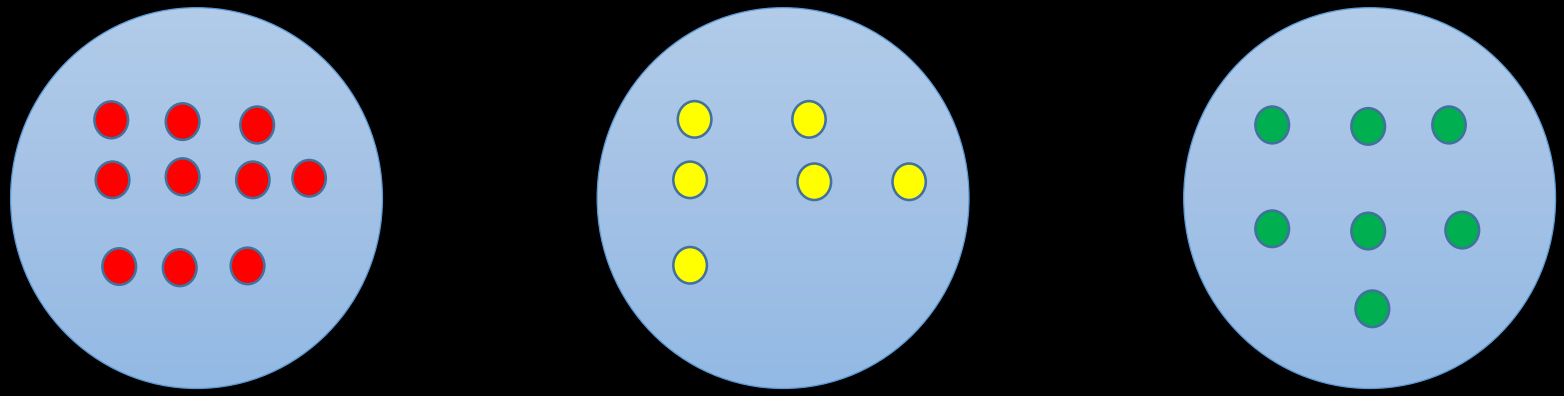
Learning Mixture of Discrete Distributions over Product Spaces

Prateek Jain
Microsoft Research, India

Joint work with Sewoong Oh (UIUC)

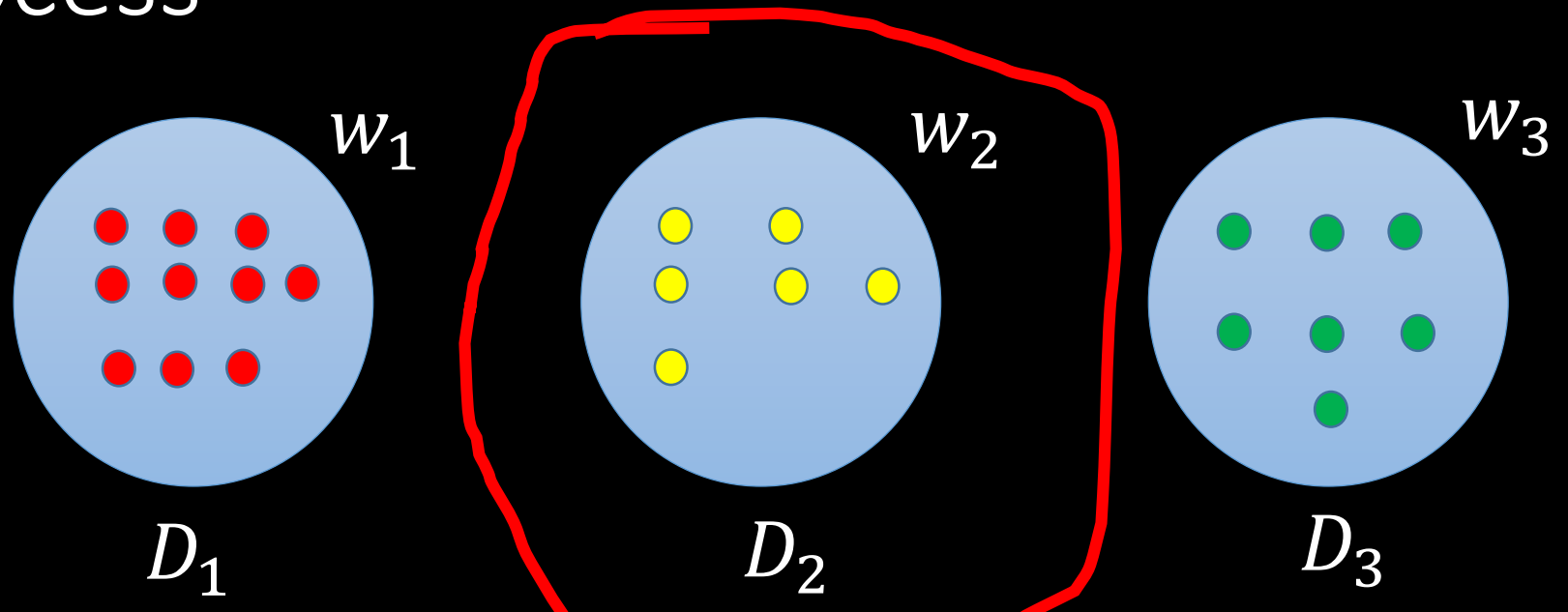
Problem Definition



- Perform clustering and learn distributions
- No labels are given
- Discrete points

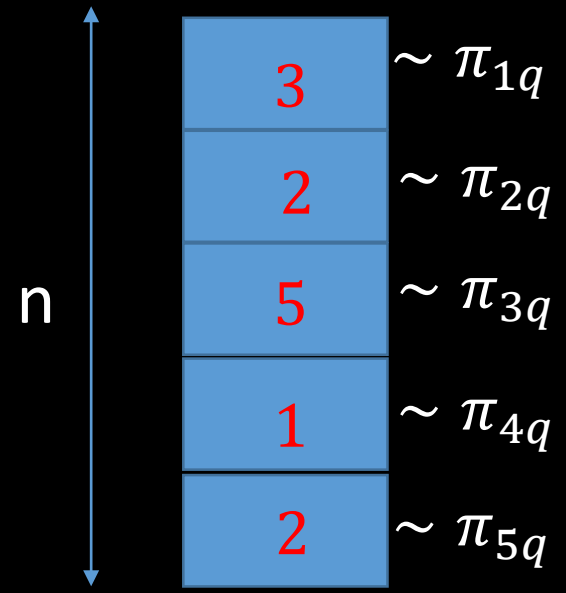
Generative Process

- k clusters

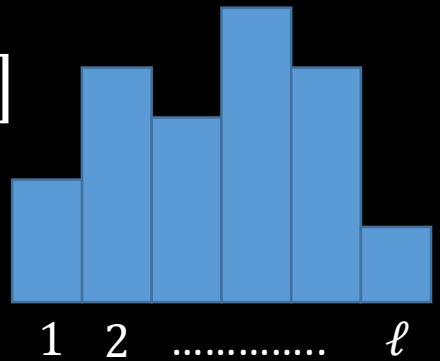


- Pick cluster q with probability w_q
- Generate a point with distribution D_q
- D_q: discrete distribution over $[1, 2, \ldots, \ell]^n$
  - Product distribution

Product Distribution



- π_{jq}: Distribution over $[1, 2, \ldots, \ell]$



Why this Problem?

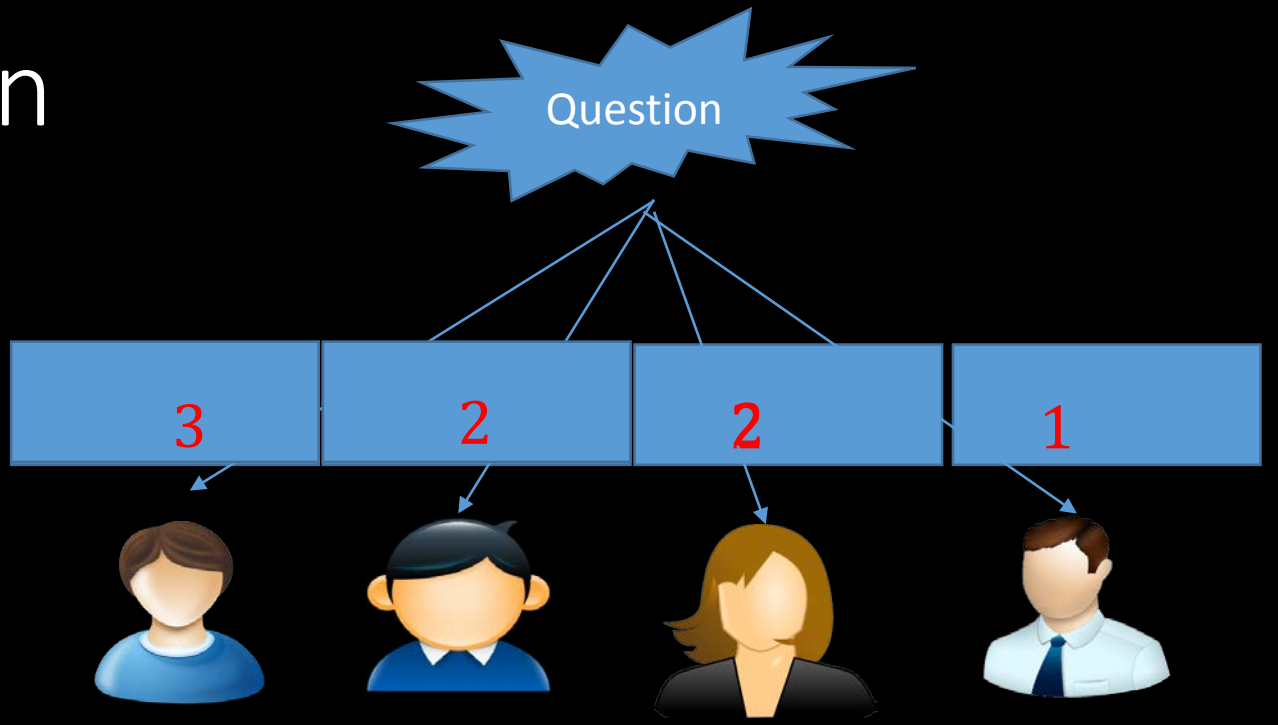
- Introduced in 1994 by
  - Kearns, Mansour, Ron, Rubinfeld, Schapire, Sellie

- Certain specific cases solved, general problem still open

- Key application:
  - Learning theory
  - Crowdsourcing
  - Learning population stratification
  - Recommendation systems

Motivating Application

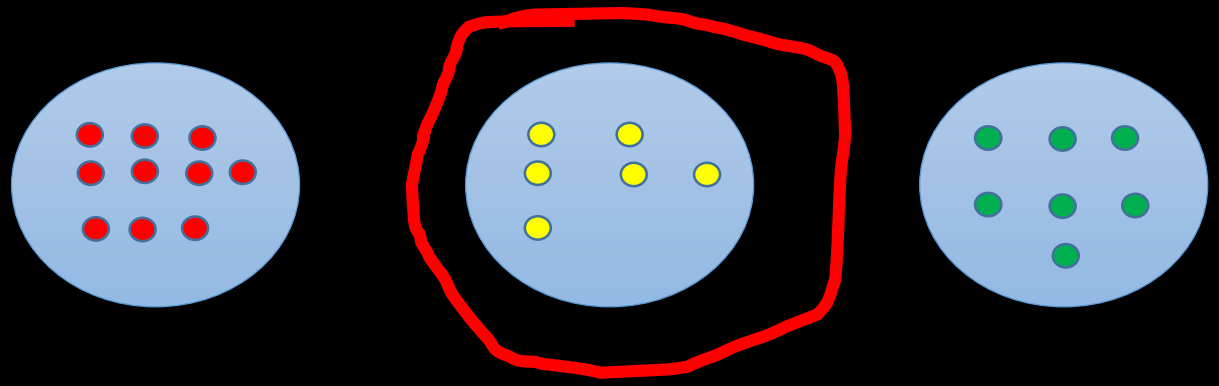
- Crowd-sourcing
  - Answers collected from workers



- Goal:
  - Assess quality of each worker for a particular question type
  - Also, figure out the correct answer

Crowdsourcing (Dawid and Skene Model)

Question Types (k):



Answers (ℓ):

3	2	2	1

Workers (n):

Existing Works

- Two categories:
  - Distributional assumptions
    - Distribution has a “spread” (means are far away) [CR08]
  - No assumptions on distribution
    - Either exponential in k (no. of clusters) or ℓ (no. of options) [FOS’08]
    - Or k, ℓ restricted to be at most a constant [FM’98, CHRZ07]

Our Results

- Efficient algorithm
  - Poly (n, k, ℓ) time/sample complexity
- Do not require any additional "spread" assumption
- Can handle a small amount of noise

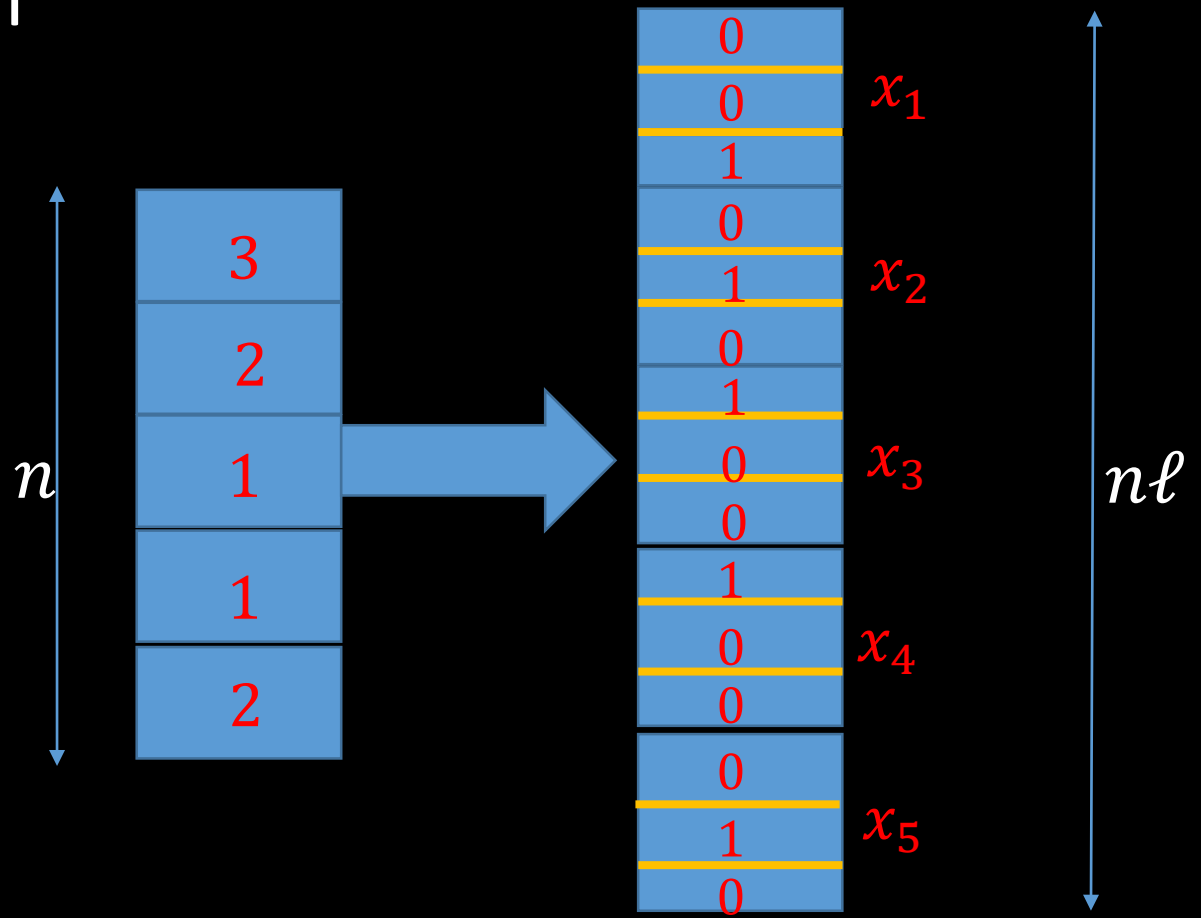
- Caveats:
  - Sample complexity depends on:
    - w_{min}
    - condition no. of distribution
  - Still don't have efficient solution to k-leaf decision tree problem
  - Assume "incoherence" of a particular matrix

Approach

- Moment matching approach
  - Anandkumar et al 2012, Hsu and Kakade 2012, ….
- Key idea:
  - Typically, 2^{nd} and 3^{rd} (sometimes 4^{th}) moment are good “signatures”
  - Form 3^{rd} order tensor → “whiten” it using 2^{nd} order moment
  - Tensor decomposition

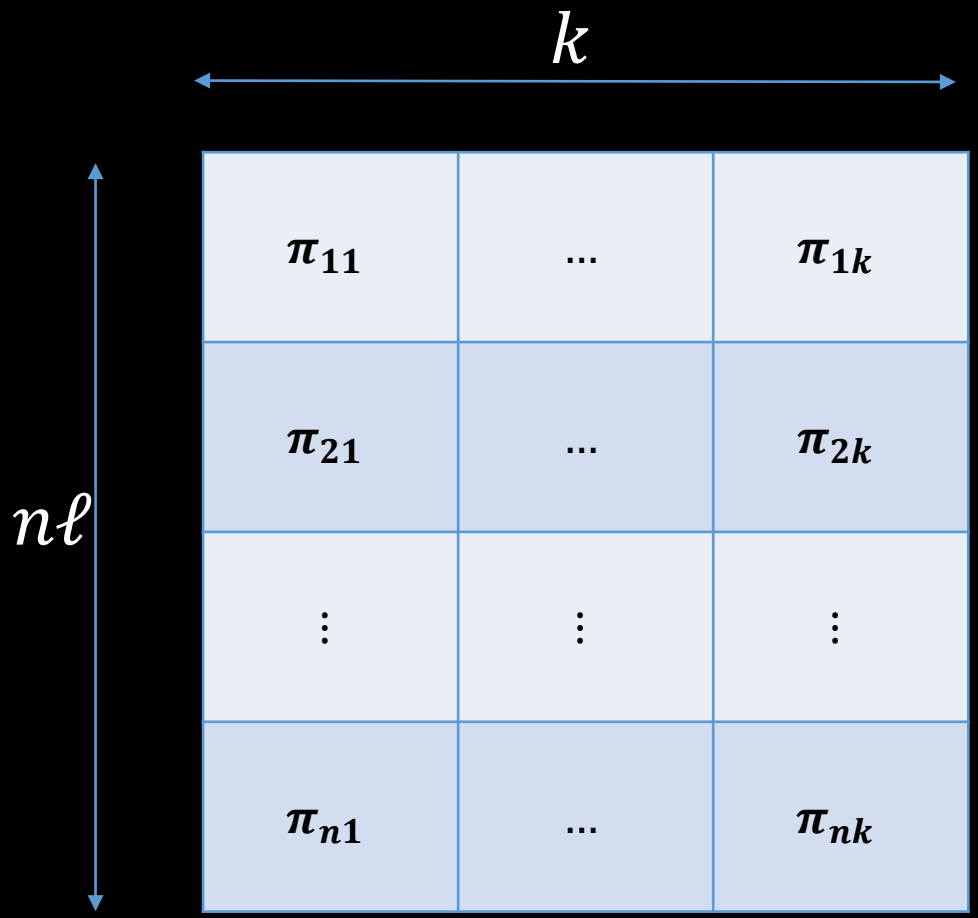
- Our approach: follows same lines
  - But need to “fill in the blanks”

Approach

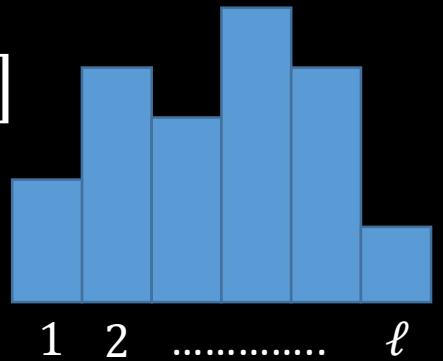


Approach

- Estimate Π



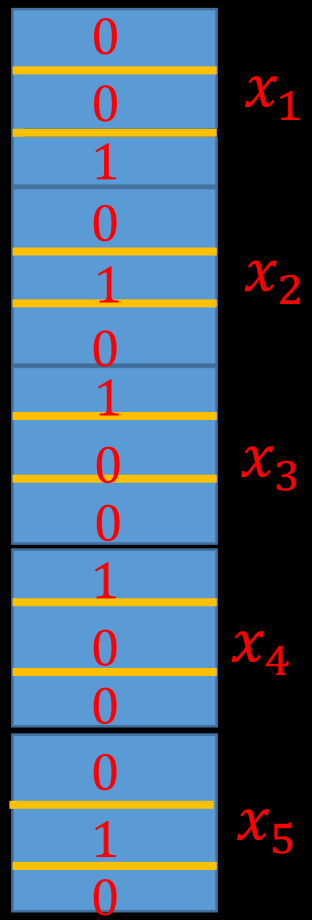
- π_{jq}: Distribution over $[1, 2, \ldots, \ell]$



Moment Matching Approach

- $\Pi = U\Sigma V^T$
- Form $M_2 = \Pi\Pi^T = U\Sigma^2 U^T$
- Form $M_3 = \Pi \otimes \Pi \otimes \Pi$
- Whiten M_3 using M_2
  - $\widehat{M}_3 = M_3[\Sigma^{-1}U^T, \Sigma^{-1}U^T, \Sigma^{-1}U^T] = V \otimes V \otimes V$
- Orthogonal tensor
  - Use tensor decomposition method by AGHKT12

Second Moment



$x_1x_1^T \neq \pi_1\pi_1^T$	$x_1x_2^T = \pi_1\pi_2^T$	$x_1x_3^T = \pi_1\pi_3^T$
$x_2x_1^T = \pi_2\pi_1^T$	$x_2x_2^T \neq \pi_2\pi_2^T$	$x_2x_3^T = \pi_2\pi_3^T$
$x_3x_1^T = \pi_3\pi_1^T$	$x_3x_2^T = \pi_3\pi_2^T$	$x_3x_3^T \neq \pi_3\pi_3^T$

Alternating Minimization

- Second moment: low-rank matrix – block diagonal matrix
- Use alternating minimization technique to fill in the “blanks”
- $M_2 = UV^T$
- $\Omega = \{Non - diagonal\ blocks\}$

$$U_{t+1} = \arg\min_{\widehat{U}} ||P_{\Omega}\left(M_2 - \widehat{U}\ V_t^T\right)||_F^2$$

$$V_{t+1} = \arg\min_{\widehat{V}} ||P_{\Omega}(M_2 - U_{t+1}\ \widehat{V}^T||_F^2$$

- Provable recovery of true M_2 using small no. of steps
  - Assume that Π is incoherent
- Similar step for estimating third moment

Finite Samples

- $\Pi = U\Sigma V^T$
- Form $M_2 = \Pi\Pi^T = U\Sigma^2 U^T + E_2$
- Form $M_3 = \Pi \otimes \Pi \otimes \Pi + \mathrm{E}_3$
- Whiten M_3 using M_2
  - $\widehat{M}_3 = M_3\left[\hat{\Sigma}^{-1}\widehat{U}^T, \hat{\Sigma}^{-1}\widehat{U}^T, \hat{\Sigma}^{-1}\widehat{U}^T\right] = V \otimes V \otimes V + E_4$
- Orthogonal tensor
  - Use robust tensor decomposition method by AGHKT12
- Combine all the steps together to give finite sample guarantees

Main Result

- No. of samples:

$$|S| \geq \frac{n^3 k^7 \kappa_\Pi^9}{w_{min}^2 \epsilon^2}$$

κ_Π: condition no. of the distribution matrix $\Pi = [\pi_1 \ \pi_2 \ldots \pi_k]$

- No. of workers (n):

$$n \geq O(k^4 \kappa_\Pi^5)$$

  - Recent results bring it down to: $n \geq O(k\ell)$

- Guarantee:

$$||\pi_q - \hat{\pi}_q|| \leq \epsilon$$

Summary

- Problem: learning mixture of discrete distributions
  - Each component is a product distribution
- Our method: moment matching method
- Our results: sample/time complexity poly(n, k, ℓ)

Future Work

- Remove dependence on condition no.
  - Will imply several strong learning theory results
  - Study the problem in PAC-model
- Study EM-based approaches for this problem

Thanks!!!

Main Result

- No. of samples:

$$|S| \geq \frac{n^2 \kappa^4}{\epsilon^2}$$

κ: condition no. of the distribution matrix $\Pi = [\pi_1 \ \pi_2 \ldots \pi_k]$

- No. of workers (n):

$$n \geq O(r^4)$$

- Guarantee:

$$||\pi_q - \hat{\pi}_q|| \leq \epsilon$$

Thanks!!!