples, find a, b s.t. $b - a = O(1/\varepsilon)$ and $\mathbb{P}[X \in [a, b]] \geqslant 1 - \delta/4$.
  - Apply Birge's algorithm to $X_{[a,b]}$ (# samples $= O(\ln(1/\varepsilon)/\varepsilon^3)$)

Learning Poisson Binomial Distributions

Birgé's algorithm for unimodal distributions: $O\big(\log n/\varepsilon^3\big)$ samples.

Distinguish "Heavy" and "Sparse" Cases [DaskDiakServ 11]

- Heavy case, $\sigma^2 \geqslant \Omega(1/\varepsilon^2)$:
  - Estimate variance mean $\hat{\mu}$ and $\hat{\sigma}^2$ of X using $O(\ln(1/\delta)/\varepsilon^2)$ samples.
  - (Discretized) $\mathrm{Normal}(\hat{\mu}, \hat{\sigma}^2)$ is ε-close to X.
- Sparse case, variance is small:
  - Estimate support: using $O(\ln(1/\delta)/\varepsilon^2)$ samples, find a, b s.t. $b - a = O(1/\varepsilon)$ and $\mathbb{P}[X \in [a, b]] \geqslant 1 - \delta/4$.
  - Apply Birge's algorithm to $X_{[a,b]}$ (# samples $= O(\ln(1/\varepsilon)/\varepsilon^3)$)
- Using hypothesis testing, select the best approximation.

samples improved to $\tilde{O}(\ln(1/\delta)/\varepsilon^2)$ (best possible even for binomials)

Estimating $\boldsymbol{p} = (p_1, \ldots p_n)$: $\Omega(2^{1/\varepsilon})$ samples [Diak., Kane, Stew., 16]

Learning Sequences of Poisson Binomial Distributions

- $\mathcal{F} = (f_1, f_2, \ldots, f_k, \ldots)$ sequence of functions with $f_k : [0,1] \to [0,1]$ and $f_1(x) = x$.
- PBD $X = \sum_{i=1}^{n} X_i$ defined by $\boldsymbol{p} = (p_1, \ldots, p_n)$.

Learning Sequences of Poisson Binomial Distributions

- $\mathcal{F} = (f_1, f_2, \ldots, f_k, \ldots)$ sequence of functions with $f_k : [0,1] \to [0,1]$ and $f_1(x) = x$.
- PBD $X = \sum_{i=1}^{n} X_i$ defined by $\boldsymbol{p} = (p_1, \ldots, p_n)$.
- PBD sequence $X^{(k)} = \sum_{i=1}^{n} X_i^{(k)}$, where each $X_i^{(k)}$ is a 0/1 Bernoulli with $\mathbb{E}\left[X_i^{(k)}\right] = f_k(p_i)$.
- Learning algorithm selects k (possibly adaptively) and draws random sample from $X^{(k)}$.

Learning Sequences of Poisson Binomial Distributions

- $\mathcal{F} = (f_1, f_2, \ldots, f_k, \ldots)$ sequence of functions with $f_k : [0,1] \to [0,1]$ and $f_1(x) = x$.
- PBD $X = \sum_{i=1}^n X_i$ defined by $\boldsymbol{p} = (p_1, \ldots, p_n)$.
- PBD sequence $X^{(k)} = \sum_{i=1}^n X_i^{(k)}$, where each $X_i^{(k)}$ is a 0/1 Bernoulli with $\mathbb{E}\left[X_i^{(k)}\right] = f_k(p_i)$.
- Learning algorithm selects k (possibly adaptively) and draws random sample from $X^{(k)}$.
- Given $\mathcal{F}$ and sample access to $(X^{(1)}, X^{(2)}, \ldots, X^{(k)}, \ldots)$, can we learn them all with less samples than learning each $X^{(k)}$ separately?
- Simple and structured sequences, e.g., **powers** $f_k(x) = x^k$ (related to random coverage valuations and Newton identities).

Motivation: Random Coverage Valuations

- Set U of n items.
- Family $\mathcal{A} = \{A_1, \ldots, A_m\}$ random subsets of U.
- Item i is included in A_j independently with probability p_i.
- Distribution of $\#$ items included in union of k subsets,
  i.e., distribution of $|\cup_{j\in[k]} A_j|$
- Item i is included in the union with probability $1-(1-p_i)^k$
- $\#$ items in union of k sets is distributed as $n - X^{(k)}$

PBD Powers Learning Problem

- Let $X = \sum_{i=1}^{n} X_i$ be a PBD defined by $\boldsymbol{p} = (p_1, \ldots, p_n)$.
- $X^{(k)} = \sum_{i=1}^{n} X_i^{(k)}$ is the k-th PBD power of X defined by $\boldsymbol{p}^k = (p_1^k, \ldots, p_n^k)$.
- Learning algorithm that draws samples from selected powers and ε-approximates all powers of X with probability $\geqslant 1 - \delta$.

Learning the Powers of $\mathrm{Bin}(n, p)$

- Estimator $\hat{p} = \left(\sum_{i=1}^{N} s_i\right)/(Nn)$. If p small, e.g., $p \leqslant 1/\mathrm{e}$,

$$|p - \hat{p}| \leqslant \mathrm{err}(n, p, \varepsilon) \Rightarrow |p^k - \hat{p}^k| \leqslant \mathrm{err}\big(n, p^k, \varepsilon\big)$$

Intuition: error $\approx 1/\sqrt{n}$ leaves important bits of p unaffected.

Learning the Powers of $\mathrm{Bin}(n, p)$

- Estimator $\hat{p} = \left(\sum_{i=1}^{N} s_i\right)/(Nn)$. If p small, e.g., $p \leqslant 1/\mathrm{e}$,

$$|p - \hat{p}| \leqslant \mathrm{err}(n, p, \varepsilon) \Rightarrow |p^k - \hat{p}^k| \leqslant \mathrm{err}\big(n, p^k, \varepsilon\big)$$

  Intuition: error $\approx 1/\sqrt{n}$ leaves important bits of p unaffected.

- But if $p \approx 1 - \frac{1}{n}$,

$$p = 0.\underbrace{99\ldots9}_{\log n}\ \underbrace{458382}_{\text{“value”}}$$

Learning the Powers of $\mathrm{Bin}(n, p)$

- Estimator $\hat{p} = \left(\sum_{i=1}^{N} s_i\right) / (Nn)$. If p small, e.g., $p \leqslant 1/\mathrm{e}$,

$$|p - \hat{p}| \leqslant \mathrm{err}(n, p, \varepsilon) \Rightarrow |p^k - \hat{p}^k| \leqslant \mathrm{err}\big(n, p^k, \varepsilon\big)$$

  Intuition: error $\approx 1/\sqrt{n}$ leaves important bits of p unaffected.

- But if $p \approx 1 - \frac{1}{n}$,

$$p = 0.\underbrace{99\ldots9}_{\log n}\ \underbrace{458382}_{\text{“value”}}$$

  - Sampling from the first power does not reveal “right” part p, since error $\approx \sqrt{p(1-p)/n} \approx 1/n$.
  - Not good enough to approximate all binomial powers (e.g., $n = 1000,\ p = 0.9995,\ 0.9995^{1000} \approx 0.6064, 0.9997^{1000} \approx 0.7407$)

Learning the Powers of $\mathrm{Bin}(n, p)$

- Estimator $\hat{p} = \left(\sum_{i=1}^{N} s_i\right) / (Nn)$. If p small, e.g., $p \leqslant 1/\mathrm{e}$,

$$|p - \hat{p}| \leqslant \mathrm{err}(n, p, \varepsilon) \Rightarrow |p^k - \hat{p}^k| \leqslant \mathrm{err}\big(n, p^k, \varepsilon\big)$$

  Intuition: error $\approx 1/\sqrt{n}$ leaves important bits of p unaffected.

- But if $p \approx 1 - \frac{1}{n}$,

$$p = 0.\underbrace{99\ldots9}_{\log n}\ \underbrace{458382}_{\text{“value”}}$$

  - Sampling from the first power does not reveal “right” part p, since error $\approx \sqrt{p(1-p)/n} \approx 1/n$.
  - Not good enough to approximate all binomial powers (e.g., $n = 1000,\ p = 0.9995,\ 0.9995^{1000} \approx 0.6064, 0.9997^{1000} \approx 0.7407$)
  - For $\ell = \frac{1}{\ln(1/p)}$, $p^\ell = 1/\mathrm{e}$: sampling from ℓ-power reveals “right” part.

Sampling from the Right Power

Algorithm 1 Binomial Powers

1: Draw $O\big(\ln(1/\delta)/\varepsilon^2\big)$ samples from $\mathrm{Bin}(n,p)$ to obtain $\hat{p}_1$.
2: Let $\hat{\ell} \leftarrow \lceil 1/\ln(1/\hat{p}_1)\rceil$.
3: Draw $O\big(\ln(1/\delta)/\varepsilon^2\big)$ samples from $B(n,p^{\hat{\ell}})$ to get estimation $\hat{q}$ of $p^{\hat{\ell}}$.
4: Use estimation $\hat{p} = \hat{q}^{1/\hat{\ell}}$ to approximate all powers of $\mathrm{Bin}(n,p)$.

- We assume that $p \leqslant 1 - \varepsilon^2/n$. If $p \geqslant 1 - \varepsilon^2/n^d$, we need $O\big(\ln(d)\ln(1/\delta)/\varepsilon^2\big)$ samples to learn the right power ℓ.

Learning the Powers vs Parameter Learning

Question: Learning PBD Powers $\Leftrightarrow$ Estimating $\boldsymbol{p} = (p_1, \ldots, p_n)$?

- Lower bound of $\Omega(2^{1/\varepsilon})$ for parameter estimation holds if we draw samples from selected powers.
- If p_i's are well-separated, we can learn them exactly by sampling from powers.

Lower Bound on PBD Power Learning

- PBD defined by $\boldsymbol{p}$ with $n/(\ln n)^4$ groups of size $(\ln n)^4$ each. Group i has $p_i = 1 - \frac{a_i}{(\ln n)^{4i}}$, $a_i \in \{1, \ldots, \ln n\}$.
- Given $(Y^{(1)}, \ldots, Y^{(k)}, \ldots)$ that is ε-close to $(X^{(1)}, \ldots, X^{(k)}, \ldots)$, we can find (e.g., by exhaustive search) $(Z^{(1)}, \ldots, Z^{(k)}, \ldots)$ where $q_i = 1 - \frac{b_i}{(\ln n)^{4i}}$ and ε-close to $(X^{(1)}, \ldots, X^{(k)}, \ldots)$.

Lower Bound on PBD Power Learning

- PBD defined by $\boldsymbol{p}$ with $n/(\ln n)^4$ groups of size $(\ln n)^4$ each. Group i has $p_i = 1 - \frac{a_i}{(\ln n)^{4i}}$, $a_i \in \{1, \dots, \ln n\}$.
- Given $(Y^{(1)}, \dots, Y^{(k)}, \dots)$ that is ε-close to $(X^{(1)}, \dots, X^{(k)}, \dots)$, we can find (e.g., by exhaustive search) $(Z^{(1)}, \dots, Z^{(k)}, \dots)$ where $q_i = 1 - \frac{b_i}{(\ln n)^{4i}}$ and ε-close to $(X^{(1)}, \dots, X^{(k)}, \dots)$.
- For each power $k = (\ln n)^{4i-2}$,
  $\left|\mathbb{E}\left[X^{(k)}\right] - \mathbb{E}\left[Z^{(k)}\right]\right| = \Theta(|a_i - b_i|(\ln n)^2)$ and
  $\left|\mathbb{V}\left[X^{(k)}\right] + \mathbb{V}\left[Z^{(k)}\right]\right| = O((\ln n)^3)$.

Lower Bound on PBD Power Learning

- PBD defined by $\boldsymbol{p}$ with $n/(\ln n)^4$ groups of size $(\ln n)^4$ each. Group i has $p_i = 1 - \frac{a_i}{(\ln n)^{4i}}$, $a_i \in \{1, \ldots, \ln n\}$.
- Given $(Y^{(1)}, \ldots, Y^{(k)}, \ldots)$ that is ε-close to $(X^{(1)}, \ldots, X^{(k)}, \ldots)$, we can find (e.g., by exhaustive search) $(Z^{(1)}, \ldots, Z^{(k)}, \ldots)$ where $q_i = 1 - \frac{b_i}{(\ln n)^{4i}}$ and ε-close to $(X^{(1)}, \ldots, X^{(k)}, \ldots)$.
- For each power $k = (\ln n)^{4i-2}$,
  $\left|\mathbb{E}\left[X^{(k)}\right] - \mathbb{E}\left[Z^{(k)}\right]\right| = \Theta(|a_i - b_i|(\ln n)^2)$ and
  $\left|\mathbb{V}\left[X^{(k)}\right] + \mathbb{V}\left[Z^{(k)}\right]\right| = O((\ln n)^3)$.
- By sampling appropriate powers, we learn a_i exactly:
  $\Omega(n \ln \ln n/(lnn)^4)$ samples.

Parameter Learning through Newton Identities

$$\begin{pmatrix} 1 & & & & \\ \mu_1 & 2 & & & \\ \mu_2 & \mu_1 & 3 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mu_{n-1} & \mu_{n-2} & \dots & \mu_1 & n \end{pmatrix} \begin{pmatrix} c_{n-1} \\ c_{n-2} \\ c_{n-3} \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} -\mu_1 \\ -\mu_2 \\ -\mu_3 \\ \vdots \\ -\mu_n \end{pmatrix} \Leftrightarrow \boldsymbol{Mc} = -\boldsymbol{\mu}\,,$$

where $\mu_k = \sum_{i=1}^{n} p_i^k$ and c_k are the coefficients of $p(x) = \prod_{i=1}^{n}(x - p_i) = x^n + c_{n-1}x^{n-1} + \ldots + c_0$.

Parameter Learning through Newton Identities

$$\begin{pmatrix} 1 & & & & \\ \mu_1 & 2 & & & \\ \mu_2 & \mu_1 & 3 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mu_{n-1} & \mu_{n-2} & \dots & \mu_1 & n \end{pmatrix} \begin{pmatrix} c_{n-1} \\ c_{n-2} \\ c_{n-3} \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} -\mu_1 \\ -\mu_2 \\ -\mu_3 \\ \vdots \\ -\mu_n \end{pmatrix} \Leftrightarrow \boldsymbol{Mc} = -\boldsymbol{\mu}\,,$$

where $\mu_k = \sum_{i=1}^n p_i^k$ and c_k are the coefficients of $p(x) = \prod_{i=1}^n (x - p_i) = x^n + c_{n-1}x^{n-1} + \ldots + c_0$.

- Learn (approximately) μ_k's by sampling from the first n powers.
- Solve system $\boldsymbol{Mc} = -\boldsymbol{\mu}$ to obtain $\hat{\boldsymbol{c}}$: amplifies error by $O\left(n^{3/2}2^n\right)$
- Use Pan's root finding algorithm to compute $|\hat{p}_i - p_i| \leqslant \varepsilon$: requires accuracy $2^{O(-n\max\{\ln(1/\varepsilon),\ln n\})}$ in $\hat{\boldsymbol{c}}$.

Parameter Learning through Newton Identities

$$\begin{pmatrix} 1 & & & & \\ \mu_1 & 2 & & & \\ \mu_2 & \mu_1 & 3 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mu_{n-1} & \mu_{n-2} & \dots & \mu_1 & n \end{pmatrix} \begin{pmatrix} c_{n-1} \\ c_{n-2} \\ c_{n-3} \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} -\mu_1 \\ -\mu_2 \\ -\mu_3 \\ \vdots \\ -\mu_n \end{pmatrix} \Leftrightarrow \boldsymbol{Mc} = -\boldsymbol{\mu}\,,$$

where $\mu_k = \sum_{i=1}^{n} p_i^k$ and c_k are the coefficients of $p(x) = \prod_{i=1}^{n}(x - p_i) = x^n + c_{n-1}x^{n-1} + \ldots + c_0$.

- Learn (approximately) μ_k's by sampling from the first n powers.
- Solve system $\boldsymbol{Mc} = -\boldsymbol{\mu}$ to obtain $\hat{\boldsymbol{c}}$: amplifies error by $O\big(n^{3/2}2^n\big)$
- Use Pan's root finding algorithm to compute $|\hat{p}_i - p_i| \leqslant \varepsilon$: requires accuracy $2^{O(-n\max\{\ln(1/\varepsilon),\ln n\})}$ in $\hat{\boldsymbol{c}}$.
- $\#$ samples $= 2^{O(n\max\{\ln(1/\varepsilon),\ln n\})}$

Some Open Questions

- Class of PBDs where learning **powers** is easy but **parameter** learning is hard ?
- If all $p_i \leqslant 1 - \frac{\varepsilon^2}{n}$, can we learn all powers with $o(n/\varepsilon^2)$ samples?
- If $O(1)$ different values in $\boldsymbol{p}$, can we learn all powers with $O(1/\varepsilon^2)$ samples?

Graph Binomial Distributions

- Each X_i is an independent 0/1 Bernoulli trials with $\mathbb{E}[X_i] = p_i$.
- Graph $G(V, E)$ where vertex v_i is active iff $X_i = 1$.
- Given G, learn distribution of $\#$ edges in subgraph induced by active vertices, i.e., $X_G = \sum_{\{v_i, v_j\} \in E} X_i X_j$

Graph Binomial Distributions

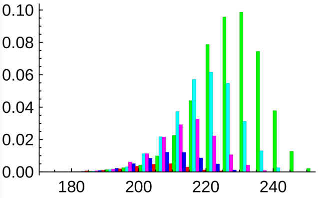
- Each X_i is an independent 0/1 Bernoulli trials with $\mathbb{E}[X_i] = p_i$.
- Graph $G(V, E)$ where vertex v_i is active iff $X_i = 1$.
- Given G, learn distribution of # edges in subgraph induced by active vertices, i.e., $X_G = \sum_{\{v_i, v_j\} \in E} X_i X_j$
- G clique: learn # active vertices k (# edges is $\frac{k(k-1)}{2}$).
- G collection of disjoint stars $K_{1,j}$, $j = 2, \ldots, \Theta(\sqrt{n})$ with $p_i = 1$ if v_i is leaf: $\Omega(\sqrt{n})$ samples are required.

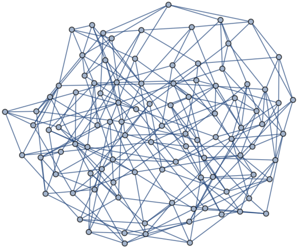
Some Observations for Single p

- If p small and G is almost regular with small degree, X is close to Poisson distribution with $\lambda = mp^2$.
- Estimating p as $\hat{p} = \sqrt{\left(\sum_{i=1}^{N} s_i\right)/(Nm)}$ gives ε-close approximation if G is almost regular, i.e., if $\sum_v \deg_v^2 = O(m^2/n)$.

Some Observations for Single p

- If p small and G is almost regular with small degree, X is close to Poisson distribution with $\lambda = mp^2$.
- Estimating p as $\hat{p} = \sqrt{\left(\sum_{i=1}^{N} s_i\right) / (Nm)}$ gives ε-close approximation if G is almost regular, i.e., if $\sum_v \deg_v^2 = O(m^2/n)$.
- Nevertheless, characterizing structure of X_G is wide open:





Thank you!