Efficient Spectral Methods for Learning Mixture Models

Qingqing Huang

2016 February

Massachusetts Institute of Technology

LIDS Laboratory for Information & Decision Systems

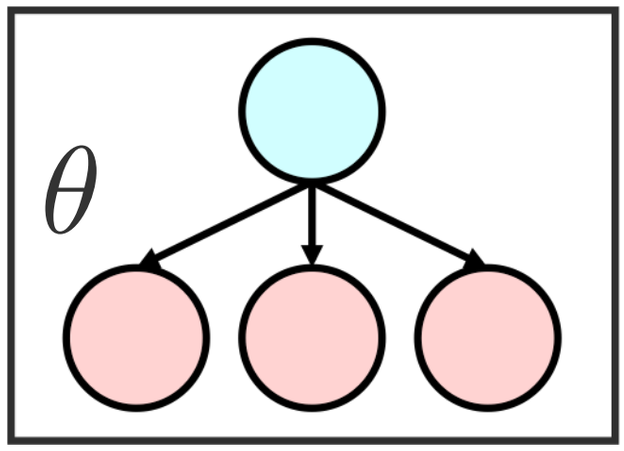
Based on joint works with Munther Dahleh, Rong Ge, Sham Kakade, Greg Valiant.

Learning

- Goal: Infer about the object of interest θ
  (Estimation, approximation, property testing, optimization of f(θ))

- Challenge: Design fast algorithm that uses as few as possible data X to achieve the targeted accuracy in θ

Learning Mixture Models



Hidden variable $H \in \{1, \ldots, K\}$

Observed variables $X = (X_1, X_2, \ldots, X_M) \in \mathcal{X}$

$\theta = (\text{mixing weight, conditional distributions})$

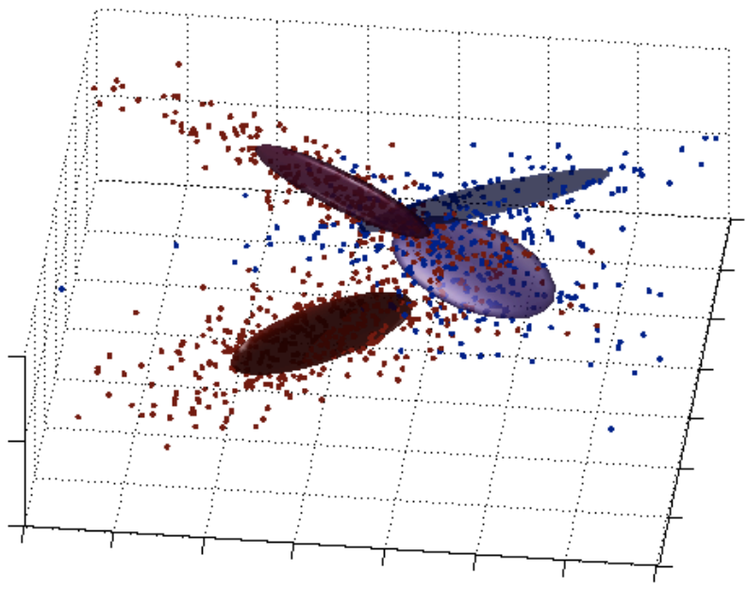
- Structured distribution of the M-dimensional random vector:

$$\Pr(X) = \sum_{k=1}^{K} \underbrace{\Pr(H = k)}_{\text{mixing weights}} \cdot \underbrace{\Pr(X|H = k)}_{\text{conditional probabilities}}$$

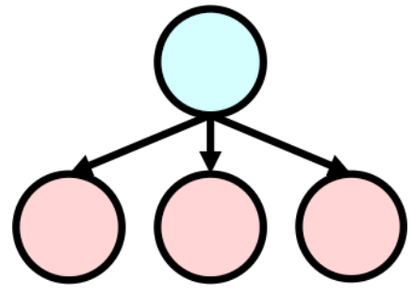
- Given N i.i.d. samples $X(1), X(2), \ldots, X(N)$ N: sample complexity

  estimate the model parameters θ such that $\left\|\widehat{\theta} - \theta\right\| \leq \epsilon$

Learning Mixture Models



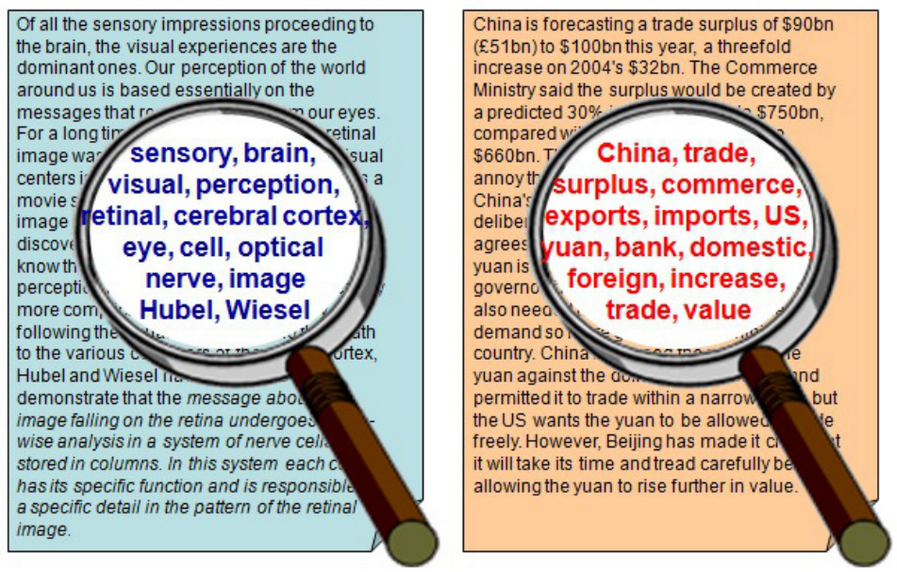
Gaussian Mixtures (GMMs)



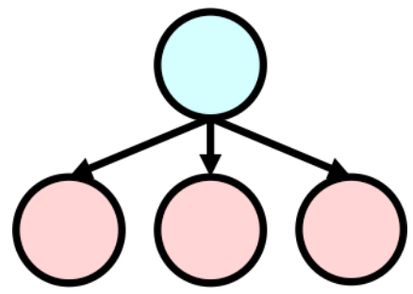
Cluster

θ

n-dim data points



Topic Models (Bag of Words)

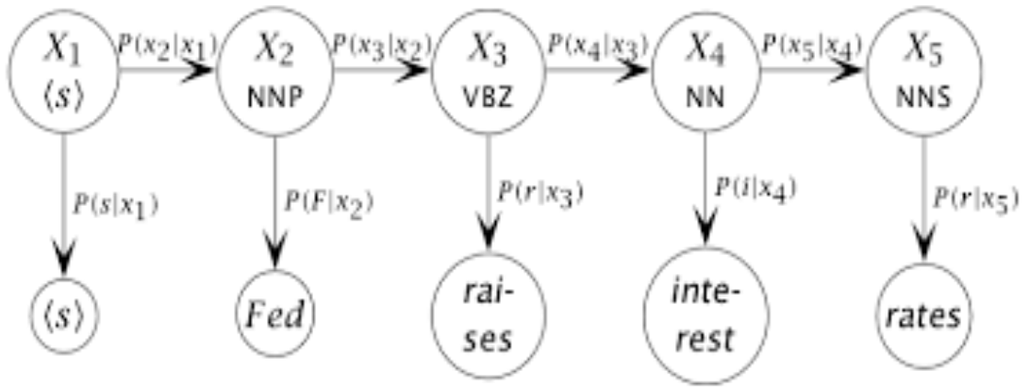


Topic

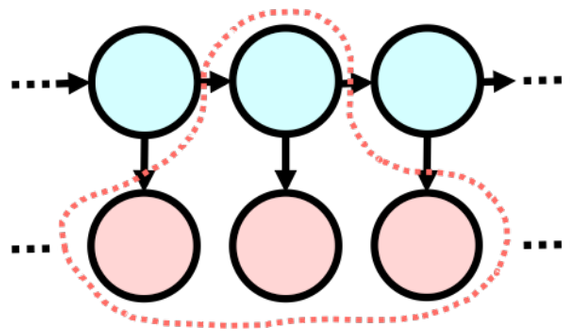
θ

n words
in each document

Learning Mixture Models



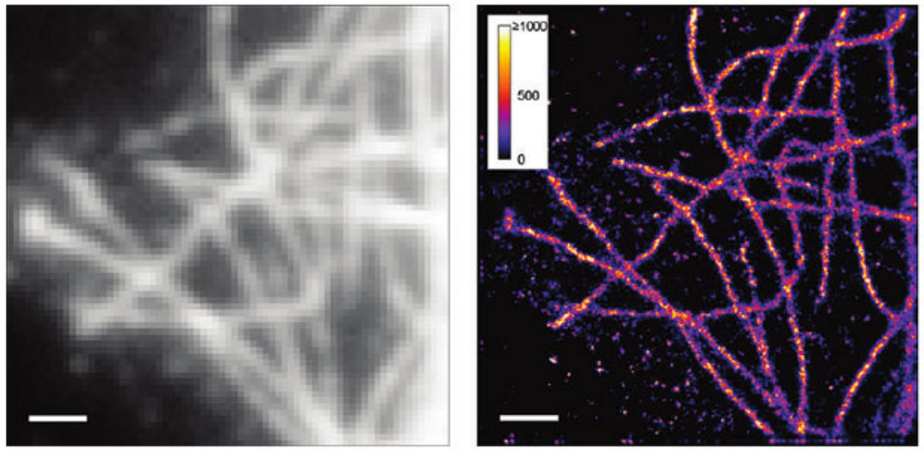
Hidden Markov Models (HMM)

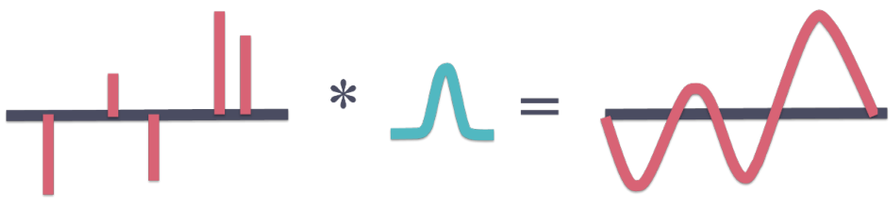


Current state

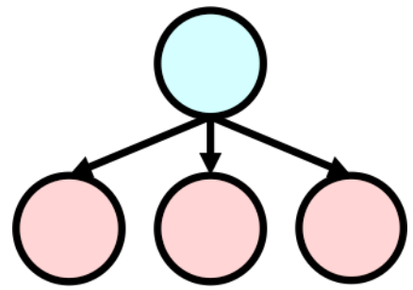
θ

Past, current, future observations





Super-Resolution Microscopy

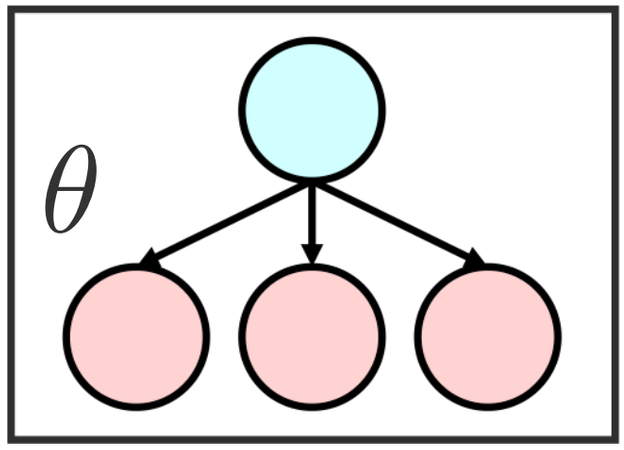


Source

θ

Complex sinusoids

Parameter Learning / Estimation



Hidden variable $H \in \{1, \ldots, K\}$

Observed variables $X = (X_1, X_2, \ldots, X_M) \in \mathcal{X}$

- Given N i.i.d. samples of X, estimate model parameters θ to target accuracy
- Maximum Likelihood Estimation (MLE)

$$\widehat{\theta}_{MLE} = \arg\max \sum_{i=1}^{N} \log\left(\Pr(X(i)|\theta)\right)$$

$$= \arg\max \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \Pr(H = k|\theta) \Pr(X(i)|H = k, \theta)\right)$$

- Impossibility results on complexity (worst case exponential lower bounds)
- MLE (is asymptotically statistical efficient) but non-convex optimization

Challenges

- Can we learn non-worst-cases with provable guarantees?

- Can we achieve optimal sample complexity in a tractable way?

Contributions / Outline

- Can we learn non-worst-cases with provable guarantees?

  Yes!

  Polynomial algorithms and analysis for GMMs, HMMs, Super-resolution

  PART 1

- Can we achieve optimal sample complexity in a tractable way?

  Yes!

  Estimation of low rank probability matrices with linear sample complexity

  PART 2

PART 1

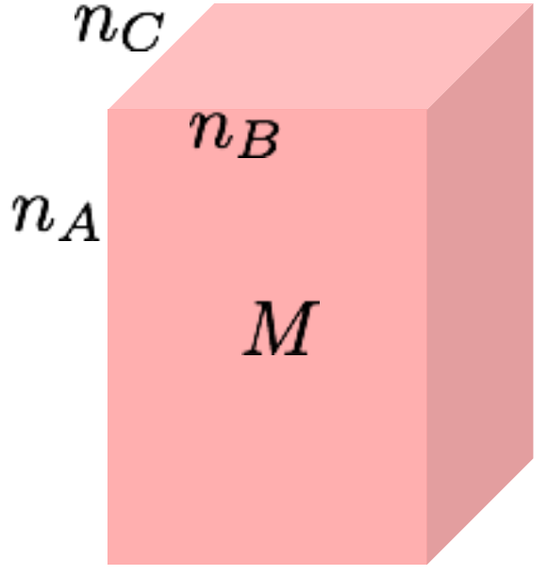
Can we learn non-worst-cases with provable guarantees?

Polynomial Algorithms for Learning **GMM, HMM and S-R**

- Propose provable **algorithms** to learn mixture models
- Go beyond worst-case **analysis** with probabilistic arguments

Spectral Algorithms and Tensor Decomposition

- Multi-way array in matlab
- 2-way tensor =matrix
- 3-way tensor:



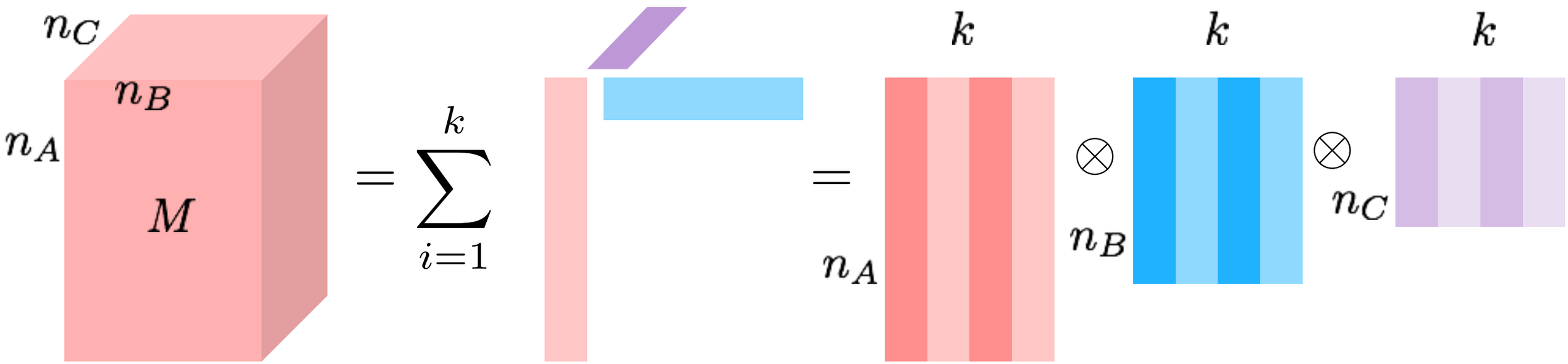
$$M_{j_1,j_2,j_3}, \quad j_1 \in [n_A], j_2 \in [n_B], j_3 \in [n_C]$$

Tensor Decomposition

- Sum of rank one tensors $[\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}]_{j_1,j_2,j_3} = a_{j_1} b_{j_2} c_{j_3}$

$$M = \sum_{i=1}^{k} A_{[:,i]} \otimes B_{[:,i]} \otimes C_{[:,i]} = A \quad \otimes \quad B \quad \otimes \quad C$$

Tensor rank: minimum number of summands in a rank decomposition



Tensor Decomposition

$$M = \sum_{i=1}^{k} A_{[:,i]} \otimes B_{[:,i]} \otimes C_{[:,i]} = A \quad \otimes \quad B \quad \otimes \quad C$$

- Necessary condition for unique tensor decomposition

  Suppose M admits rank-k decomposition $M = A \otimes B \otimes C$

  If A and B are of full rank k, and C has rank ≥ 2

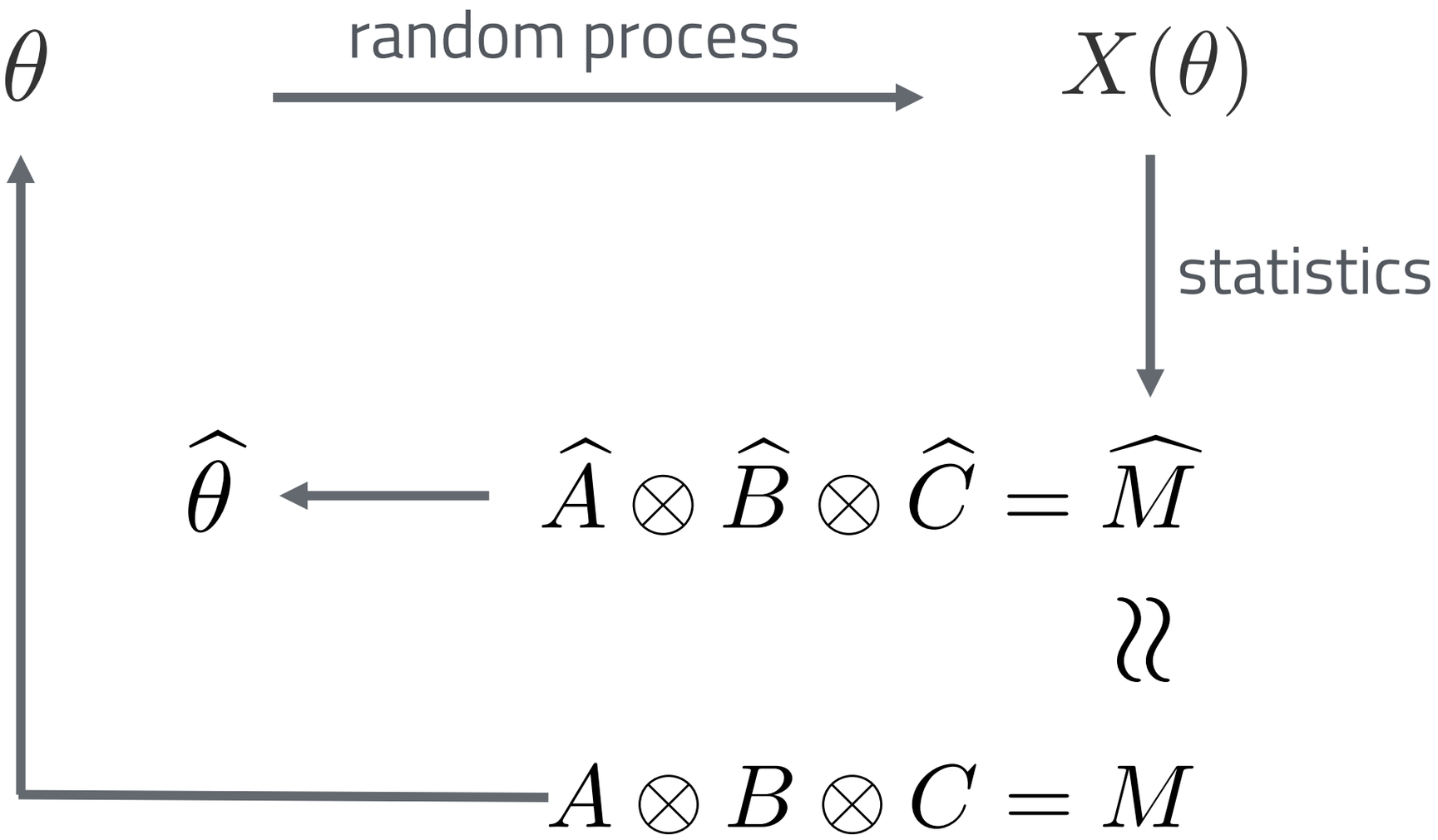
  we can decompose M to uniquely find the factors A B C

  in poly time and stability depends poly on condition number A B C

- ✓ Low rank tensor decomposition algorithm boils down to matrix SVD
- ✓ This is the only thing about tensor we need for this talk

  perhaps also the only provable and practical thing about tensor decomposition

Spectral Algorithms and Tensor Decomposition



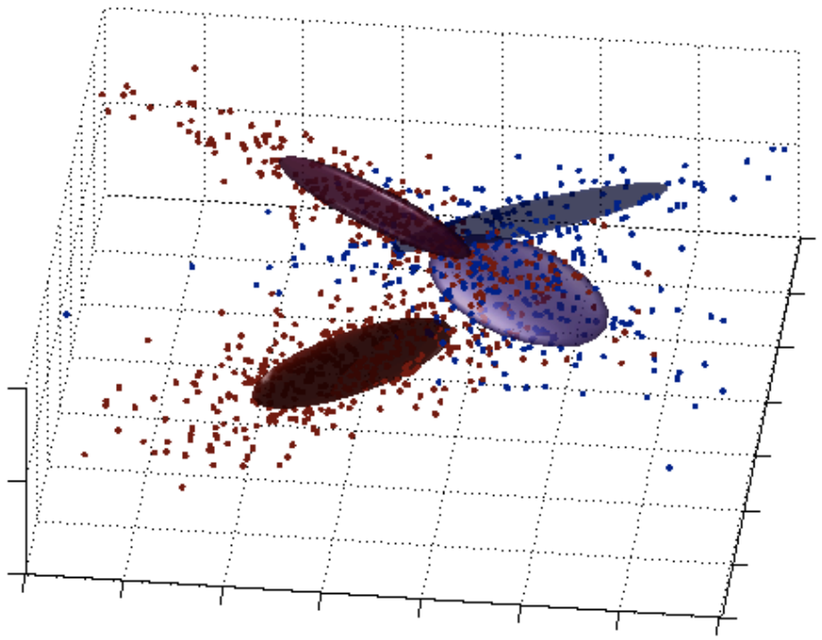
- ✓ Mixture model -> conditional independence -> Low rank of M
- ✓ Algorithm design (what statistics?) Algorithm analysis (algorithm stable?)

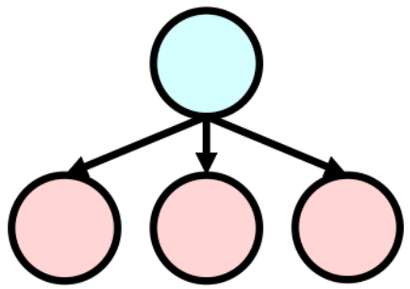
PART 1.1

Can we learn non-worst-cases with provable guarantees?

Polynomial Spectral Algorithms for Learning **Gaussian Mixtures**

1.1 Learn GMMs: Setup





Cluster $[k]$

θ

M-dim data points $x \in \mathbb{R}^n$

- n-dimensional k-mixture

  Parameters: weights w_i means $\mu^{(i)}$ covariance matrices $\Sigma^{(i)}$

  Draw samples : $x = \mathcal{N}(\mu^{(i)}, \Sigma^{(i)}), \quad i \sim w_i$

Can we learn the parameters with **poly algorithm** for every GMM ?

1.1 Learn GMMs: Hardness Result

Can we learn the parameters to accuracy ε in poly time using poly samples for **every GMM** instance ? $Poly(n, k, 1/\epsilon)$

No!

Exponential dependence in k is unavoidable in general. [Moitra&Valiant]

Can we learn the parameters with **poly algorithm** for **most** instances?

Yes!

Worst cases are not everywhere

1.1 Learn GMMs: Prior Works

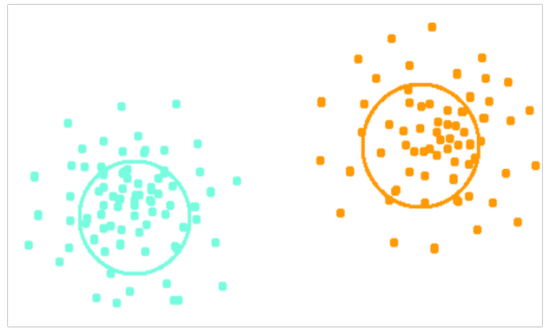
- **General case** $Poly(n, e^{O(k)^k})$

  Moment matching method [Moitra&Valiant] [Belkin&Sinha]

- **With additional assumptions on** θ $Poly(n, k)$

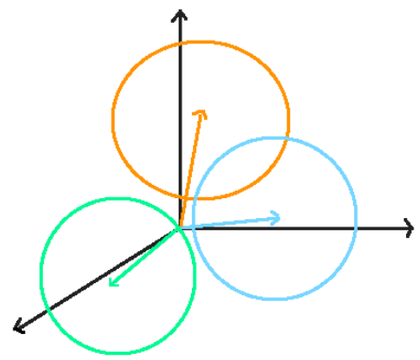
  ✓ Non-overlapping clusters

  Pair wise clustering [Dasgupta]…[Vempala&Wang]



  ✓ Spherical, n>k, independent mean vectors

  Lower order moments tensor decomposition [Hsu&Kakade]



- **Density estimation** [Chan et al]

  1-dim $Poly(k)$ Higher dim e^n

1.1 Learn GMMs

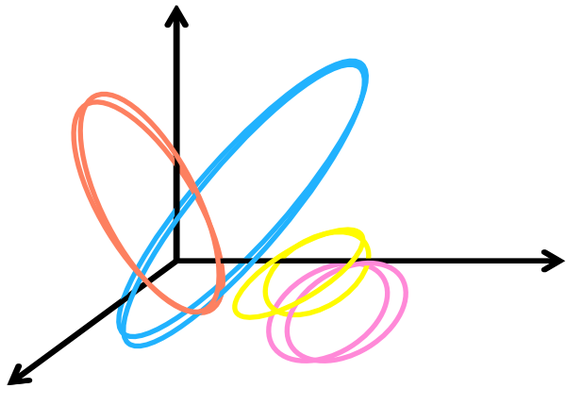
Main Theorem

- Our algorithm learns the GMM parameters up to accuracy ε
  - ✓ For high enough dimension $n = \Omega(k^2)$
  - ✓ With high probability under smoothed analysis $(1 - O(e^{-n^c}))$
  - ✓ Fully polynomial time and sample complexity $Poly(n, k, 1/\epsilon)$

1.1 Learn GMMs: Smoothed Analysis Framework

Escape from the worst cases

Given an arbitrary instance θ



Nature perturbs the parameters with a small amount (**ρ**) of noise $\widetilde{\theta}$

Observe data generated by $\widetilde{\theta}$, design and analyze algorithm for $\widetilde{\theta}$

Bridge worst case and average case algo analysis [Spielman&Teng]

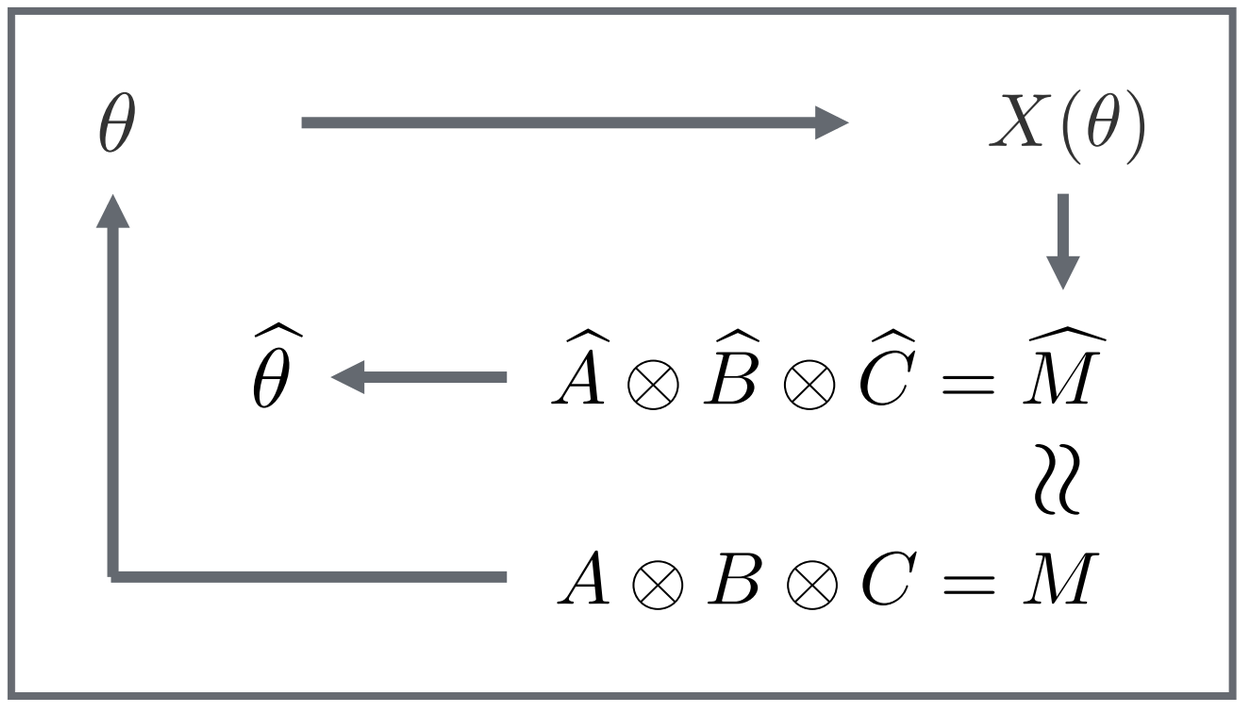
Hope: With high probability over nature's perturbation, an arbitrary instance escapes from the degenerate cases, and becomes well conditioned

Our Goal:

Given samples from smoothed GMM, learn the smoothed parameters with negligible failure probability $O(e^{-n^c})$ over nature's perturbation

1.1 Learn GMMs: Algorithmic Ideas

- Method of moments: (match 4-th and 6-th moments M_4 M_6)
  Decomposing moments tensor (tensor not low rank, but structured)



$$x = \mathcal{N}(\mu^{(i)}, \Sigma^{(i)}), \quad i \sim w_i$$

(n-dimensional k-mixture)

$$M_4 = \mathbb{E}[x^{\otimes 4}]$$
$$M_6 = \mathbb{E}[x^{\otimes 6}]$$

- Why “high dimension” & “smoothed analysis” help us to learn?
  - ✓ Enough number of moment matching constraints for identifiability

    #parameters $\Omega(kn^2)$ #6-th moments $\Omega(n^6)$
  - ✓ Enough randomness in nature perturbation for well-condition with high probability

    Gaussian matrix $X \in \mathbb{R}^{n \times m}$, with prob at least $1 - O(\epsilon^n)$ $\sigma_m(X) \geq \epsilon\sqrt{n}$.

1.1 Learn GMMs: Algorithmic Ideas

Example: Learn mixture of 0-mean Gaussians (clean moment structure)

$$x = \mathcal{N}(0, \Sigma^{(i)}), \quad i \sim w_i$$

$$M_4 = \mathcal{F}_4(X_4)$$

Structured tensor estimate from data

Structured linear projection

Desired low rank tensor!

$$M_6 = \mathcal{F}_6(X_6)$$

$$X_4 = \sum_{i=1}^{k} \Sigma^{(i)} \otimes \Sigma^{(i)},$$

$$X_6 = \sum_{i=1}^{k} \Sigma^{(i)} \otimes \Sigma^{(i)} \otimes \Sigma^{(i)}.$$

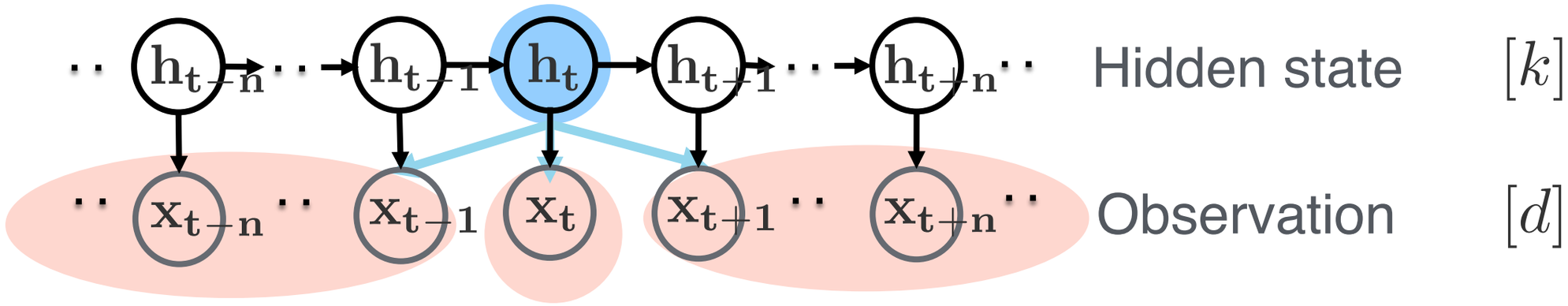
- Recover low rank tensors from their linear projections
- Exploit the structure of the linear projections particular to GMM

PART 1.2

Can we learn non-worst-cases with provable guarantees?

Polynomial Spectral Algorithms for Learning **Hidden Markov Models**

1.2 Learn HMMs: Setup



N = 2n+1 window size

$$\theta = (Q, O)$$

Transition prob: $Q \in \mathbb{R}^{k \times k}$ $\quad Q_{j,i} = \Pr(h_{t+1} = j \big| h_t = i)$

Observation prob: $O \in \mathbb{R}^{d \times k}$ $\quad O_{j,i} = \Pr(x_t = j \big| h_t = i)$

Given sequences of observation, how to recover Q, O ?

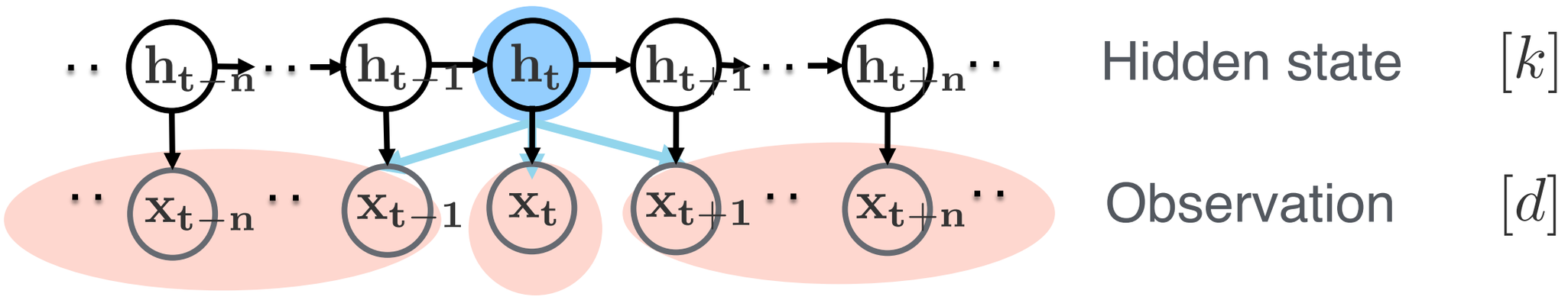
How large the window size N needs to be?

1.2 Learn HMMs: Hardness Results

- Cannot learn $\theta = (Q, O)$ in poly time, unless RP=NP
- HMM is not efficiently PAC learnable, under noisy parity assumption

Construct an instance, reduction to parity of noise [Abe,Warmuth] [Kearns]

- ✓ If number of states = 4k - 1, then required window size N = k
- ✓ Complexity is $\Omega(d^k)$



N = 2n+1 window size

1.2 Learn HMMs: Main Result

- Excluding a measure 0 set in the parameter space of $\theta = (Q, O)$ for all most all HMM instances, the required window size is

$$N = \Theta(\log_d k)$$

- Spectral alg achieves sample complexity and runtime both $poly(d, k)$

1.2 Learn HMMs: Algorithmic idea

$$M = \Pr\left((x_{n-1}, \ldots, x_{-1}), x_0, (x_1, \ldots, x_n)\right) \in \mathbb{R}^{d^n \times d^n \times d}$$

1. Estimate $\widehat{M}$, "pretend" $\widehat{M}$ is M , try low rank tensor decomp

$$M = A \otimes B \otimes C$$

Tensor decomposition condition:
A, B rank = k, C rank ≥ 1

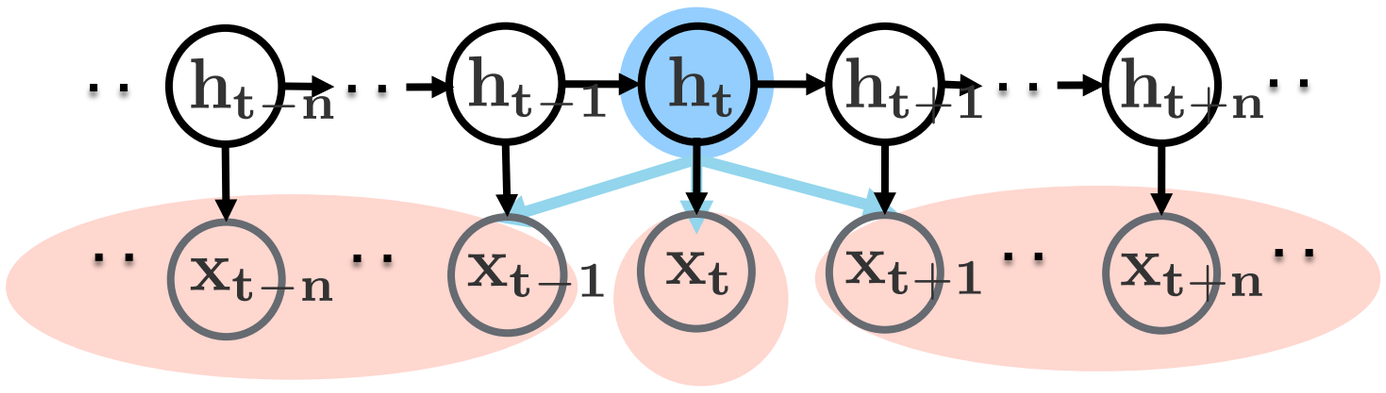
$$A = \Pr\left(x_1, x_2, \ldots, x_n \middle| h_0\right)$$
$$B = \Pr\left(x_{-1}, x_{-2}, \ldots, x_{-n} \middle| h_0\right)$$
$$C = \Pr\left(x_0, h_0\right)$$
$$A, B \in \mathbb{R}^{d^n \times k} \text{ and } C \in \mathbb{R}^{d \times k}$$

2. Extract Q, O from tensor factors A B

$$A = \underbrace{(O \odot (O \odot (O \odot \ldots (O \odot O}_{n} \underbrace{Q) \ldots)Q)Q)Q}_{n},$$
$$B = \underbrace{(O \odot (O \odot (O \odot \ldots (O \odot O}_{n} \underbrace{\tilde{Q}) \ldots)\tilde{Q})\tilde{Q})\tilde{Q}}_{n},$$
$$C = Odiag(\pi)$$

1.2 Learn HMMs: Key lemma

$$M = A \otimes B \otimes C \qquad M \in \mathbb{R}^{d^n \times d^n \times d}$$



N = 2n+1 window size

Generic analysis

- If $N = \Theta(\log_d k)$, excluding a measure 0 set in parameter space of Q, O we have A and B of full rank k, and rank C >1.
- Worst cases all in a measure 0 set!

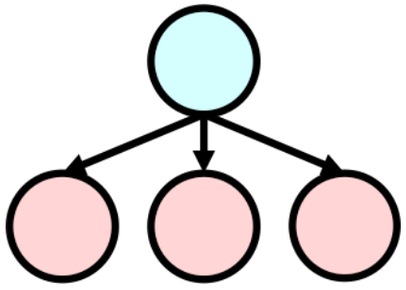
PART 1.3

Can we learn non-worst-cases with provable guarantees?

Polynomial Spectral Algorithms for **Super-Resolution**

1.3 Super-Resolution: Setup

θ —— Band-limited Fourier measurements ——▶ $X(\theta)$



Source

θ

Complex sinusoids

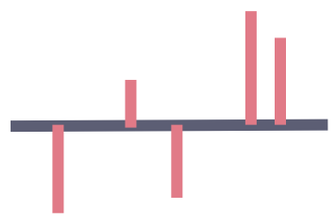
How to recover the point sources with **coarse** measurement of the signal?

- ✓ small number of measurements
- ✓ Low cutoff frequency

1.3 Super-Resolution: Problem Formulation

✧ Recover point sources (d-dimensional k-mixture)

$$x(t) = \sum_{j=1}^{k} w_j \delta_{\mu(j)}.$$



✧ using bandlimited and noise corrupted measurements. 

(Fourier measurements) $f(s) = \int_{t\in\mathbb{R}^d} e^{i\pi<t,s>} x(\mathrm{d}t) = \sum_{j=1}^{k} w_j e^{i\pi<\mu^{(j)},s>}.$

$$\|s\|_\infty \leq \text{cutoff freq}$$

(Measurement noise) $\widetilde{f}(s) = f(s) + z(s), \quad |z(s)| \leq \epsilon_z, \forall s.$

✧ Achieve target accuracy $\|\widehat{\mu}^{(j)} - \mu^{(j)}\|_2 \leq \epsilon, \forall j \in [k]$

1.3 Super-Resolution: Prior Works

$$\widetilde{f}(s) = \sum_{j=1}^{k} w_j e^{i\pi <\mu^{(j)}, s>} + z(s)$$

Minimum separation $\Delta = \min_{j \neq j'} \|\mu^{(j)} - \mu^{(j')}\|_2$

- **1-dim**
  - ✓ Take uniform measurements on the grid $s \in \{-N, \ldots, -1, 0, 1, \ldots, N\}$
  - ✓ Cut-off frequency $N = \Omega(\frac{1}{\Delta})$ [Candes, Fernandez-Granda]
  - ✓ **Hardness result** $N > \frac{C}{\Delta}$ [Moitra]
  - ✓ Compressed sensing off the grid [Tang, Bhaskar, Shah, Recht]
    use $k \log(k)$ random measurements to recover $2N$ measurements
- **Mult-dim**
  - ✓ Mult-dim grid $s \in \{-N, \ldots, -1, 0, 1, \ldots, N\}^d$
  - ✓ Algo complexity $O\left(poly(k, \frac{1}{\Delta})\right)^d$

1.3 Super-Resolution: Main Theorem

- Our algorithm achieves stable recovery
  - ✓ with number of measurements and runtime $O((k+d)^2)$
  - ✓ cutoff freq of the measurements bounded by $O(1/\Delta)$
  - ✓ with runtime $O((k+d)^3)$
- ✓ For a small error probability

	cutoff freq	measurements	runtime
SDP	$\frac{C_d}{\Delta_\infty}$	$(\frac{1}{\Delta_\infty})^d$	$poly((\frac{1}{\Delta_\infty})^d, k)$
MP	-	-	-
Ours	$\frac{\log(kd)}{\Delta}$	$(k\log(k)+d)^2$	$(k\log(k)+d)^2$

1.3 Super-Resolution: Algorithmic Idea

$$\widetilde{f}(s) = \sum_{j=1}^{k} w_j e^{i\pi <\mu^{(j)}, s>} + z(s)$$

- ✧ Tensor decomposition with measurements on **random frequencies**
- ✧ Random samples S such that F admits particular low rank decomp

$$F = V_{S'} \otimes V_{S'} \otimes (V_2 D_w),$$

(Rank-k 3-way tensor)
$d \times d \times 2$

$$V_S = \begin{bmatrix} e^{i\pi <\mu^{(1)}, s^{(1)}>} & \dots & e^{i\pi <\mu^{(k)}, s^{(1)}>} \\ e^{i\pi <\mu^{(1)}, s^{(2)}>} & \dots & e^{i\pi <\mu^{(k)}, s^{(2)}>} \\ \vdots & \dots & \vdots \\ e^{i\pi <\mu^{(1)}, s^{(m)}>} & \dots & e^{i\pi <\mu^{(k)}, s^{(m)}>} \end{bmatrix} \cdot$$

(Vandermonde Matrix with complex nodes)
$d \times k$

- ✓ Skip intermediate step of recovering all the $\Omega(N^d)$ measurements on the hyper-grid
- ✓ Prony's method (Matrix-Pencil / MUSIC / ESPRIT) as a special case of choosing S

1.3 Super-Resolution: Algorithmic Idea

$$\widetilde{f}(s) = \sum_{j=1}^{k} w_j e^{i\pi <\mu^{(j)}, s>} + z(s)$$

✧ Tensor decomposition with measurements on **random frequencies**

✧ Why **we** do not contradict the **hardness result**?

$$O((k+d)^2) \quad \textbf{VS} \quad O\left(poly(k, \frac{1}{\Delta})\right)^d$$

✓ If we design a **fixed** grid of S to take measurements $f(s)$
there always exists model instances such that the particular grid fails

✓ Let the locations of S be **random** (with structure for tensor decomp)
then for any model instance, algo works with high probability

Take-away from PART 1

- Provable algorithms to learn mixture models
  - ✓ Different algorithm for each problem
  - ✓ Underlying sparsity → Conditional independence → Low rank

- Go beyond worst-case analysis
  - ✓ Randomness in the model parameters (to escape worst case)
    - ★ Smoothed analysis for GMM
    - ★ Generic analysis for HMM
  - ✓ Randomness in the algorithm (with a large chance to succeed)
    - ★ Random sampling off the grid for S-R

Spectral methods for Learning Mixture Models

- Pros:
  - ✓ Fast runtime
    (spectral decomposition of low rank matrix/tensor)
  - ✓ Polynomial sample complexity
    (when escaping non-worst-cases)

- Cons:
  - ✓ High sample complexity
    (when compared to MLE)
  - ✓ Sensitive to model misspecification
    (rely on structural properties)

Efficiency

Robustness

PART 2

Can we achieve optimal sample complexity in a tractable way?

Estimate **low rank probability matrices** with linear sample complexity

- At the core of many spectral methods
- Our algorithm achieves statistical efficiency

2. Low rank matrix

Setup

θ ⟶ X

Probability Matrix $\mathbb{B} \in \mathbb{R}_{+}^{M \times M}$
(distribution over M^2 outcomes)

N i.i.d. samples from the distr
(freq counts over M^2 outcomes)

$\mathbb{B}$ is of rank 2: $\mathbb{B} = pp^\top + qq^\top$

.18	.14	.08	.07	.07
.14	.29	.09	.07	.10
.08	.09	.05	.40	.04
.08	.07	.04	.04	.04
.07	.10	.04	.05	.05

$\mathbb{B}$

.40	.15
.20	.40
.15	.15
.15	.10
.10	.20

p q

$M = 5$

$B = \text{Poisson}(N\mathbb{B})$

5	3	2	1	1
3	4	1	0	1
2	2	1	0	1
2	1	0	1	0
1	2	1	0	0

B

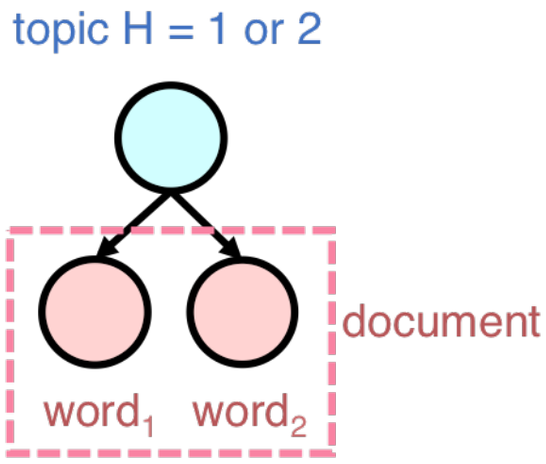
$N = 20$

Goal: find rank-2 $\widehat{B}$ such that $\|\widehat{B} - \mathbb{B}\|_1 \leq \epsilon$

N sample complexity: upper bound algorithm, lower bound

2. Low rank matrix Connection to mixture models

Topic model

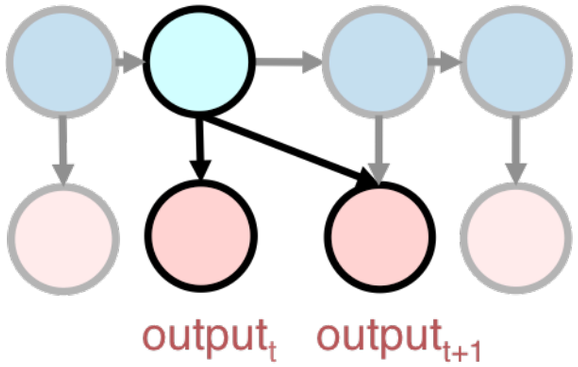


$$\Pr(\text{word}_1, \text{word}_2 | \text{topic} = T_1) = pp^\top$$

$$\Pr(\text{word}_1, \text{word}_2 | \text{topic} = T_2) = qq^\top$$

$\mathbb{B}$ joint distribution over word pairs

HMM



$$\Pr(\text{output}_1, \text{output}_2 | \text{state} = S_i) = O_i(OQ_i)^\top$$

$\mathbb{B}$ distribution of consecutive outputs

N data samples ↓ empirical counts B → find low rank $\widehat{B}$ close to $\mathbb{B}$ ↑ Extract parameters estimates

2. Low rank matrix

Attempt

θ —— generate ——▶ X

Probability Matrix $\mathbb{B} \in \mathbb{R}_{+}^{M \times M}$
(distribution over M^2 outcomes)

N i.i.d. samples from the distr
(freq counts over M^2 outcomes)

$\mathbb{B}$ is of rank 2: $\mathbb{B} = \rho\rho^\top + \Delta\Delta^\top$ $B = \text{Poisson}(N\mathbb{B})$

1st SVD $\rho : \sum \rho_i = 1$

2nd SVD $\Delta : \sum \Delta_i = 0$

◀—— estimate ——

$\widehat{\rho}_i = \sum_j \frac{1}{N} B_{i,j}$

?

2. Low rank matrix Sub-Optimal Attempt

$$\frac{1}{N}B = \frac{1}{N}\text{Poisson}(N\mathbb{B}) \to \mathbb{B}, \text{ as } N \to \infty$$

- Set $\widehat{B}$ to be the rank 2 truncated SVD of B

- To achieve estimation accuracy $\|\widehat{B} - \mathbb{B}\|_1 \leq \epsilon$
  need $N = \Omega(M^2 \log M)$

- Not sample efficient!

  MLE: $N = \Omega(M)$

- Small data in practice!

  Word distribution in language has fat tail.
  More sample documents N , larger the vocabulary size M

2. Low rank matrix Main Theorem

Matching upper and lower bound of **linear** sample complexity

- Our upper bound algorithm:
  - ✓ Rank-2 estimate $\widehat{B}$ with accuracy $\|\widehat{B} - \mathbb{B}\|_1 \leq \epsilon \quad \forall \epsilon > 0$
  - ✓ Using $N = O(M/\epsilon^2)$ number of sample counts
  - ✓ Runtime $O(M^3)$

- We prove (strong) lower bound:
  - ✓ Need a sequence of $\Omega(M)$ observations to **test** whether the sequence is i.i.d. of unif (M) or generated by a 2-state HMM

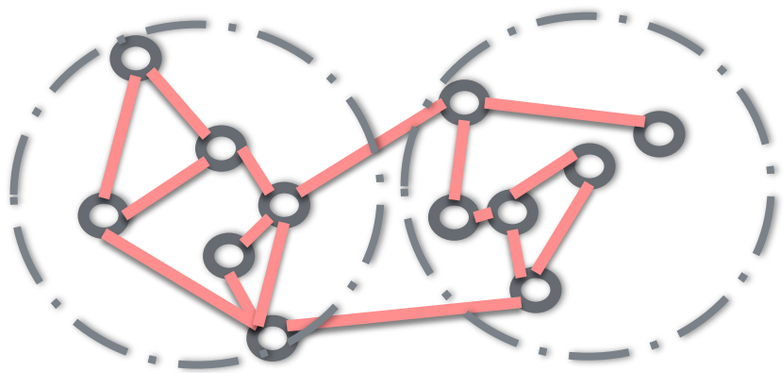
Algorithmic Idea

Capitalize the idea of community detection in sparse random network

M nodes 2 communities

Expected connection $\mathbb{B} = pp^\top + qq^\top$

Adjacency matrix $B = \text{Bernoulli}(N\mathbb{B})$



.09	.09	.09	.02	.02	.02
.09	.09	.09	.02	.02	.02
.09	.09	.09	.02	.02	.02
.02	.02	.02	.09	.09	.09
.02	.02	.02	.09	.09	.09
.02	.02	.02	.09	.09	.09

$\mathbb{B}$

.30	.03
.30	.03
.30	.03
.03	.30
.03	.30
.03	.30

p q

generate

estimate

1	1	0	0	1	0
1	1	1	0	1	1
0	1	1	0	1	0
0	0	0	0	1	1
1	1	1	1	1	1
0	1	0	0	1	1

B

2. Low rank matrix

Algorithmic Idea

Capitalize the idea of community detection in sparse random network

M nodes 2 communities

Expected connection $\mathbb{B} = pp^\top + qq^\top$

Adjacency matrix $B = \text{Bernoulli}(N\mathbb{B})$

Regularize Truncated SVD:
remove heavy row/column from B (hubs)
rank-2 SVD on remaining graph
[Le, Levina, Vershynin]

.09	.09	.09	.02	.02	.02
.09	.09	.09	.02	.02	.02
.09	.09	.09	.02	.02	.02
.02	.02	.02	.09	.09	.09
.02	.02	.02	.09	.09	.09
.02	.02	.02	.09	.09	.09

$\mathbb{B}$

.30	.03
.30	.03
.30	.03
.03	.30
.03	.30
.03	.30

p q

generate

estimate

1	1	0	0	1	0
1	1	1	0	1	1
0	1	1	0	1	0
0	0	0	0	1	1
1	1	1	1	1	1
0	1	0	0	1	1

B

2. Low rank matrix Algorithmic Idea 1, **Binning**

$M \times M$

Probability matrix $\mathbb{B} = \rho\rho^\top + \Delta\Delta^\top$

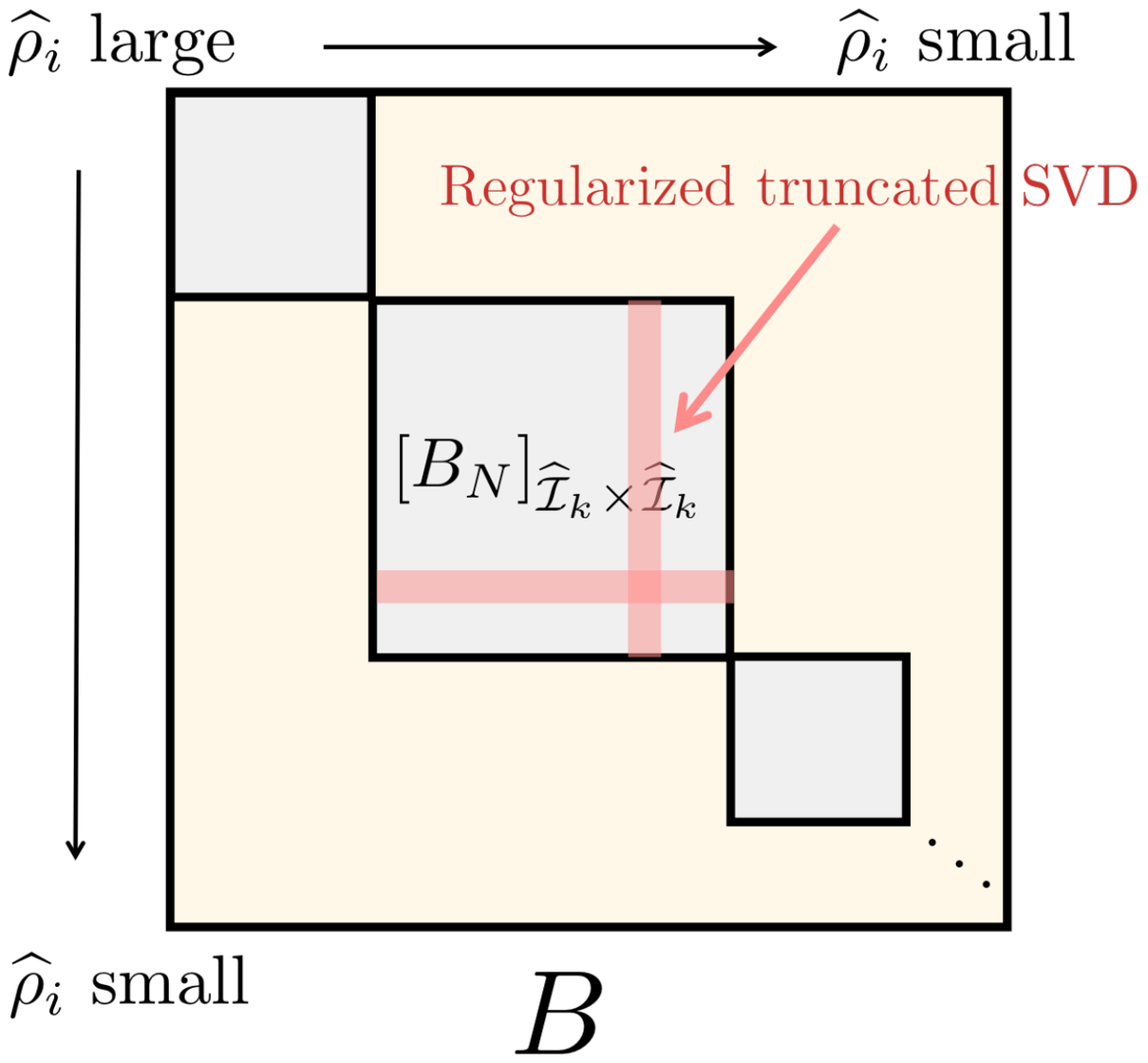
Sample counts $B = \text{Poisson}(N\mathbb{B})$

Phase 1

1. Estimate non-uniform marginal $\widehat{\rho}$
2. Bin M words according to $\widehat{\rho}_i$
3. Regularized t-SVD in each bin $\times$ bin block of B to estimate

Need to deal with spillover!

Piece together estimates over bins!



2. Low rank matrix Algorithmic Idea 2, **Refinement**

$M \times M$

Probability matrix $\mathbb{B} = \rho\rho^\top + \Delta\Delta^\top$

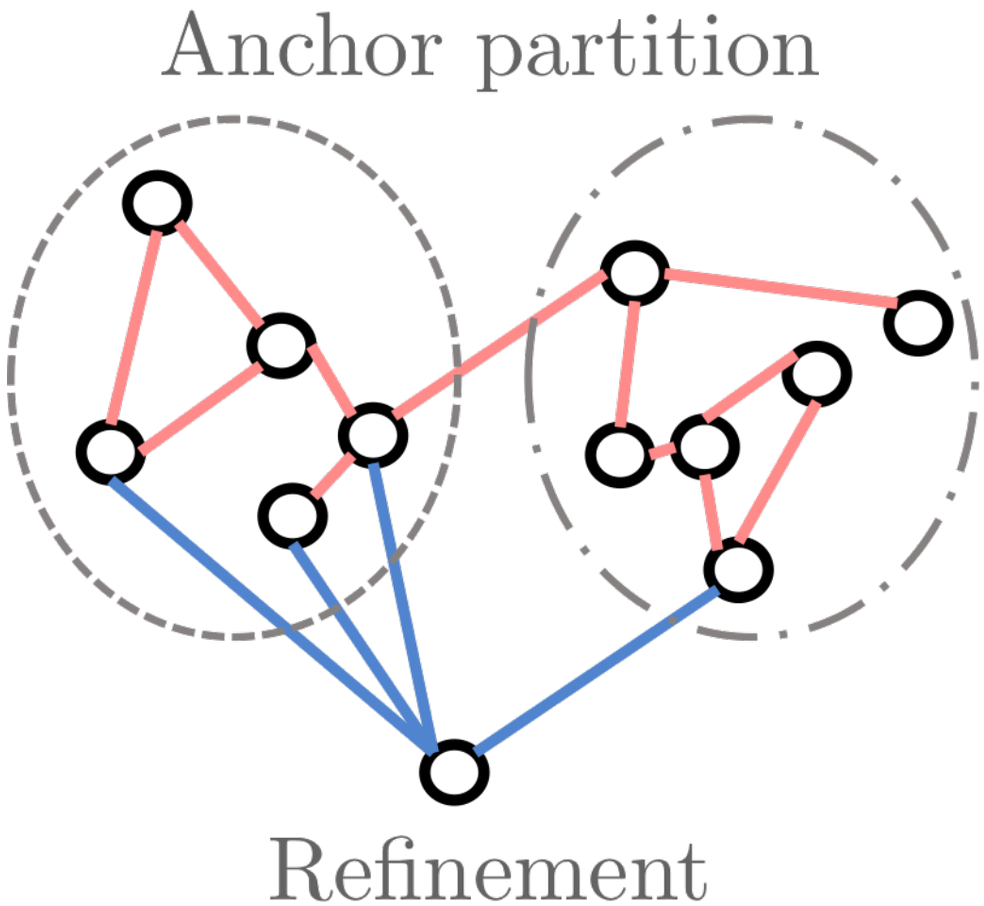
Sample counts $B = \mathrm{Poisson}(N\mathbb{B})$

Phase 2

1. Get rough estimate $\widehat{\rho}, \Delta$ for many words from Phase 1

2. Make this global info to refine $\widehat{\rho}_i, \widehat{\Delta}_i$ the estimate for each word (just linear regression)

3. Achieve sample complexity $N = O(M/\epsilon^2)$



Take-away from PART 2

- Spectral method is not solving the optimization of MLE

  Need careful algorithm design to improve statistical efficiency

- We identify a problem that lies at the core of many spectral algos

  Our result (upper / lower bound) leads to improved algos

- Coming soon: *estimation / approximation / property testing of structured probability distribution*

Conclusion and Future works

- Spectral methods for Learning Mixture Models
  - ✓ Pros: Fast runtime
    Poly sample complexity — **PART 1**
  - ✓ Cons: High sample complexity — Efficiency — **PART 2**
    Sensitive to model misspecification — Robustness

- **Future work:**
  - ✓ **Dynamics:**
    Extend the analysis techniques and algorithmic ideas to learning of random processes, with streaming data, iterative algorithms…
  - ✓ **Robustness:**
    Agnostic learning, generalization error analysis…

References

- “Learning Mixture of Gaussians in High dimensions” Rong Ge, **H**, Sham Kakade (STOC 2015)

- “Super-Resolution off the Grid” **H**, Sham Kakade (NIPS 2015)

- “Minimal Realization Problems for Hidden Markov Models” **H**, Rong Ge, Sham Kakade, and Munther Dahleh (IEEE Transactions on Signal Processing, 2016)

- “Recovering Structured Probability Matrices “ **H**, Sham Kakade, Wenhao Kong, Gregory Valiant, (submitted to STOC 2016)

Thank you !

Question?