Efficient Algorithms for Learning Mixture Models

Thesis defense 2016 May 27

Qingqing Huang

Thesis Committee:
Munther Dahleh
Sham Kakade
Pablo Parrilo

Statistical Learning

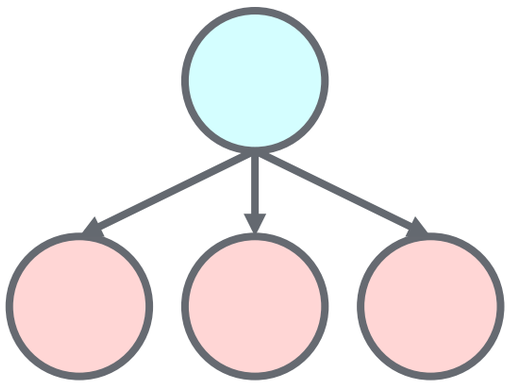
θ — Data generation → $X(\theta)$ — Learning ←

- Given data, infer about the underlying rule θ
  (Estimation, approximation, property testing, optimization of f(θ))

- Exploit our prior for structure of the underlying θ
  to design fast algorithm that uses as few as possible data X
  to achieve the target accuracy in learning θ

Computation Complexity ↓ Sample Complexity ↓ $\uparrow \dim(\theta)$

Mixture Models

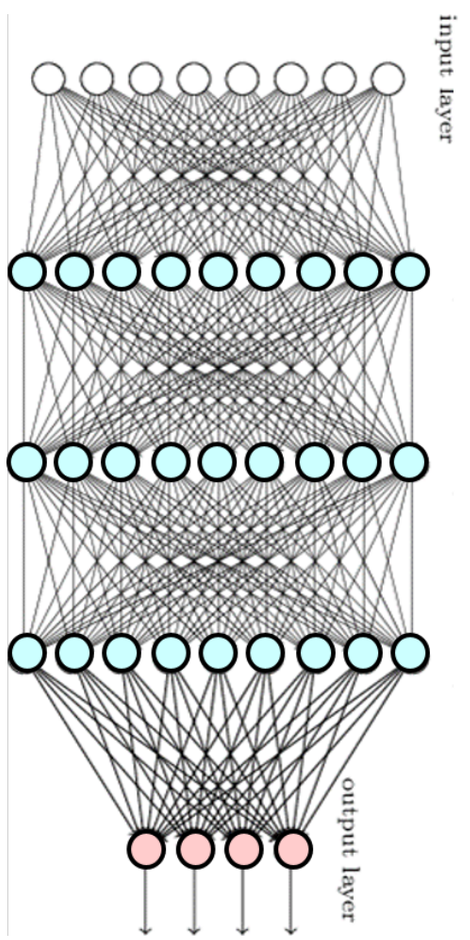


Hidden

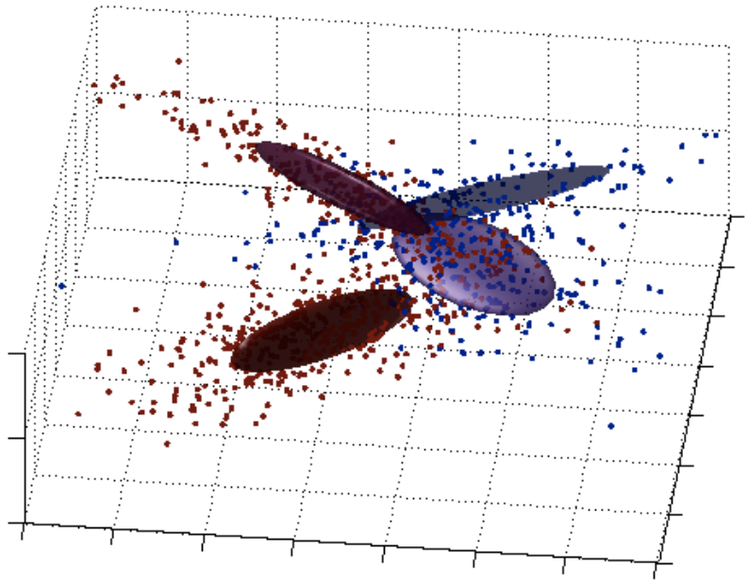
Observed

θ : a "Shallow" network

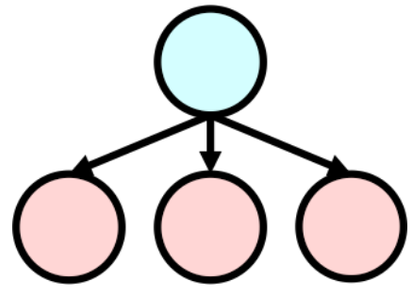
Data: a mixture of unlabeled sub-populations



Examples of Mixture Models



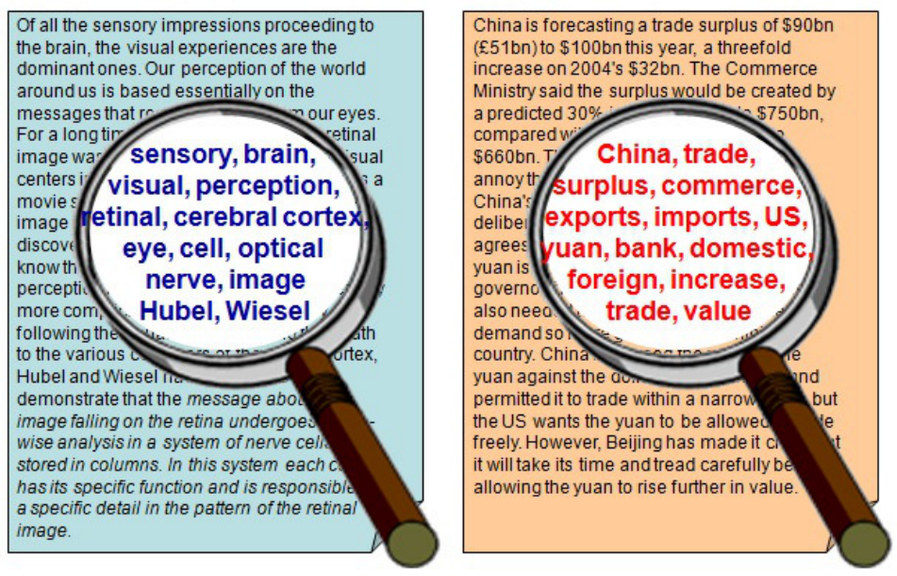
Gaussian Mixtures (GMMs)



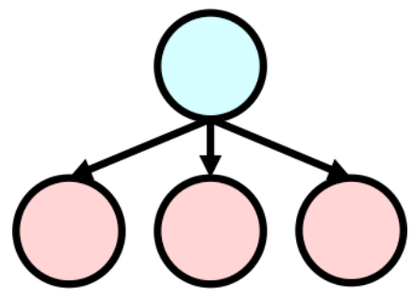
Cluster

θ

data points in space



Topic Models (Bag of Words)

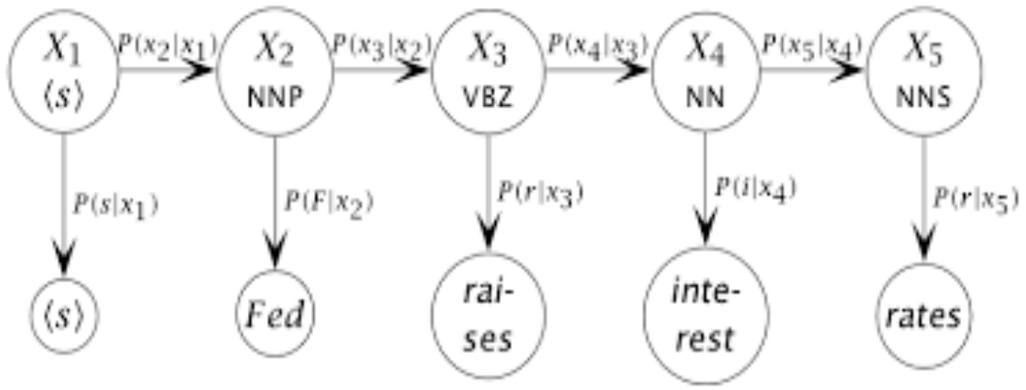


Topic

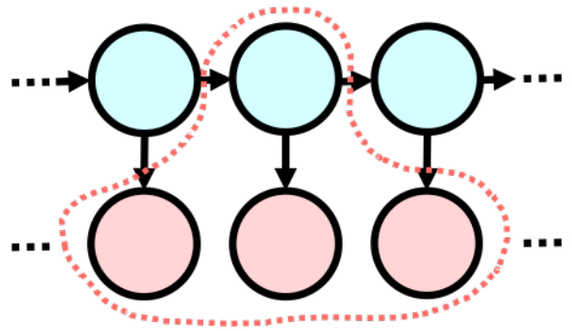
θ

words
in each document

Examples of Mixture Models



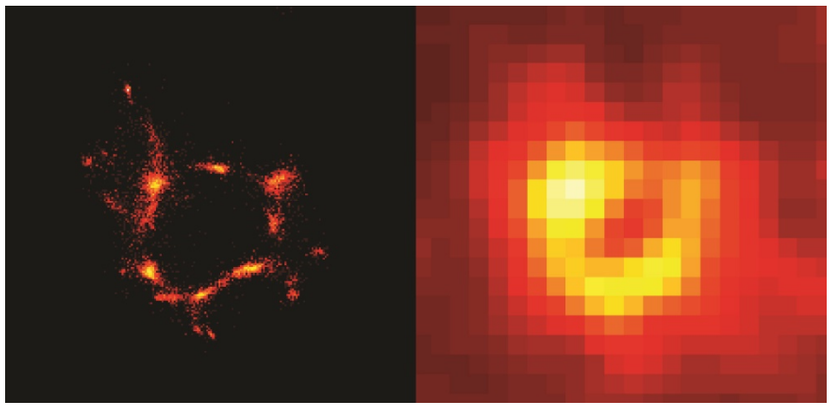
Hidden Markov Models (HMM)



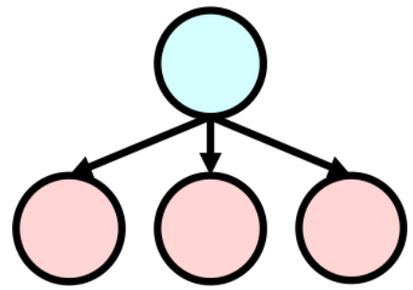
Current state

θ

Past, current, future observations



Super-Resolution

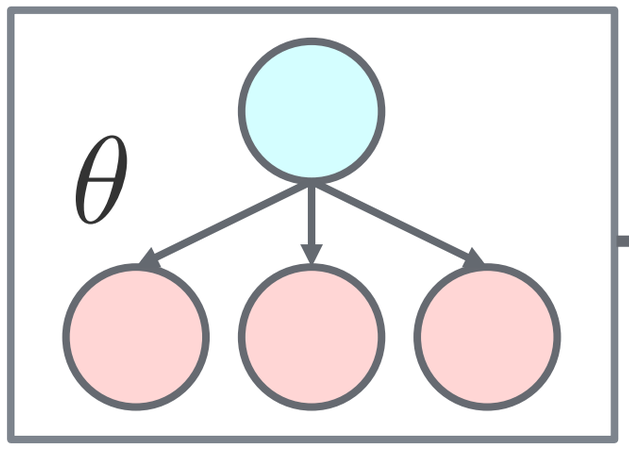


Source

θ

Complex sinusoids

Learning Mixture Models



Samples

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

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

- Marginal distribution of the observables is a superposition of simple distributions

$$\Pr_\theta(X) = \sum_{k=1}^{K} \underbrace{\Pr_\theta(H = k)}_{} \cdot \underbrace{\Pr_\theta(X|H = k)}_{}$$

θ = (#mixture components, mixing weights, conditional probabilities)

- Given N i.i.d. samples of observable variables, estimate the model parameters $\widehat{\theta}$

$$\left\| \widehat{\theta} - \theta \right\| \leq \epsilon$$

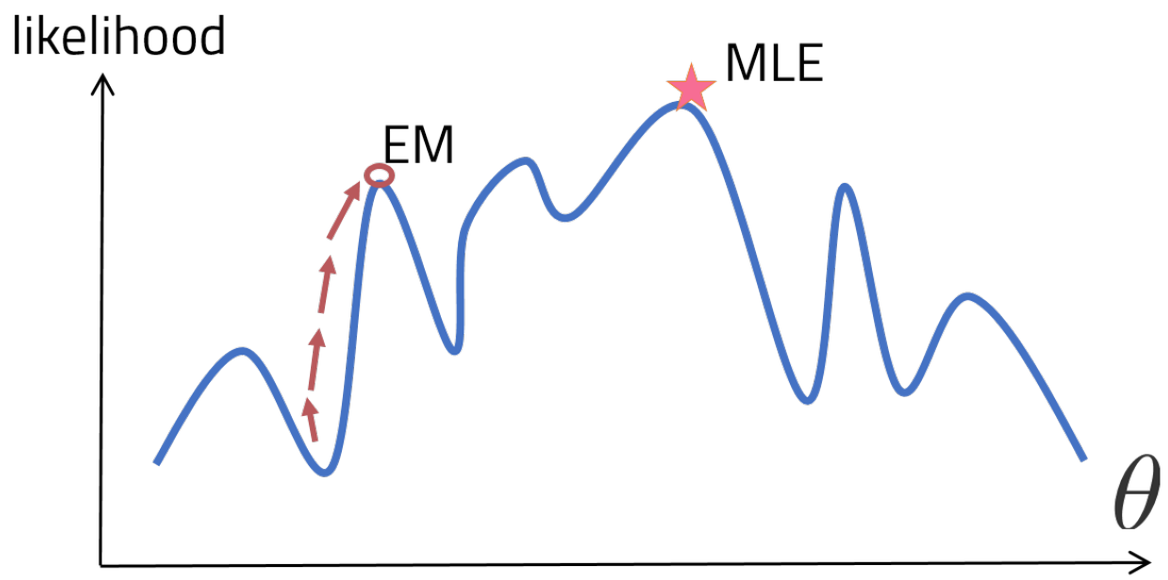
Challenges in Learning Mixture Models 1

$$\Pr_\theta(X) = \sum_k \Pr_\theta(H = k) \Pr_\theta(X|H = k)$$

- Likelihood function is non-convex in model parameters

MLE is computationally intractable

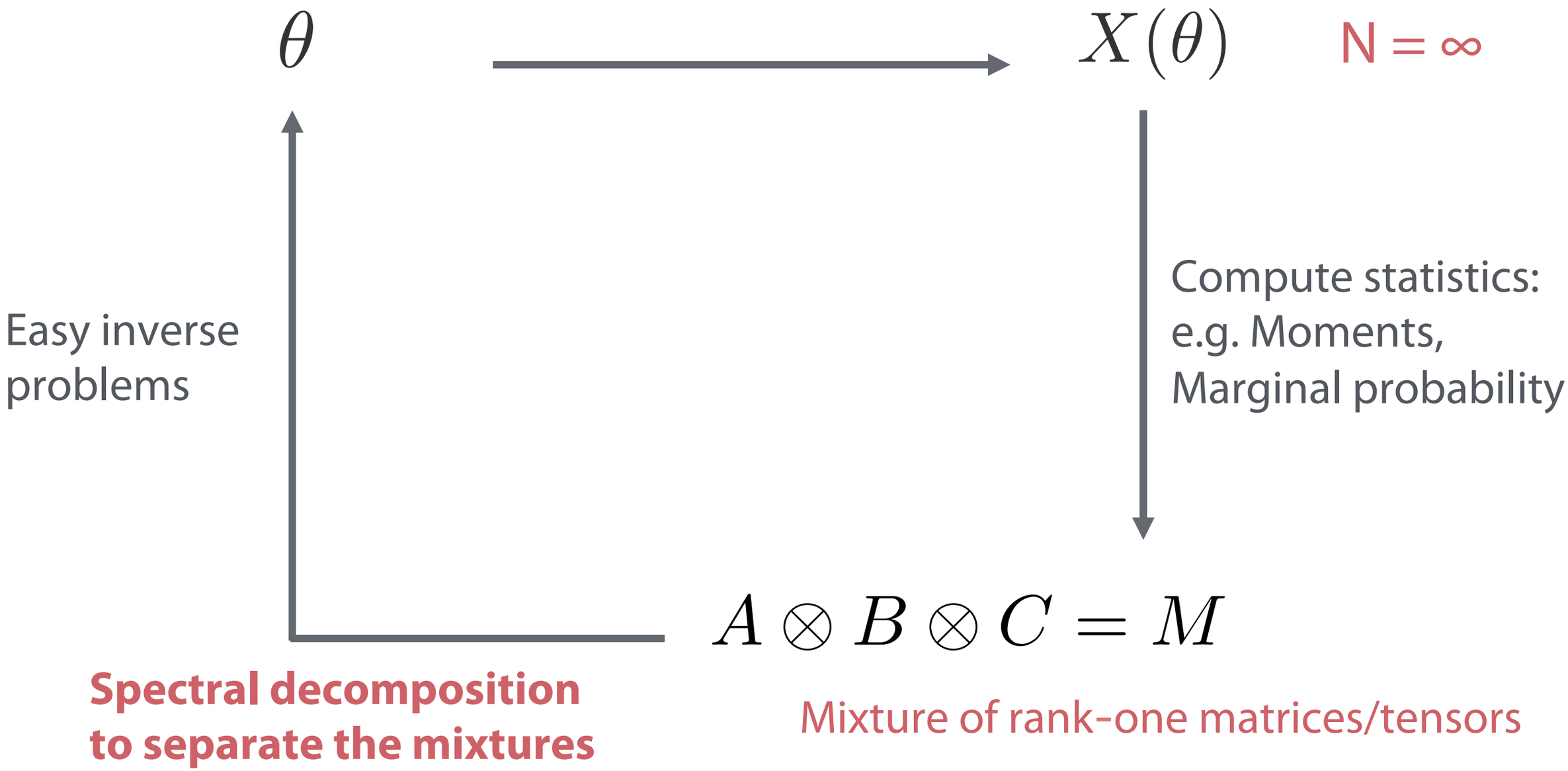
EM heuristics lack performance guarantee, get stuck at local optimums



Challenges in Learning Mixture Models 2

- Moment matching method have suboptimal sample complexity

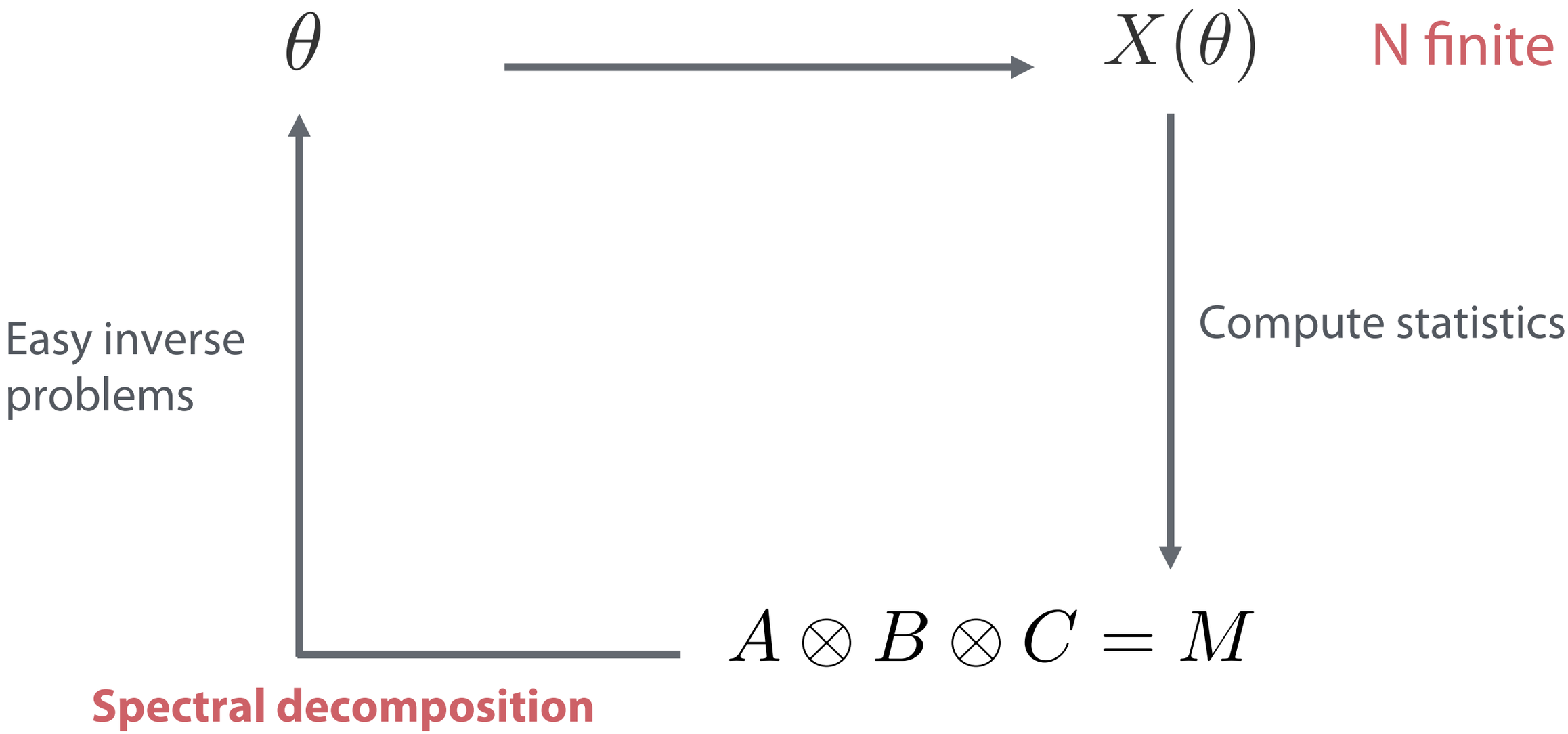
  Spectral algorithms can only handle simple models



Challenges in Learning Mixture Models 2

- Moment matching method have suboptimal sample complexity

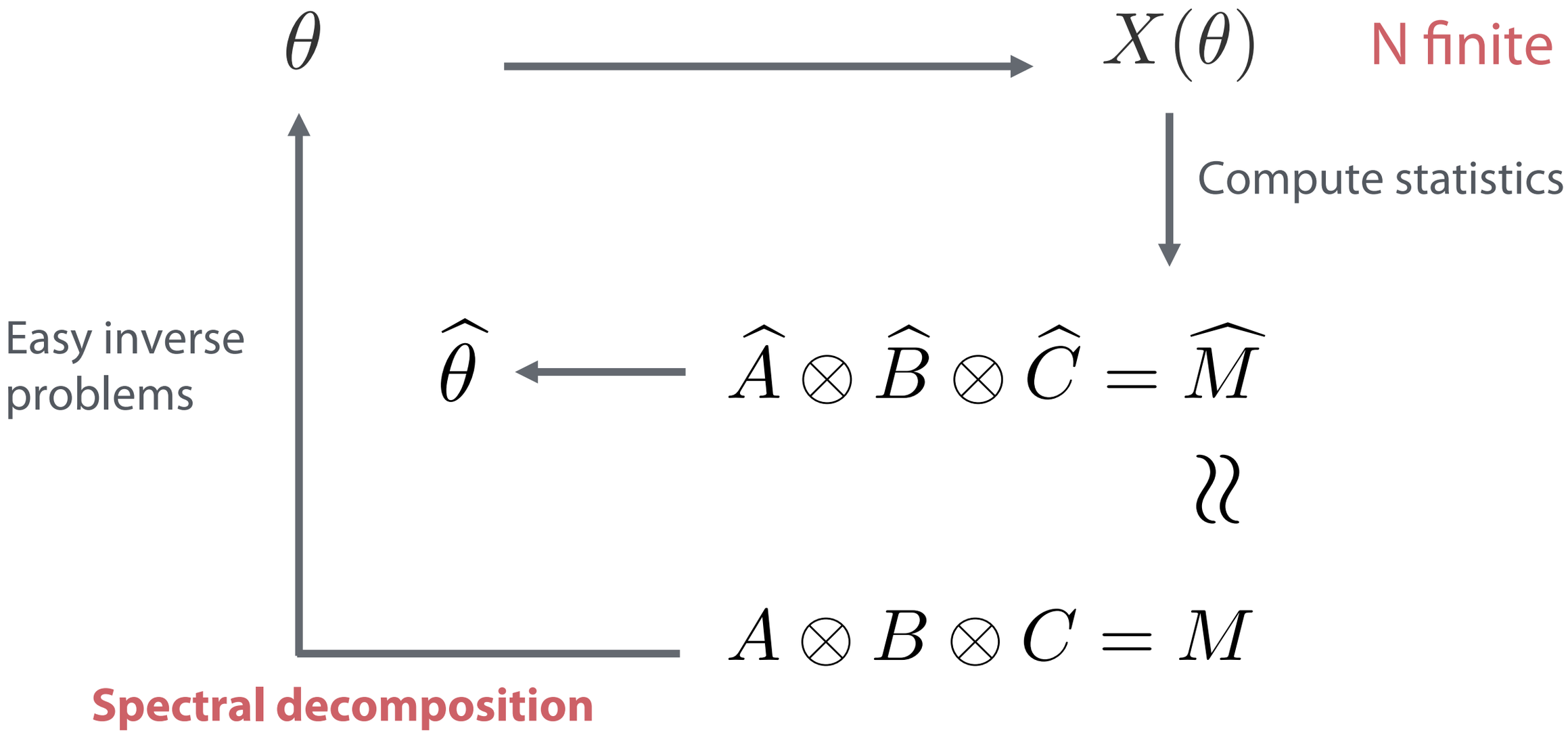
  Spectral algorithms can only handle simple models



Challenges in Learning Mixture Models 2

- Moment matching method have suboptimal sample complexity

  Spectral algorithms can only handle simple models

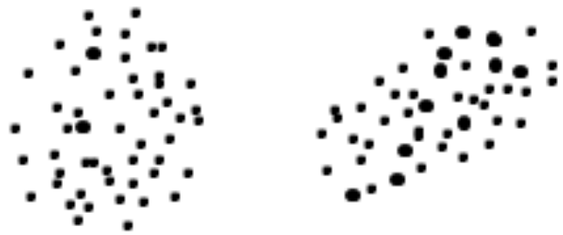


PCA, CCA, Spectral clustering, Subspace system ID,… all fit into this paradigm

Challenges in Learning Mixture Models 3

- There are “hard” mixture models, which have sample complexity lower bound that scales exponentially with model dimensions

Bad instance



Good instance

Our Contribution

Can we have statistical and computational efficient learning algorithms?

Part 1: It is possible to learn with min-max optimal sample complexity
by carefully implementing spectral algorithms.

Part 2: New algorithms for some “hard” mixture models,
analysis to show there are only a few “hard instances” ,
and our algorithms efficiently learn all other instances.

Part 3: New randomized algorithm for a “hard” mixture model,
efficiently learn any instance with high probability.

PART 1

Achieve optimal sample complexity with fast computation

PART 1

Achieve optimal sample complexity with fast computation

Estimate low rank probability matrices with linear sample complexity

Setup

θ $\longrightarrow$ X

Probability Matrix $\mathbb{B} \in \mathbb{R}_{+}^{M \times M}$

(distribution over M^2 outcomes)

N i.i.d draws

(freq counts over M^2 outcomes)

$\mathbb{B}$ is of low rank R $\quad \mathbb{B} = PWP^\top$

$$B = \frac{1}{N}\text{Poisson}(N\mathbb{B})$$

$\mathbb{B}$

.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

P

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

W

.4	.1
.1	.4

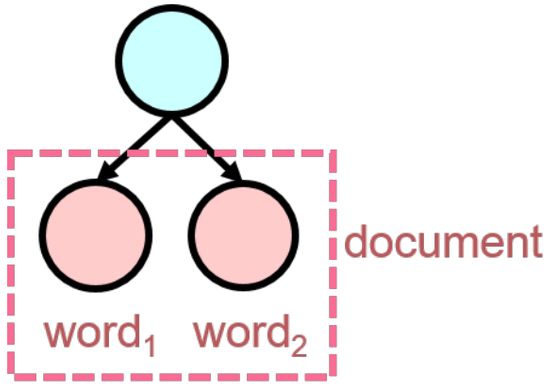
$M = 5$

$N = 20$

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

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

Connection to mixture models

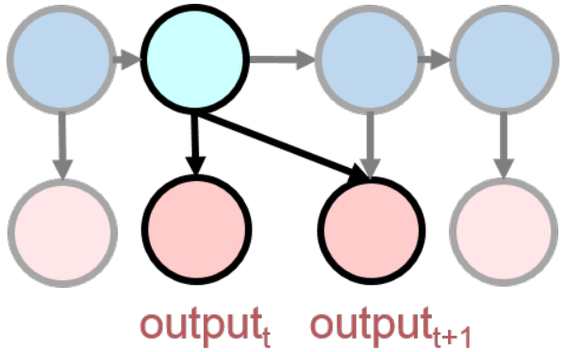


M words in vocabulary

R topics

Topic model

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



M output alphabet size

R hidden states

HMM

$\mathbb{B}$ distribution of consecutive outputs

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

Sub-optimal Attempt

θ ⟵ X

Probability Matrix $\mathbb{B} \in \mathbb{R}_{+}^{M \times M}$

$\mathbb{B}$ is of low rank R $\mathbb{B} = PWP^{\top}$

N i.i.d. samples

$B = \frac{1}{N}\text{Poisson}(N\mathbb{B})$

MLE is non-convex optimization ☹ Let's try something "spectral" ☺

Sub-optimal Attempt

θ ⟵ X

Probability Matrix $\mathbb{B} \in \mathbb{R}_{+}^{M \times M}$

$\mathbb{B}$ is of low rank R $\mathbb{B} = PWP^{\top}$

N i.i.d. samples

$B = \frac{1}{N}\text{Poisson}(N\mathbb{B})$

$$B \to \mathbb{B}, \text{ as } N \to \infty$$

- Set $\widehat{B}$ to be the rank R truncated SVD of B
- To achieve accuracy $\|\widehat{B} - \mathbb{B}\|_1 \leq \epsilon$ need $N = \Omega(M^2 \log M)$
- Not sample efficient! Hopefully $N = \Omega(M)$
- Small data in practice !

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

Main Result

- Our upper bound algorithm:
  - ✓ Rank R estimate $\widehat{B}$ with accuracy $\|\widehat{B} - \mathbb{B}\|_1 \leq \epsilon \quad \forall \epsilon > 0$
  - ✓ Using $N = O\left(max(\frac{MR^2}{\epsilon_0^4}, \frac{MR}{\epsilon^2})\right)$ number of sample draws
  - ✓ Runtime $O(M^3)$

  Lead to improved spectral algorithms for learning

- 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

  Testing property is no easier than estimating ?!

"Recovering Structured Probability Matrices " **H**, S. Kakade, W. Kong, G. Valiant

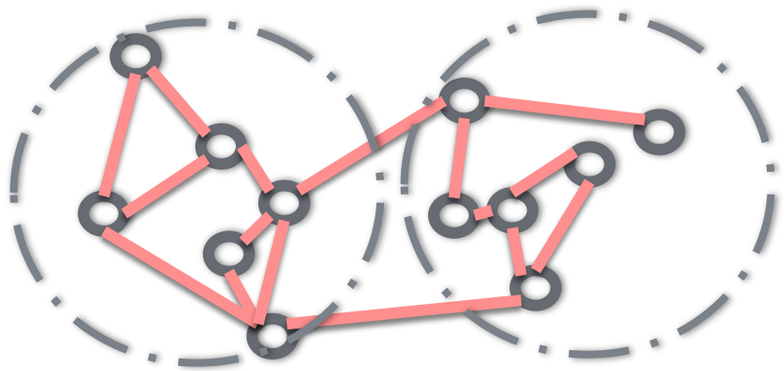
Algorithmic Idea

We capitalize the idea of community detection in stochastic block model.
SBM is a special case of our formulation, with homogeneous nodes.

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

Algorithmic Idea

We capitalize the idea of community detection in stochastic block model.
SBM is a special case of our formulation, with homogeneous nodes.

M nodes 2 communities

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

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

Regularize Truncated SVD [Le, Levina, Vershynin]

remove heavy row/column from B, run rank-2 SVD on the remaining graph

.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

Algorithmic Idea

We capitalize the idea of community detection in stochastic block model.
SBM is a special case of our formulation, with homogeneous nodes.

M nodes 2 communities	Expected connection	$\mathbb{B} = pp^\top + qq^\top$
	Adjacency matrix	$B = \text{Bernoulli}(N\mathbb{B})$

Regularize Truncated SVD [Le, Levina, Vershynin]
remove heavy row/column from B, run rank-2 SVD on the remaining graph

$M \times M$ matrix	Probability matrix	$\mathbb{B} = PWP^\top$
	Sample counts	$B = \text{Poisson}(N\mathbb{B})$

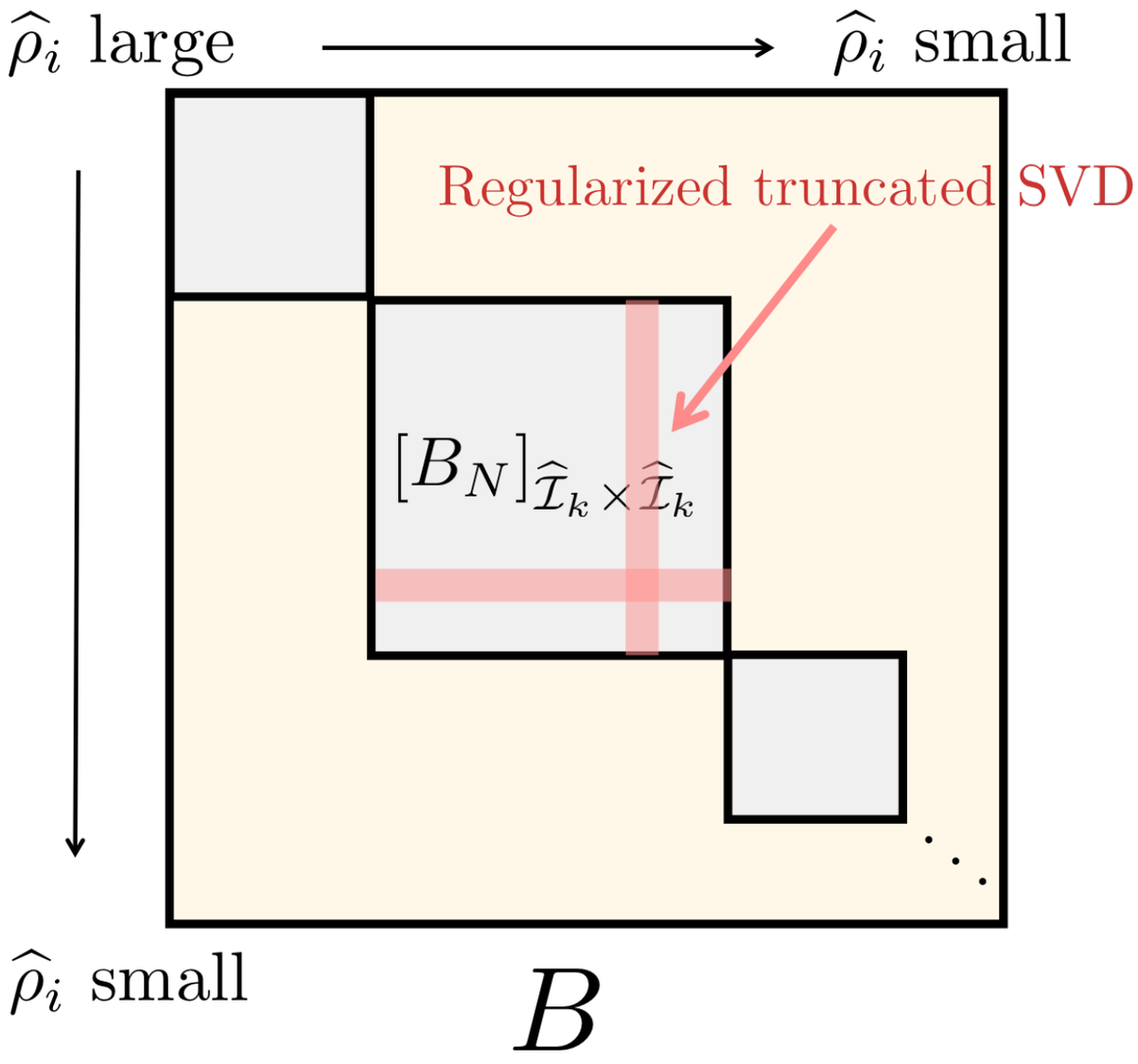
Key Challenge:
The general setup has heterogeneous nodes/ marginal probabilities

Algorithmic Idea 1, **Binning**

Sort and group nodes according to the empirical marginal probability, divide the matrix to blocks, then apply regularized t-SVD to each block

Phase 1

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



Key Challenges:

- ✓ Binning is not exact, we need to deal with spillover !
- ✓ We need to piece together estimates over bins !

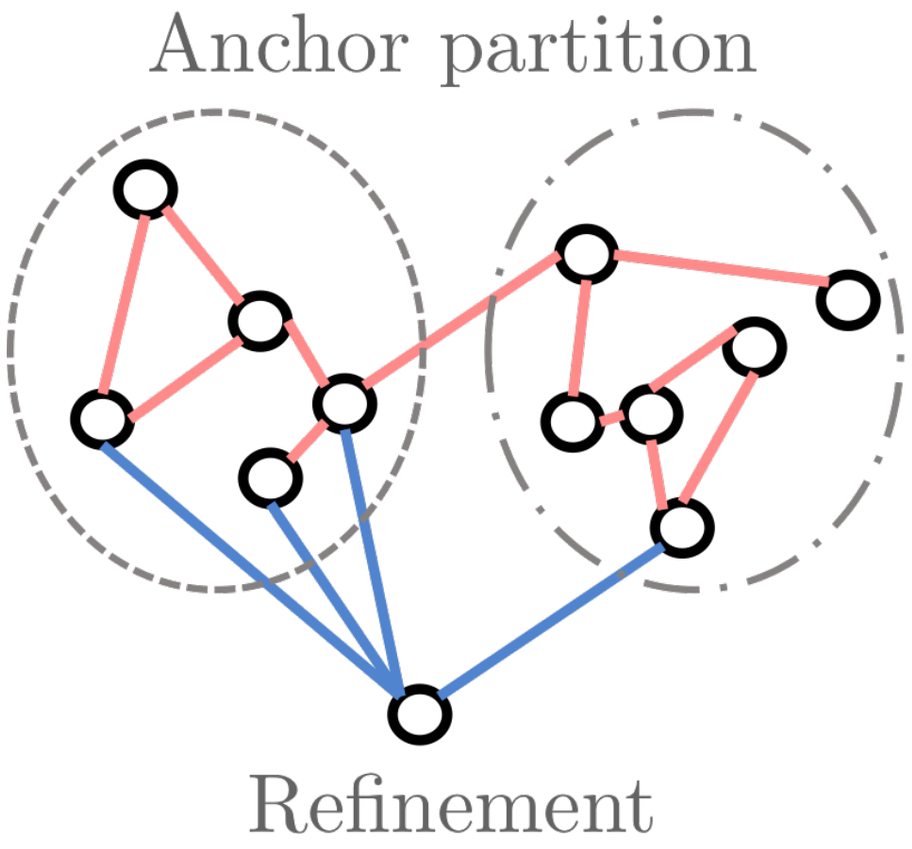
Algorithmic Idea 2, **Refinement**

The coarse estimation from Phase 1 gives some global information
Make use of that to do local refinement for each row / column

Phase 2

1. Refine the estimate for each node
   use linear regression

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



PART 2

non worst-case analysis for spectral algorithms

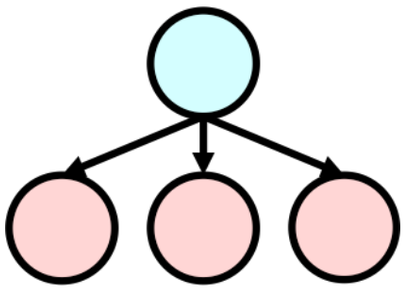
PART 2

non worst-case analysis for spectral algorithms

We study GMMs and HMMs for which there exist exponential sample complexity lower bound for worst case instances.

Worst cases are rare, and we can handle “non-worst-cases” efficiently

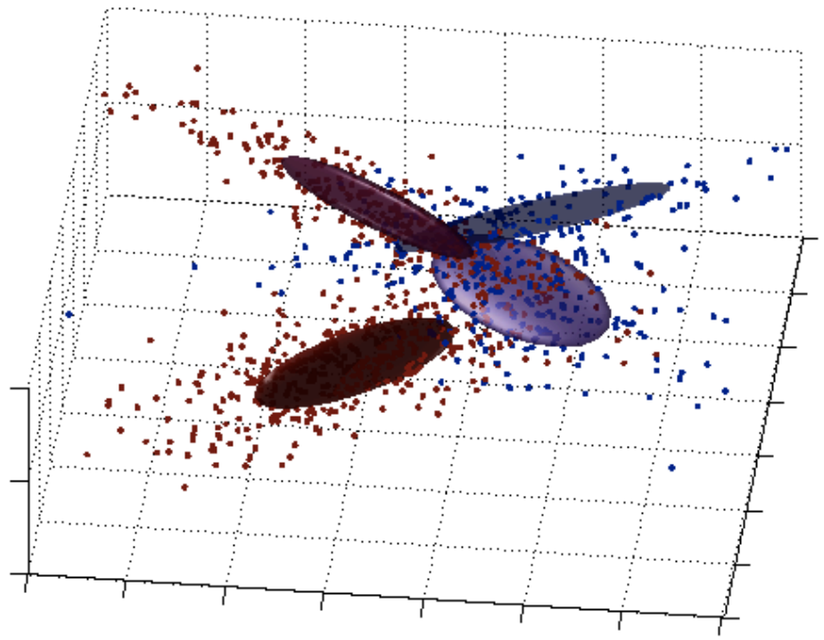
Setup



Cluster $[k]$

θ

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



mixture of k multivariate Gaussians → data points in n-dimensional space

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

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

Unsupervised clustering; customer classification; speaker recognition; object tracking…

Prior Works

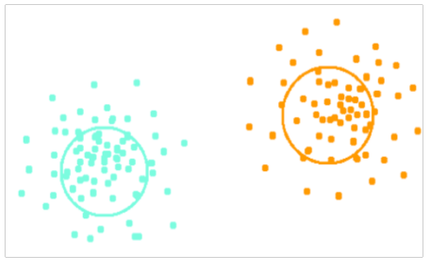
- General case

  Moment matching method [Moitra&Valiant] [Belkin&Sinha] $Poly(n, e^{O(k)^k})$

- With restrictive assumptions on model parameters

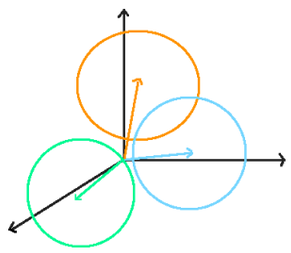
  ✓ Mean vectors are well-separated

  Pair wise clustering [Dasgupta]…[Vempala&Wang] $Poly(n, k)$



  ✓ Mean vectors of spherical Gaussians are linearly independent

  Moments tensor decomposition [Hsu&Kakade] $Poly(n, k)$



Worst case lower bound

Can we learn **every** GMM instance to target accuracy
in **poly** runtime and using **poly** samples?

No!

Exponential dependence in k for worst cases. [Moitra&Valiant]

Can we learn **most** GMM instances with **poly** algorithm?

Yes!

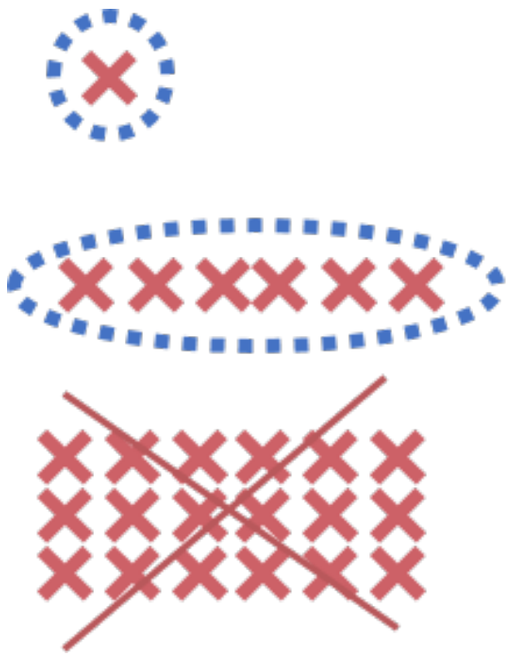
without restrictive assumptions on model parameters

Smoothed Analysis Framework

Escape from the worst cases

any instance

a ε-ball around it



bad instances surrounded
by good instances

bad instances must lie in
low-dim space of measure 0

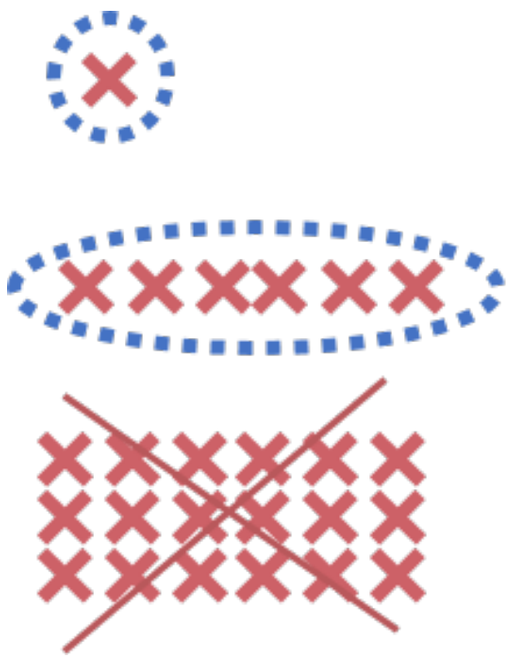
Hope: With high probability over nature's perturbation, any arbitrary instance escapes from the degenerate cases, and becomes well conditioned.

Smoothed Analysis Framework

Escape from the worst cases

any instance

a ε-ball around it



bad instances surrounded by good instances

bad instances must lie in low-dim space of measure 0

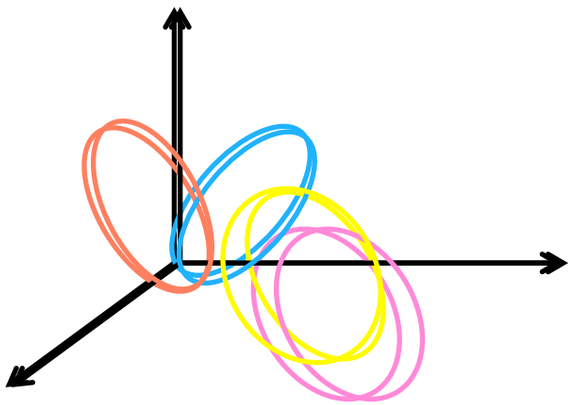
For any matrix $A \in \mathbb{R}^{m \times n}$, and $m \geq 3n$.

Perturbation E i.i.d. Gaussian $\mathcal{N}(0, \epsilon^2)$.

$$\sigma_n(A + E) \geq \epsilon\sqrt{m}$$

Smoothed Analysis Framework

Escape from the worst cases



For an arbitrary instance θ in the parameter space

Nature perturbs the parameters with a small amount (**ε**) of noise

Observe data generated by $\widetilde{\theta}$, algorithm estimate $\widetilde{\theta}$ w.h.p.

Main Results

- Our algorithm learns the GMM parameters up to target accuracy

  - ✓ With fully polynomial time and sample complexity $Poly(n, k, 1/\epsilon)$

  - ✓ Assumption: data in high enough dimension $n = \Omega(k^2)$

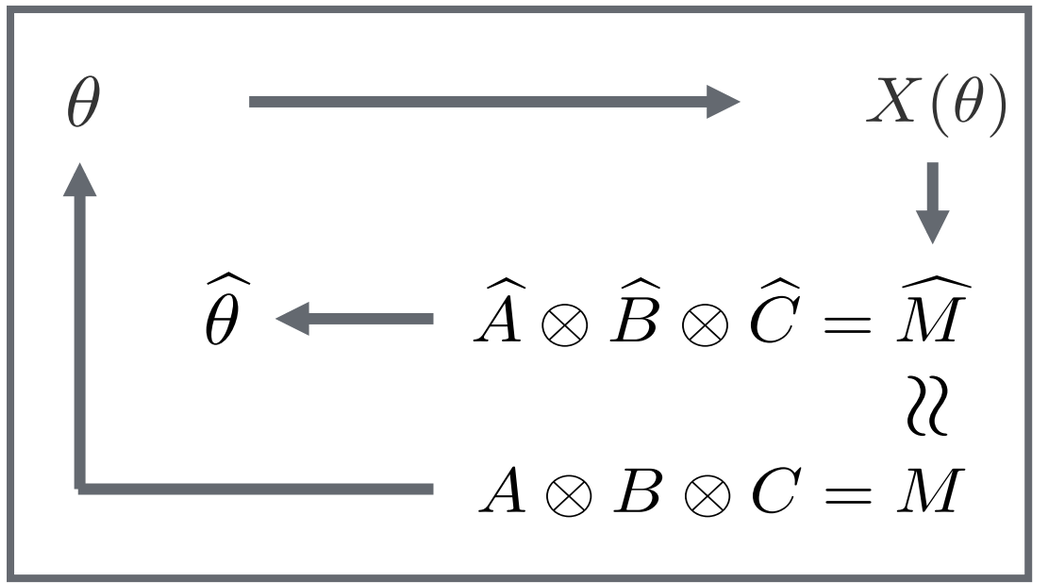
  - ✓ Under smoothed analysis: works with negligible failure probability

"Learning Mixture of Gaussians in High dimensions" R. Ge, **H**, S. Kakade (STOC 2015)

Algorithmic Ideas

Method of moments: match 4-th and 6-th order moments M_4 M_6

Key challenge: Moment tensors are not of low rank, but they have special structures



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

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

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

Structured linear projection

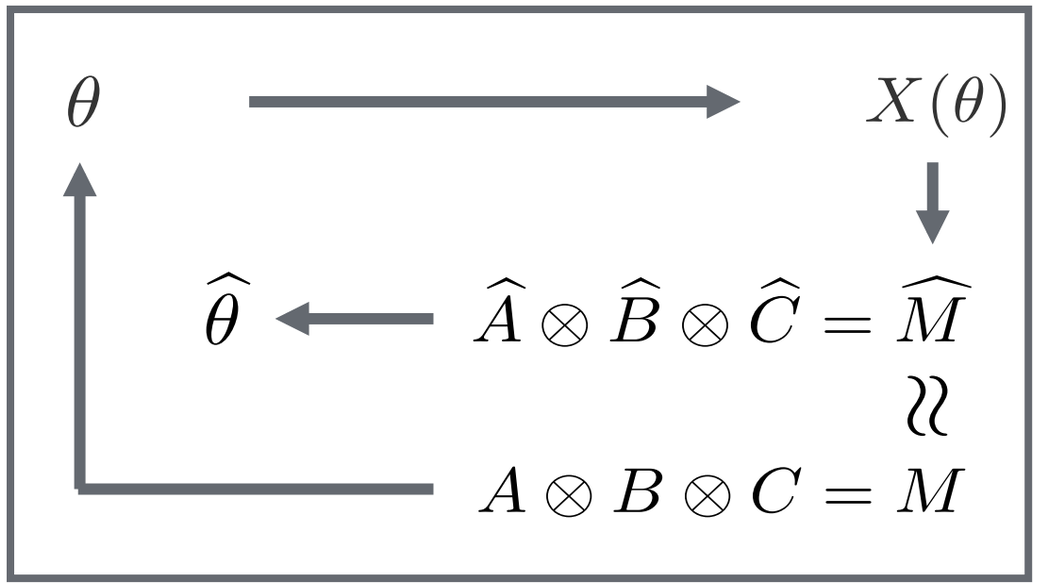
$$M_4 = \mathcal{F}_4(X_4)$$
$$M_6 = \mathcal{F}_6(X_6)$$

- Moment tensors are structured linear projections of desired low rank tensors
- Delicate algorithm to invert the structured linear projections

Algorithmic Ideas

Method of moments: match 4-th and 6-th order moments M_4 M_6

Key challenge: Moment tensors are not of low rank, but they have special structures



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

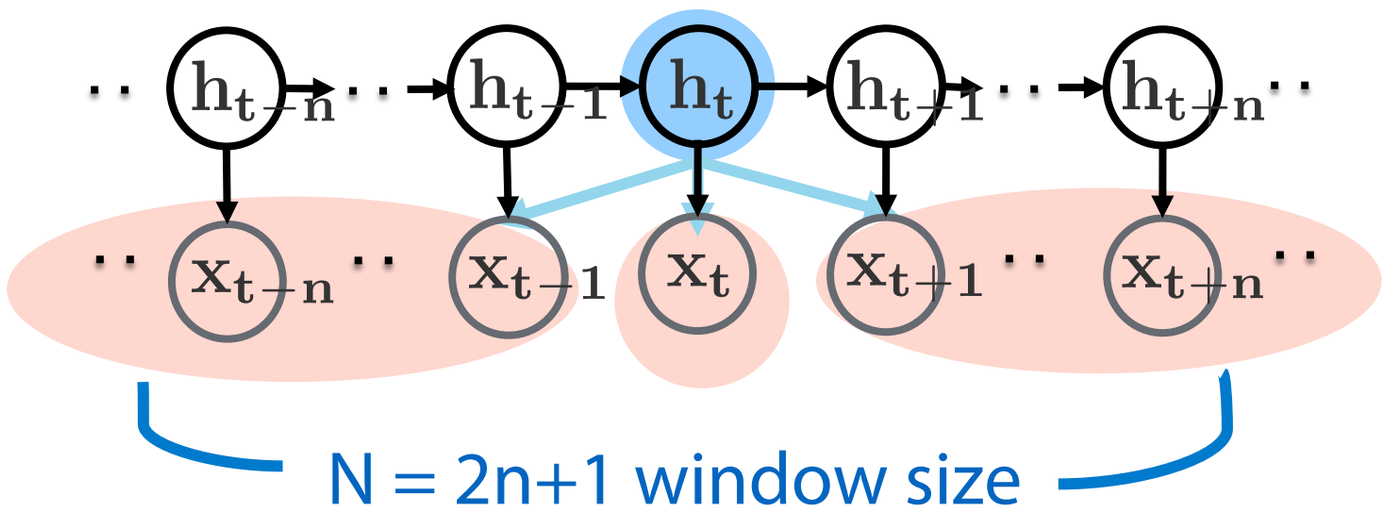
Why "high dimension n" & "smoothed analysis" help us to learn?

- ✓ We have many moment matching constraints with only low order moments

  # free parameters $\Omega(kn^2)$ < #6-th moments $\Omega(n^6)$

- ✓ The randomness in nature's perturbation makes matrices/tensors well-conditioned

HMM Setup



Hidden state $[k]$

Observation $[d]$

Transition probability matrix: $Q \in \mathbb{R}^{k \times k}$

Observation probabilities: $O \in \mathbb{R}^{d \times k}$

Given length-N output sequences, how to recover $\theta = (Q, O)$?

Our focus: How large the window size N needs to be?

Hardness Results

Hidden state $[k]$ Observation $[d]$ N = 2n+1 window size

- HMM is not efficiently PAC learnable

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

  Required window size $N = \Omega(k)$, Algorithm Complexity is $\Omega(d^k)$

Our Results

- Excluding a measure 0 set in the parameter space of $\theta = (Q, O)$

  for almost all HMM instances, the required window size is $N = \Theta(\log_d k)$

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

"Minimal Realization Problems for Hidden Markov Models"
H, R. Ge, S. Kakade, M. Dahleh (IEEE Transactions on Signal Processing, 2016)

PART 3

Randomized algorithm to tackle worst case lower bound

PART 3

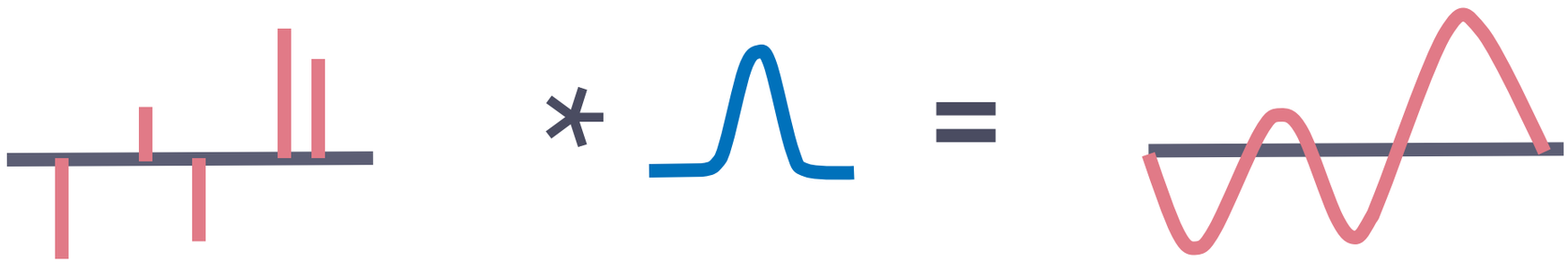
Randomized algorithm to tackle worst case lower bound

Our algorithm for super-resolution has quadratic complexity

Setup

Low pass blurring of high resolution but simple images

θ —Band-limited→ $X(\theta)$



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

- ✓ small number of Fourier measurements
- ✓ at frequencies much lower than Nyquist

Problem Formulation

✓ Recover point sources (a mixture of k points in n-dimensional space)

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

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

✓ Measure by band-limited and noisy Fourier transformation

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

$\|s\|_\infty \leq$ cutoff freq bounded noise $|z(s)| \leq \epsilon_z, \forall s$

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

Prior Works

$$\widetilde{f}(s) = \sum_{j=1}^{k} w_j e^{i\pi <\mu^{(j)}, s>} + z(s) \qquad \Delta = \min_{j \neq j'} \|\mu^{(j)} - \mu^{(j')}\|_2$$

- **1-dimensional** $\mu^{(j)}$
  - ✓ Take uniform measurements on the grid $s \in \{-N, \ldots, -1, 0, 1, \ldots, N\}$
  - ✓ SDP algorithm with cut-off frequency $N = \Omega(\frac{1}{\Delta})$ [Candes, Fernandez-Granda]
  - ✓ Lower bound result $N > \frac{C}{\Delta}$ [Moitra]
  - ✓ One can use $k \log(k)$ random measurements to recover $2N$ measurements [Tang, Bhaskar, Shah, Recht]
- **n-dimensional** $\mu^{(j)}$
  - ✓ Multi-dimensional grid $s \in \{-N, \ldots, -1, 0, 1, \ldots, N\}^n$
  - ✓ Algorithm complexity $O\left(poly(k, \frac{1}{\Delta})\right)^n$

Main Result

- Our randomized algorithm achieves stable recovery
  - ✓ uses a number of $\widetilde{O}((k+n)^2)$ Fourier measurements
  - ✓ cutoff freq of the measurements bounded by $O(1/\Delta)$
  - ✓ algorithm runtime $\widetilde{O}((k+n)^3)$
  - ✓ algorithm works with negligible failure probability

	cutoff freq	measurements	runtime
SDP	$\frac{C_n}{\Delta_\infty}$	$(\frac{1}{\Delta_\infty})^n$	$poly((\frac{1}{\Delta_\infty})^n, k)$
Ours	$\frac{\log(kn)}{\Delta}$	$(k\log(k)+n)^2$	$(k\log(k)+n)^3$

"Super-Resolution off the Grid" **H**, S. Kakade (NIPS 2015)

Algorithmic Idea

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

- ✓ Take Fourier measurements at **random frequencies** S
- ✓ Create structure so the measurements can be arranged as a low rank tensor F

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

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

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

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

- ✓ Skip intermediate step of recovering $\Omega(N^n)$ measurements on the hyper-grid directly work with a small number of random measurements

Algorithmic Idea

✧ Why **we** do not contradict the **lower bound**?

$$\widetilde{O}(k^2+n^2) \quad \text{vs} \quad O\Big(poly(k,\frac{1}{\Delta})\Big)^n$$

✓ If we design a **fixed** grid of frequency to take measurements
there always exists model instances such that the deterministic grid fails

✓ We pick the locations of frequencies at **random.**
for any model instance, the random algo works with high probability

Conclusion

- Spectral methods are powerful tools for learning mixture models.
- It's possible to learn with optimal sample complexity with carefully implemented spectral algorithm
- We can go beyond worst case analysis by exploiting the randomness in the analysis / algorithm.

Future work

- Addressing the robustness issue of spectral algorithms
- Extend the algorithmic and analysis techniques to other learning problems


THANK YOU

References

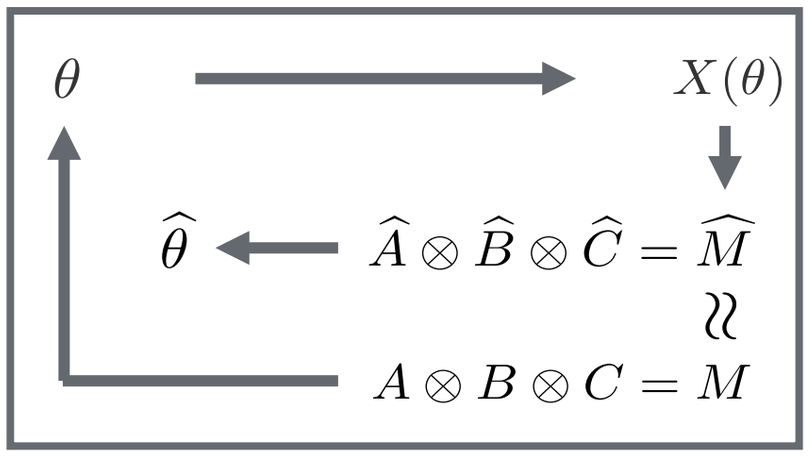
- "Learning Mixture of Gaussians in High dimensions"
  R. Ge, **H**, S. Kakade (STOC 2015)

- "Super-Resolution off the Grid"
  H, S. Kakade (NIPS 2015)

- "Minimal Realization Problems for Hidden Markov Models"
  H, R. Ge, S. Kakade, M. Dahleh (IEEE Transactions on Signal Processing, 2016)

- "Recovering Structured Probability Matrices "
  H, S. Kakade, W. Kong, G. Valiant, (submitted to FOCS 2016)

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. M is a low rank tensor of rank k

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

$$A = \Pr\left(x_1, x_2, \ldots, x_n \middle| h_0\right)$$
$$B = \Pr\left(x_{-1}, x_{-2}, \ldots, x_{-n} \middle| h_0\right)$$
$$C = \Pr\left(x_0, h_0\right)$$

2. Extract Q, O from tensor factors A B

$$A = \underbrace{(O \odot (O \odot (O \odot \ldots (O \odot O}_{n} \underbrace{Q) \ldots)Q)Q)Q}_{n},$$

$$B = \underbrace{(O \odot (O \odot (O \odot \ldots (O \odot O}_{n} \underbrace{\widetilde{Q}) \ldots)\widetilde{Q})\widetilde{Q})\widetilde{Q}}_{n},$$

Key lemma:
How large window size needs to be, so that we have unique tensor decomp

Our careful generic analysis:
If $N = \Theta(\log_d k)$, worst cases all lie in a measure 0 set!