Efficient Spectral Methods for Learning Mixture Models

Qingqing Huang

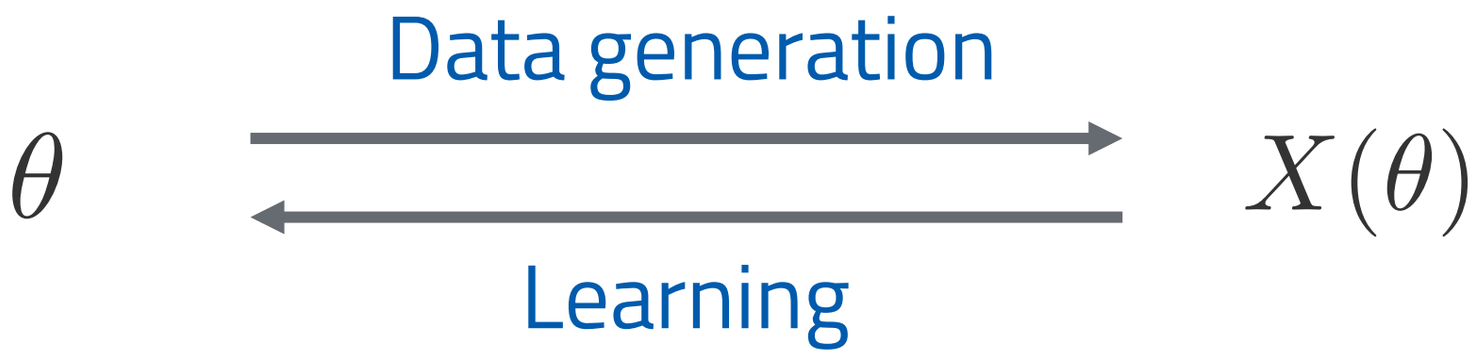
2016 February

Massachusetts Institute of Technology

LIDS Laboratory for Information & Decision Systems

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

Learning

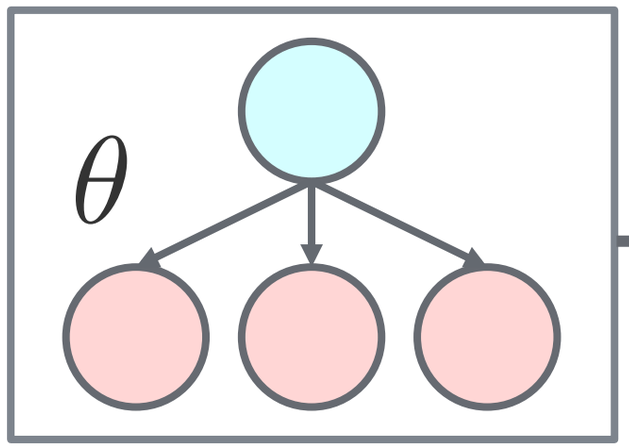


- Infer about the underlying rule θ
  (Estimation, approximation, property testing, optimization of f(θ))

- Challenge:
  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 $\uparrow \dim(\theta)$

Learning Mixture Models



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

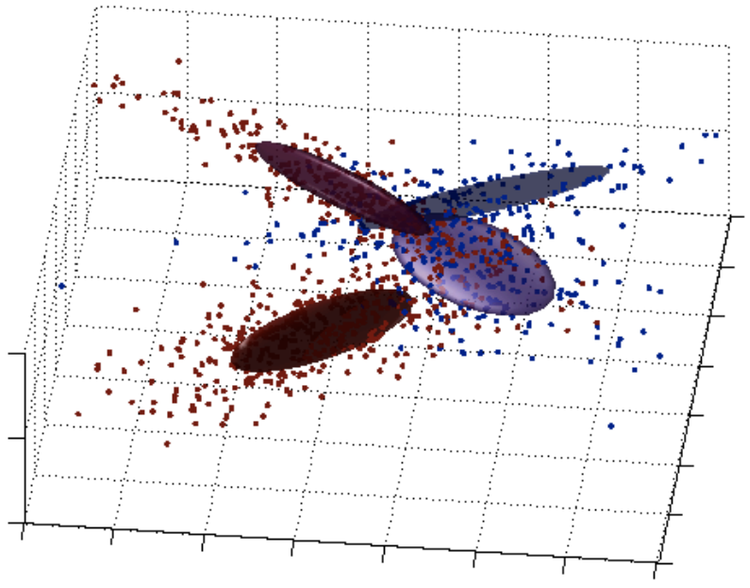
$$\Pr(X) = \sum_{k=1}^{K} \underbrace{\Pr(H = k)} \cdot \underbrace{\Pr(X|H = k)}$$

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

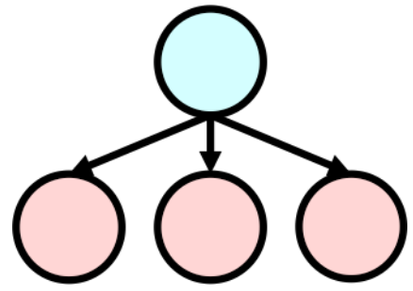
- Given N i.i.d. samples of observable variables, estimate the model parameters $\widehat{\theta}$

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

Examples of Mixture Models



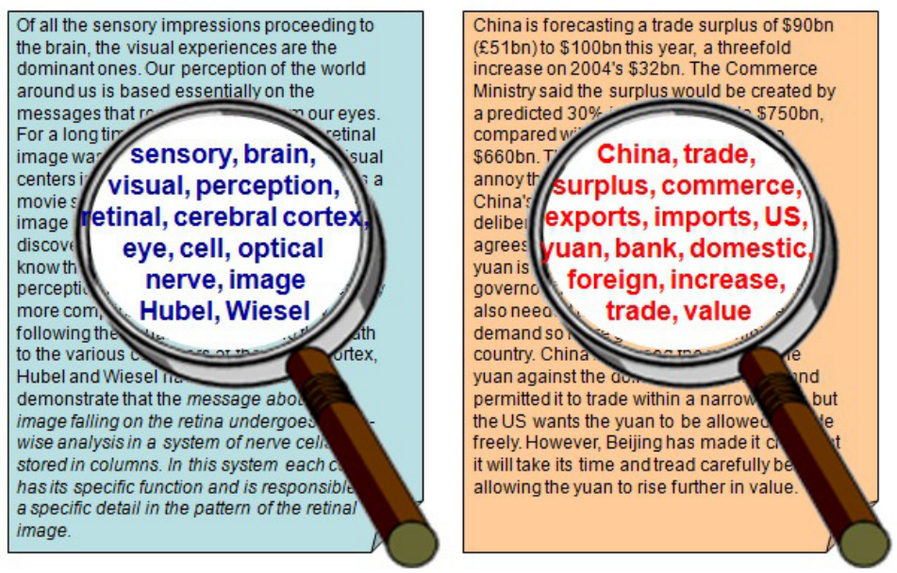
Gaussian Mixtures (GMMs)



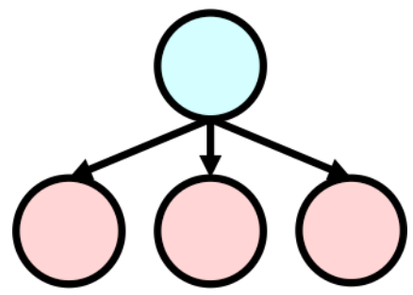
Cluster

θ

data points in space



Topic Models (Bag of Words)

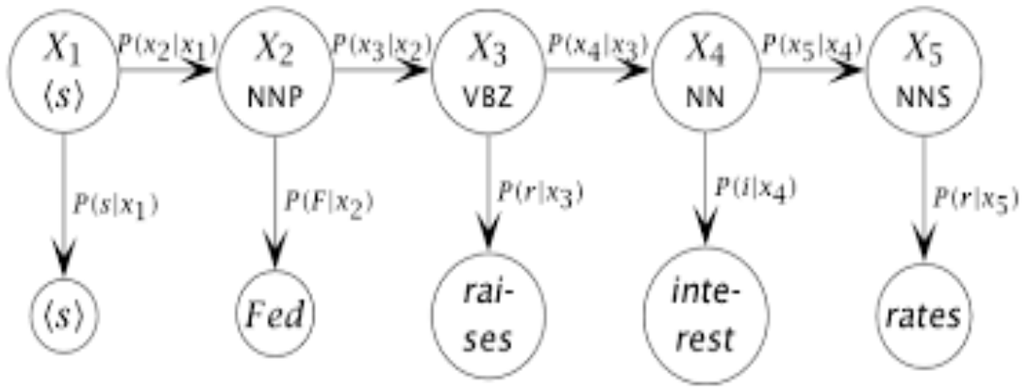


Topic

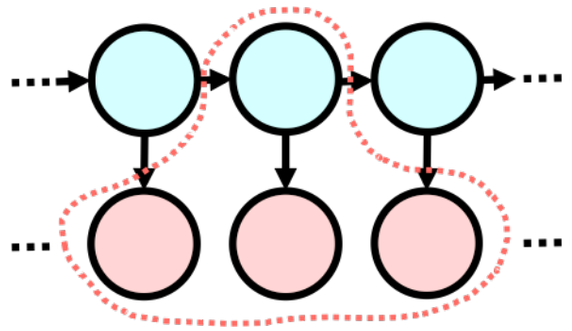
θ

words
in each document

Examples of Mixture Models



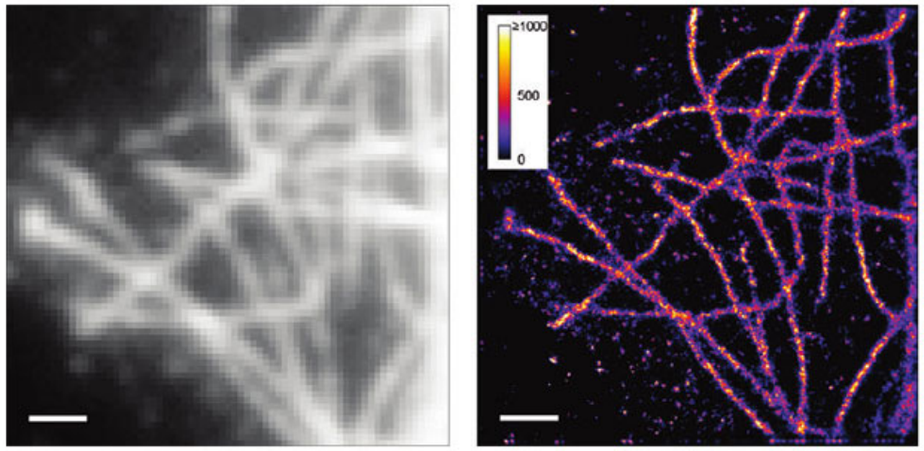
Hidden Markov Models (HMM)



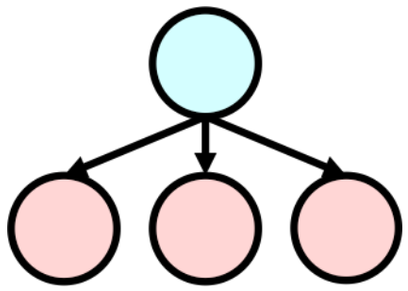
Current state

θ

Past, current, future observations



Super-Resolution

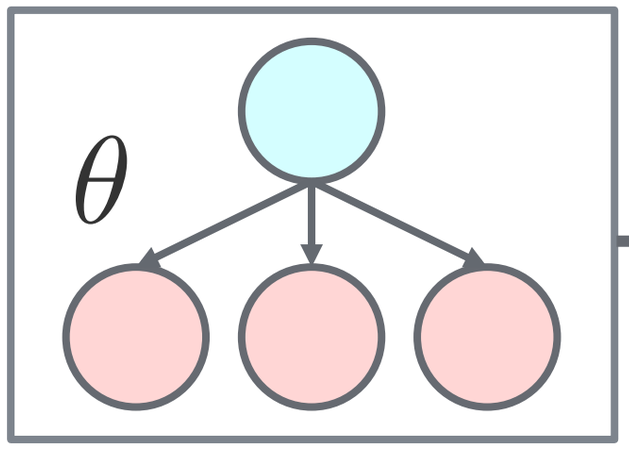


Source

θ

Complex sinusoids

Learning Mixture Models



Samples

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

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

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

What do we know about sample complexity and computation complexity ?

- There exist exponential lower bounds for sample complexity (worst case analysis)
- Maximum Likelihood Estimation is non-convex optimization (EM is heuristics)

Challenges in Learning Mixture Models

- There exist exponential lower bounds for sample complexity (worst case analysis)

Can we learn "**non-worst-cases**" with provably efficient algorithms ?

- Maximum Likelihood Estimation is non-convex optimization (EM is heuristics)

Can we achieve **statistical efficiency** with **tractable** computation?

Contribution / Outline of the talk

- There exist exponential lower bounds for sample complexity (worst case analysis)

Can we learn "**non-worst-cases**" with provably efficient algorithms ?

✓ Spectral algorithms for learning GMMs, HMMs, Super-resolution

✓ Go beyond worst cases with **randomness** in analysis and algorithm

- Maximum Likelihood Estimation is non-convex optimization (EM is heuristics)

Can we achieve **statistical efficiency** with **tractable** computation?

✓ Denoising low rank probability matrix with **linear** sample complexity (minmax optimal)

Paradigm of Spectral Algorithms for Learning

Mixture models

Mixture of conditional distributions

θ —Samples→ $X(\theta)$ N = ∞

Easy inverse problems

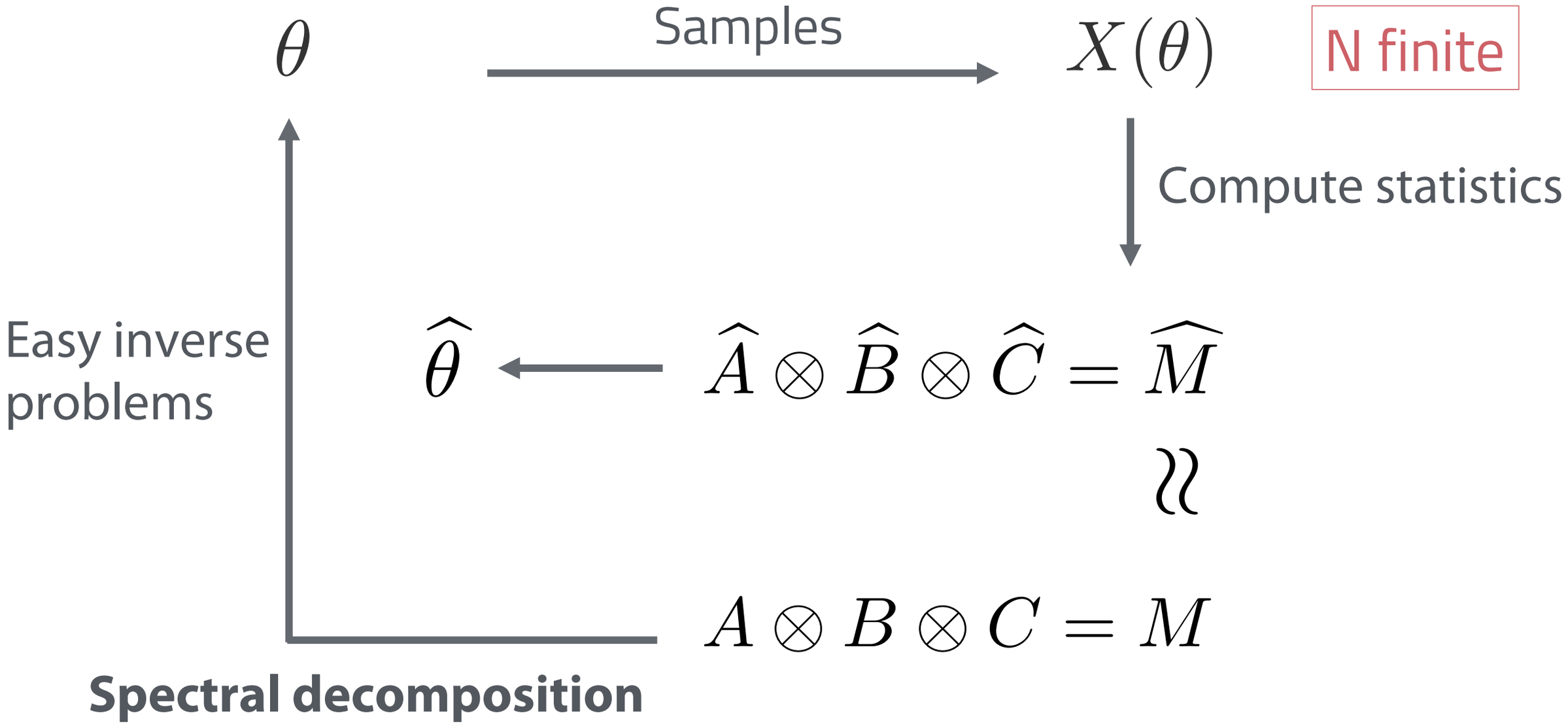
Compute statistics:
e.g. Moments,
Marginal probability

$A \otimes B \otimes C = M$

Spectral decomposition to separate the mixtures

Mixture of rank-one matrices/tensors

Paradigm of Spectral Algorithms for Learning



- ✓ PCA, Spectral clustering, Subspace system ID,… fit into this paradigm
- ✓ Problem specific algorithm design (what statistics M to use?)
  analysis (is the spectral decomposition stable?)

PART 1

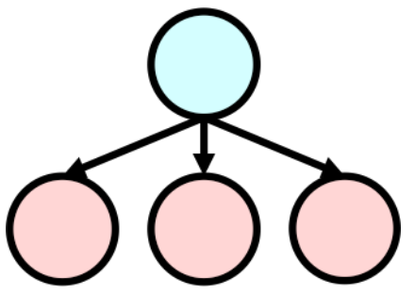
Provably efficient spectral algorithms for learning mixture models

1.1 Learn GMMs: Go beyond worst cases by **smoothed analysis**

1.2 Learn HMMs: Go beyond worst cases by **generic analysis**

1.3 Super-resolution: Go beyond worst cases by **randomized algorithm**

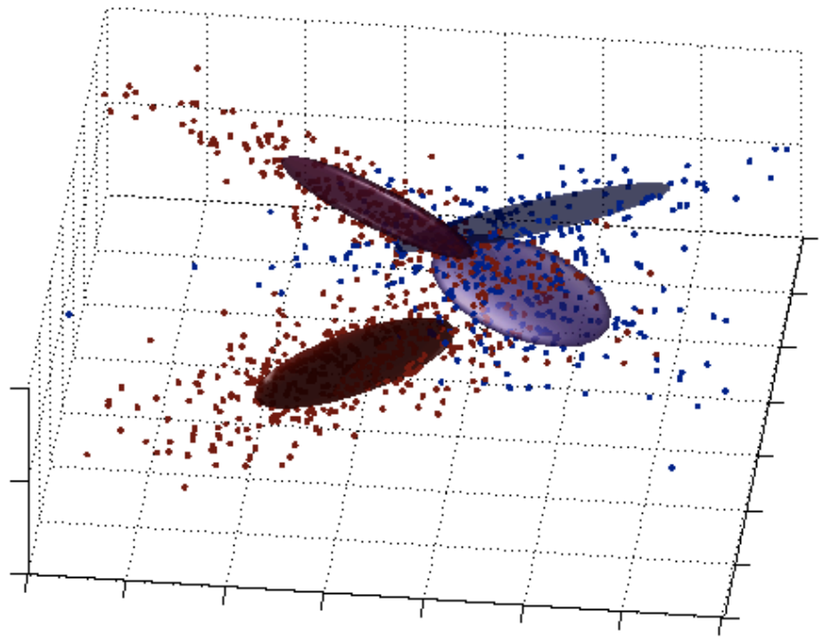
1.1 Learn GMMs: Setup



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 $\sum^{(i)}$

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

1.1 Learn GMMs: Prior Works

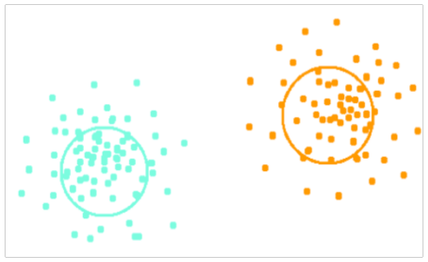
- General case

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

- With restrictive assumptions on model parameters

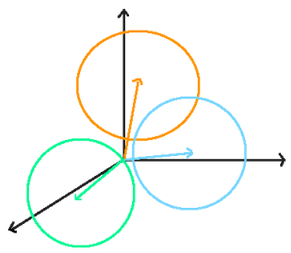
  ✓ Mean vectors are well-separated

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



  ✓ Mean vectors of spherical Gaussians are linearly independent

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



1.1 Learn GMMs: 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?
without restrictive assumptions on model parameters

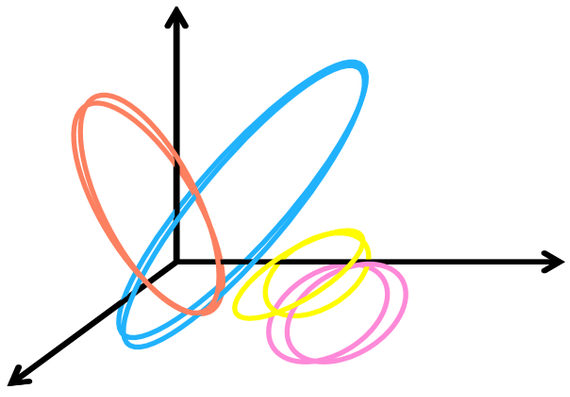
Yes!

Worst cases are not everywhere, and we can handle “non-worst-cases”

1.1 Learn GMMs: Smoothed Analysis Framework

Escape from the worst cases

For an arbitrary instance θ



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

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

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

- ✓ Bridge worst case and average case algo analysis [Spielman&Teng]
- ✓ A stronger notion than generic analysis

Our Goal:

Given samples from perturbed GMM, learn the perturbed parameters with negligible failure probability over nature's perturbation

1.1 Learn GMMs

Main Results

- Our algorithm learns the GMM parameters up to accuracy ε

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

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

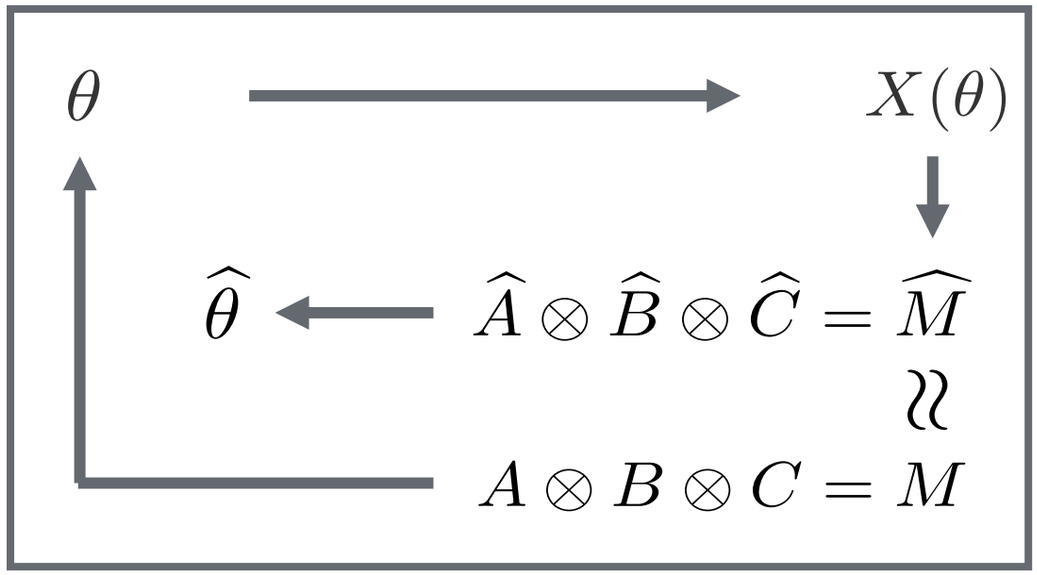
  - ✓ Under smoothed analysis: works with negligible failure probability

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

1.1 Learn GMMs: 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

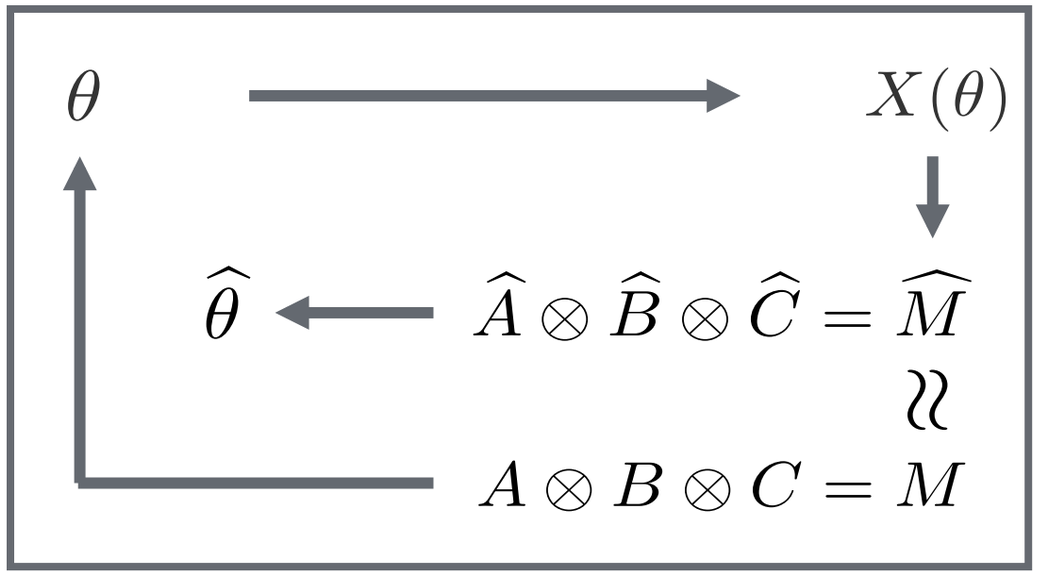
$$M_4 = \mathcal{F}_4(X_4)$$
$$M_6 = \mathcal{F}_6(X_6)$$

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

1.1 Learn GMMs: 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

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

PART 1

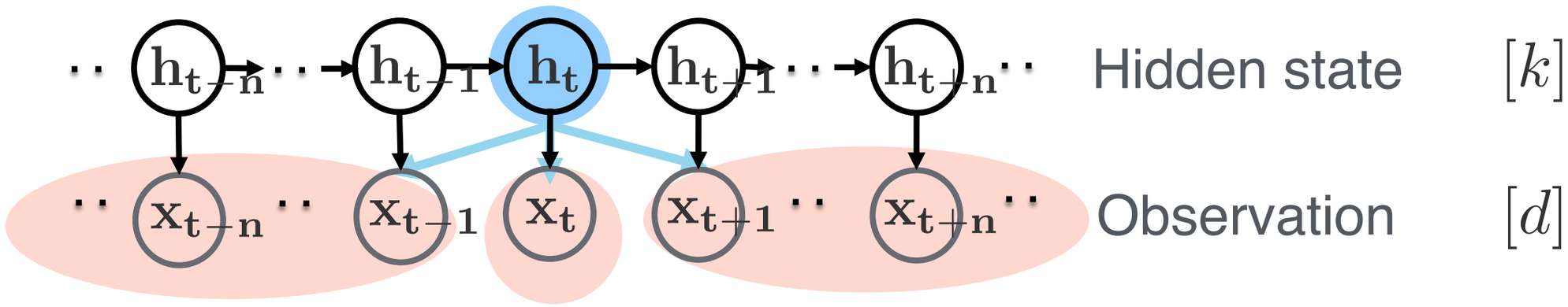
Provably efficient spectral algorithms for learning mixture models

1.1 Learn GMMs: Go beyond worst cases by **smoothed analysis**

1.2 Learn HMMs: Go beyond worst cases by **generic analysis**

1.3 Super-resolution: Go beyond worst cases by **randomized algorithm**

1.2 Learn HMMs: Setup



N = 2n+1 window size

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

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

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

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

1.2 Learn HMMs:

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

Hardness Results

- HMM is not efficiently PAC learnable, under noisy parity assumption

  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 Result

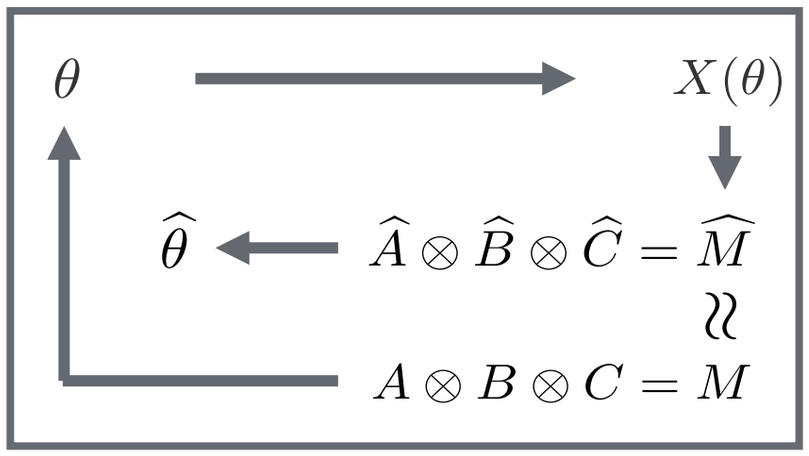
- Excluding a measure 0 set in the parameter space of $\theta = (Q, O)$

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

- Spectral algo 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)

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. 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 challenge:
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!

PART 1

Provably efficient spectral algorithms for learning mixture models

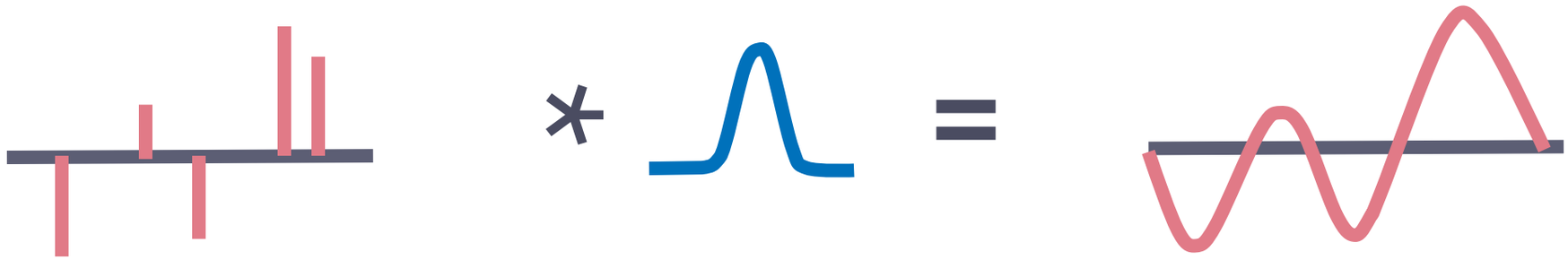
1.1 Learn GMMs: Go beyond worst cases by **smoothed analysis**

1.2 Learn HMMs: Go beyond worst cases by **generic analysis**

1.3 Super-resolution: Go beyond worst cases by **randomized algorithm**

1.3 Super-Resolution: Setup

θ —Band-limited→ $X(\theta)$



Take Fourier measurement of the band-limited observation

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

- ✓ small number of Fourier measurements
- ✓ Low cutoff frequency

1.3 Super-Resolution: Problem Formulation

✓ Recover point sources (a mixture of k vectors in d-dimensional space)

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

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

✓ Use bandlimited and noisy Fourier measurements.

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

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

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

1.3 Super-Resolution: Prior Works

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

- **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]
  - ✓ Hardness result $N > \frac{C}{\Delta}$ [Moitra]
  - ✓ One can use $k \log(k)$ random measurements to recover $2N$ measurements [Tang, Bhaskar, Shah, Recht]
- **d-dimensional** $\mu^{(j)}$
  - ✓ Mult-dim grid $s \in \{-N, \ldots, -1, 0, 1, \ldots, N\}^d$
  - ✓ Algorithm complexity $O\left(poly(k, \frac{1}{\Delta})\right)^d$

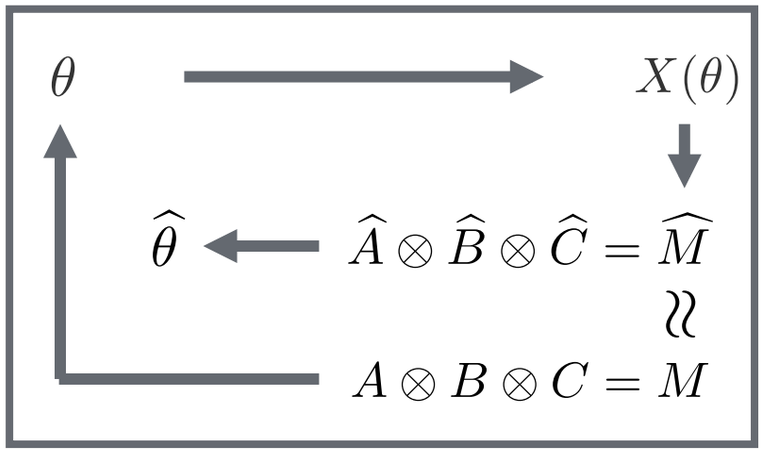
1.3 Super-Resolution: Main Result

- Our algorithm achieves stable recovery
  - ✓ using a number of $O((k+d)^2)$ Fourier measurements
  - ✓ cutoff freq of the measurements bounded by $O(1/\Delta)$
  - ✓ algorithm runtime $O((k+d)^3)$
  - ✓ algorithm works with negligible failure 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	$\frac{\log(kd)}{\Delta}$	$(k\log(k)+d)^2$	$(k\log(k)+d)^2$

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

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)

$d \times d \times 2$

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

(Vandermonde Matrix with complex nodes)

$d \times k$

- ✓ Skip intermediate step of recovering $\Omega(N^d)$ measurements on the hyper-grid
- ✓ Prony's method (Matrix-Pencil / MUSIC / …) is just choosing S deterministically

1.3 Super-Resolution: Algorithmic Idea

✧ 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

PART 2

Can we achieve optimal sample complexity in a tractable way?

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

- This problem is at the core of many spectral algorithms
- We capitalize the insights from community detection to solve it

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
(freq counts over M^2 outcomes)

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

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

.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

$\mathbb{B}$

.40
.20
.15
.15
.10

p

.15
.40
.15
.10
.20

q

$M = 5$

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

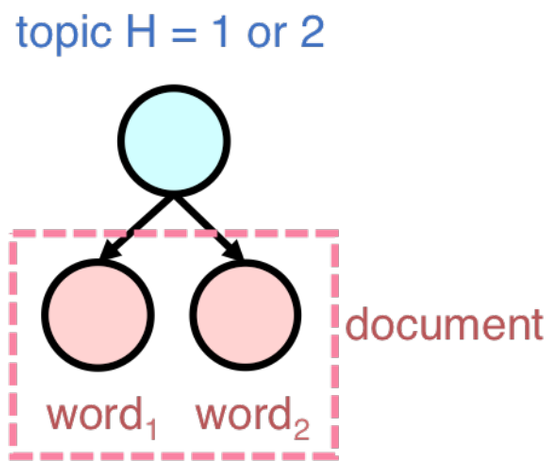
$N = 20$

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

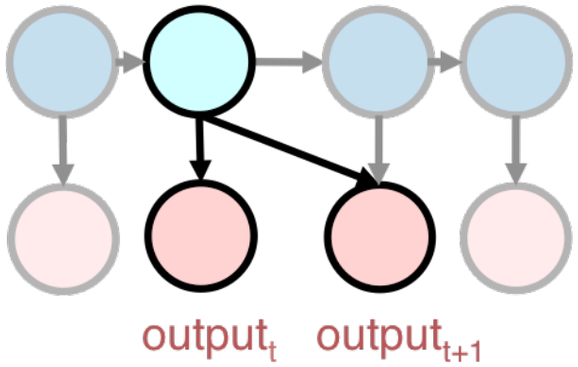


$$\Pr(\text{word}_1, \text{word}_2 | \text{topic} = T_1) = pp^\top$$

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

$\mathbb{B}$ 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

Sub-Optimal Attempt

θ  X

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

N i.i.d. samples

$\mathbb{B}$ is of rank 2: $\mathbb{B} = \rho\rho^{\top} + \Delta\Delta^{\top}$

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

MLE is non-convex optimization ☹

let's try something "spectral" ☺

2. Low rank matrix Sub-Optimal Attempt

θ ⇌ X

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

$\mathbb{B}$ is of rank 2: $\mathbb{B} = \rho\rho^\top + \Delta\Delta^\top$

N i.i.d. samples

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

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

- Set $\widehat{B}$ to be the rank 2 truncated SVD of $\frac{1}{N}B$
- 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

2. Low rank matrix

Main Result

- 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 $O(M^3)$

  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, (submitted to STOC 2016)

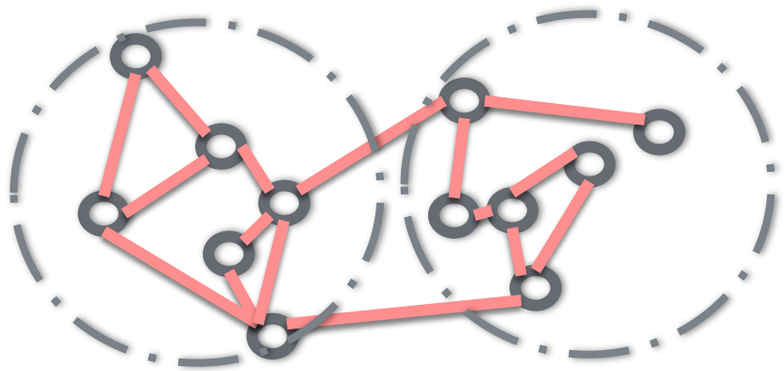
Algorithmic Idea

We capitalize the idea of community detection in sparse random network, which is a special case of our problem formulation

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

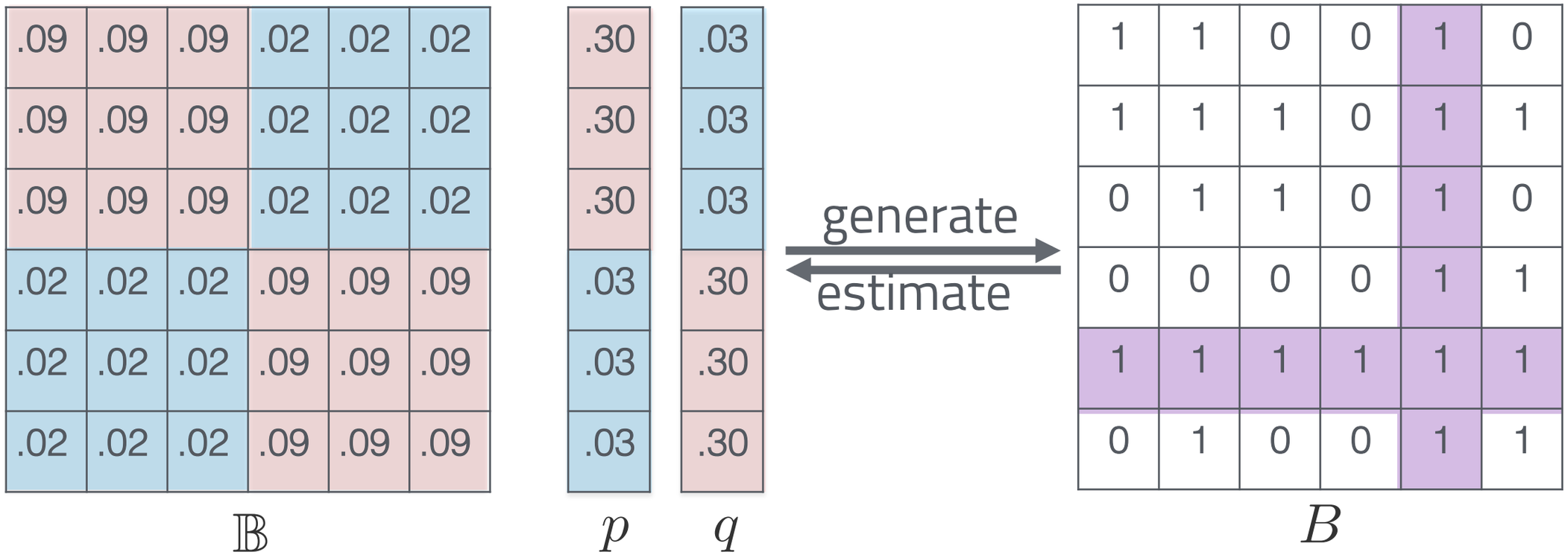
2. Low rank matrix Algorithmic Idea

We capitalize the idea of community detection in sparse random network, which is a special case of our problem formulation

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



2. Low rank matrix Algorithmic Idea

We capitalize the idea of community detection in sparse random network, which is a special case of our problem formulation

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	$\mathbb{B} = \rho\rho^\top + \Delta\Delta^\top$
	Sample counts	$B = \text{Poisson}(N\mathbb{B})$

Key Challenge:

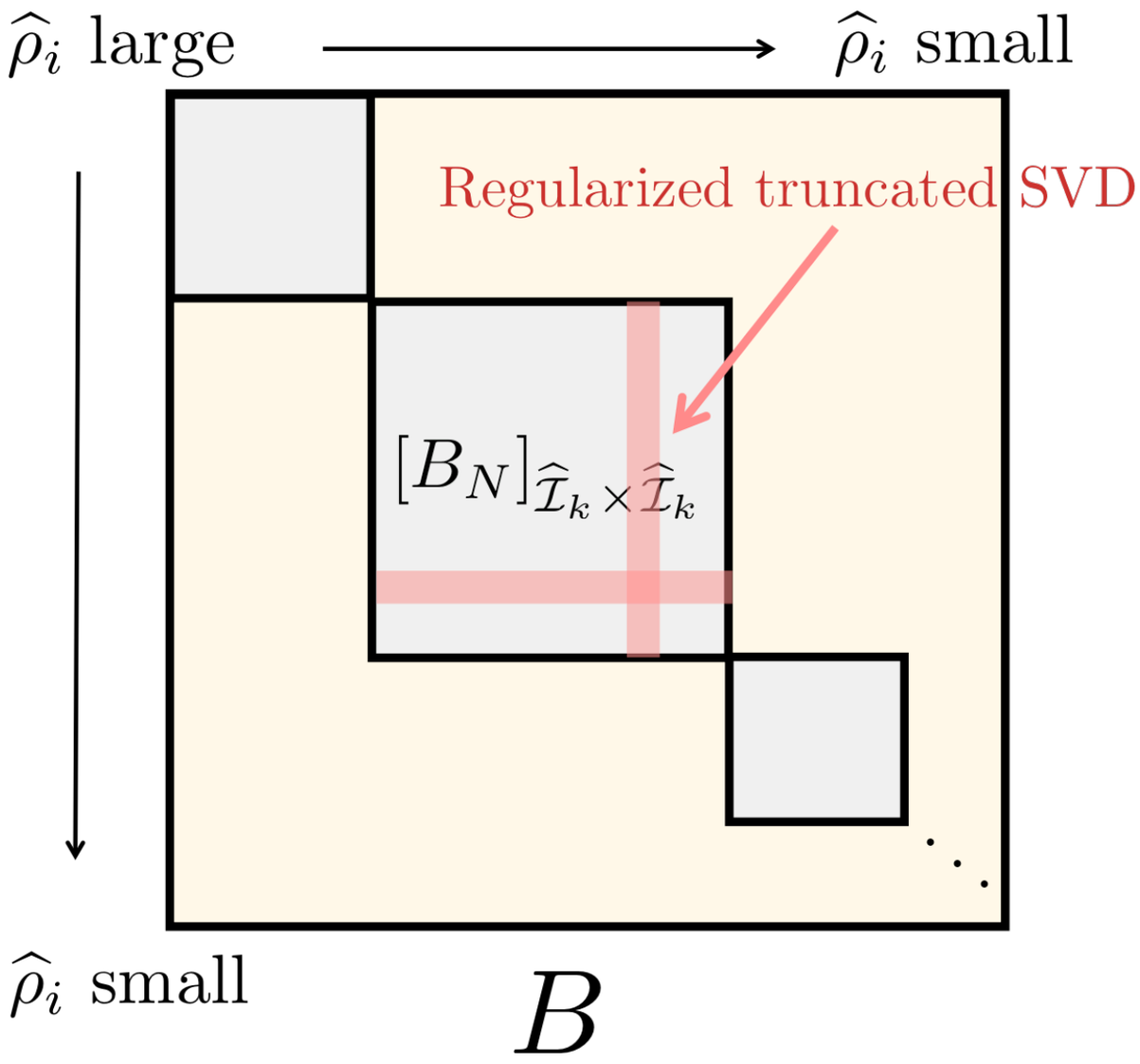
In our general setup, we do not have homogeneous marginal probabilities

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

We group words 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 words according to $\widehat{\rho}_i$
3. Regularized t-SVD in each bin × bin block of B to estimate



Key Challenges:

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

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

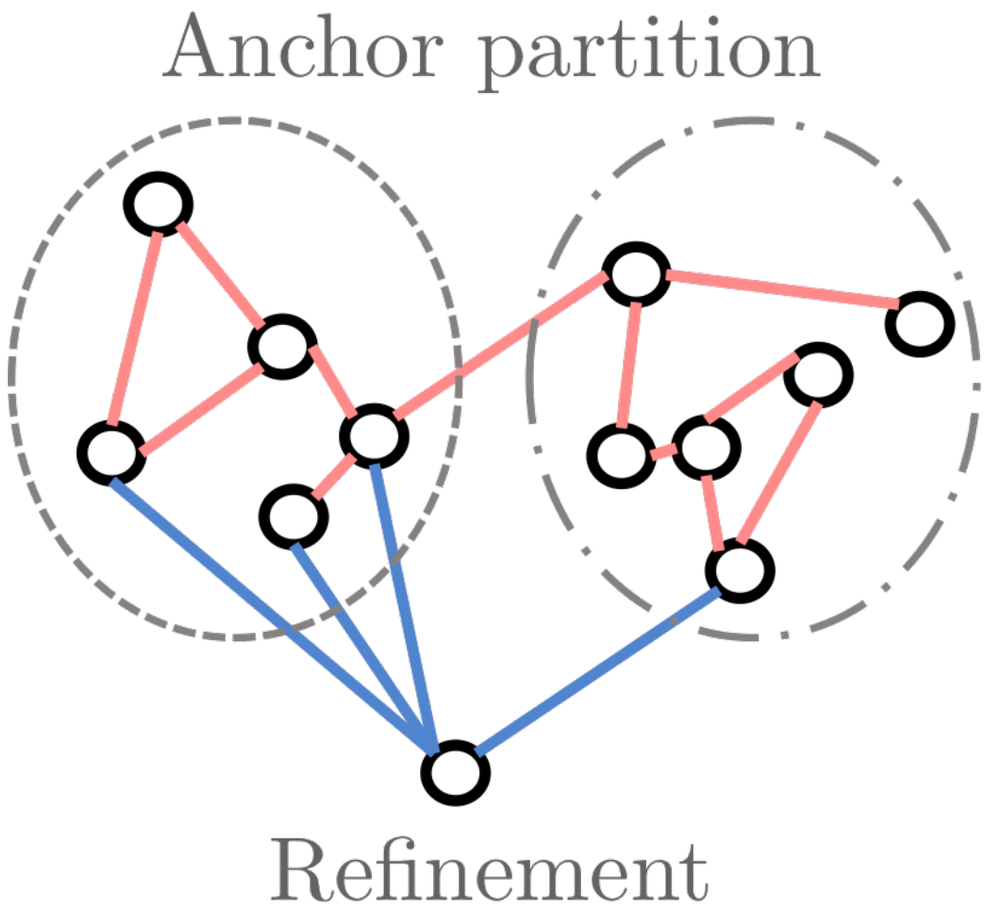
The coarse estimation from Phase 1 gives some global information
Make use of that to do local refinement for each row / column

Phase 2

1. Phase 1 gives coarse estimate for many words

2. Refine the estimate for each word use linear regression

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



Conclusion

- Spectral methods are powerful tools for learning mixture models. We can go beyond worst case analysis by exploiting the randomness in the analysis / algorithm.

- To make spectral algorithms more practical, one needs careful algorithm implementation to improve sample complexity.

Future works

- ✓ **Robustness:**
  Agnostic learning, generalization error analysis…
- ✓ **Dynamics:**
  Extend the analysis techniques and algorithmic ideas to learning of random processes, with streaming data, iterative algorithms…
- ✓ **Get hands dirty with real $X(\theta)$!**

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)

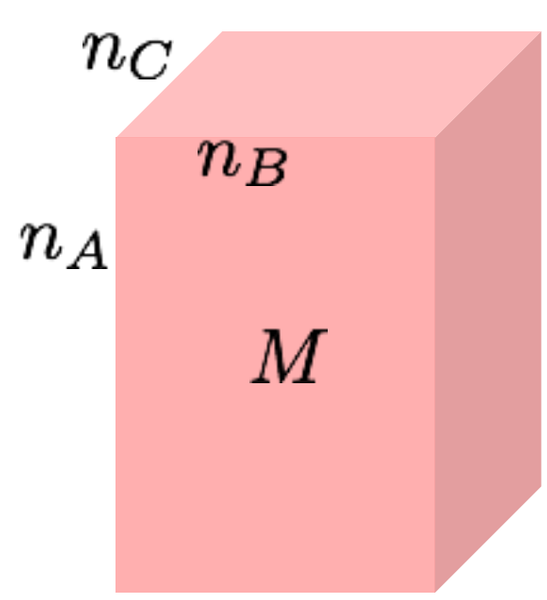
- "Recovering Structured Probability Matrices "
  H, S. Kakade, W. Kong, G. Valiant, (submitted to STOC 2016)

Thank you !

Question?

Tensor Decomposition

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



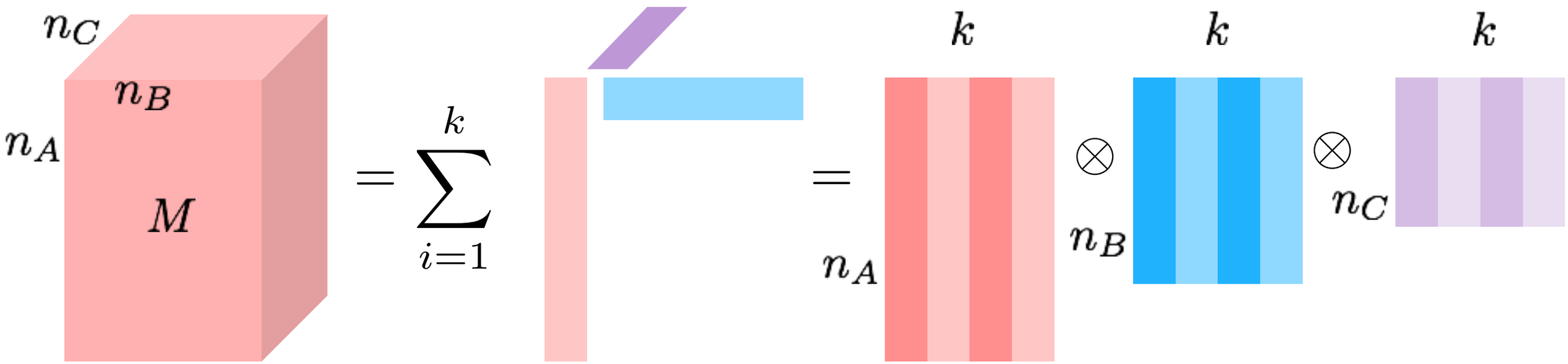
$$M_{j_1,j_2,j_3}, \quad j_1 \in [n_A], j_2 \in [n_B], j_3 \in [n_C]$$

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

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 of $A\,B\,C$

(the algorithm boils down to matrix SVD)