Learning Structured Probability Matrices

Qingqing Huang

2016 February

Massachusetts Institute of Technology

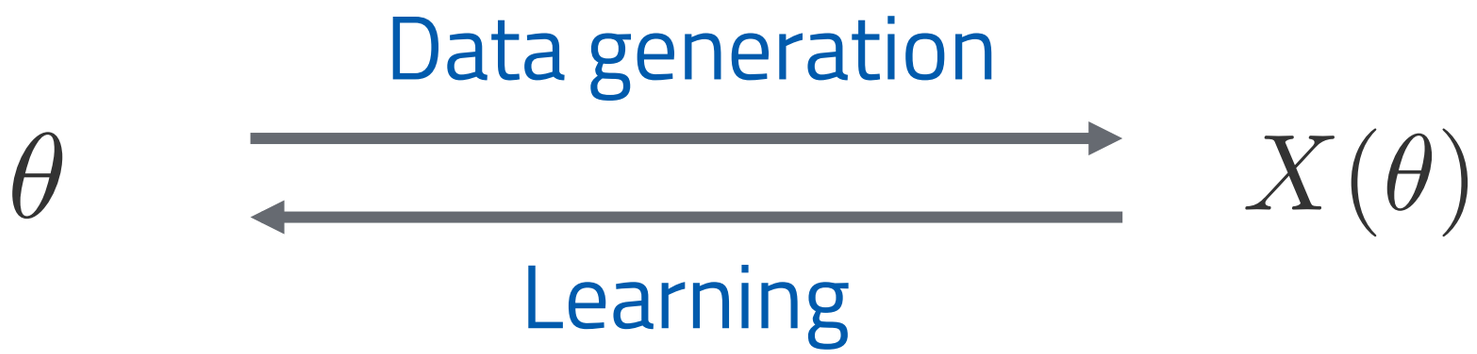
LIDS Laboratory for Information & Decision Systems

Based on joint work with Sham Kakade, Weihao Kong and Greg Valiant.

Learning

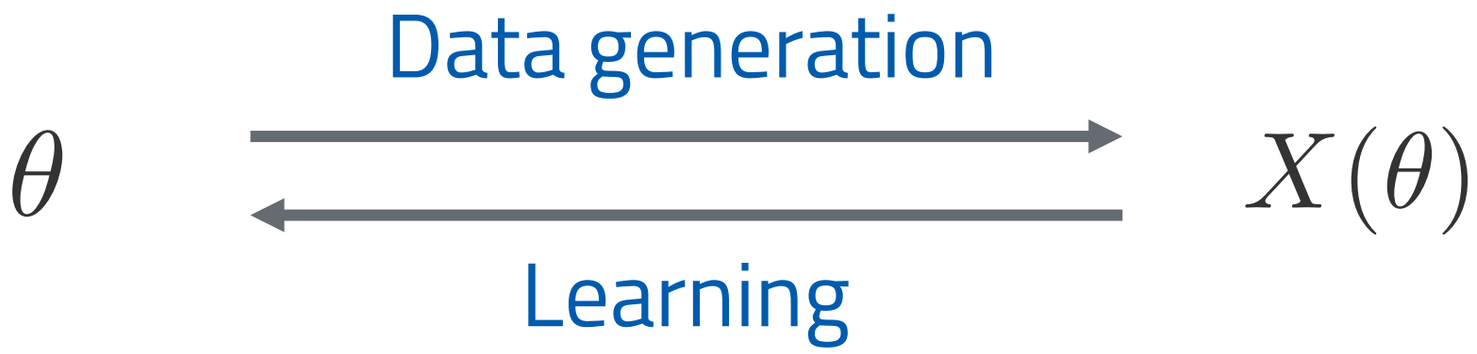
Data generation

Learning



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

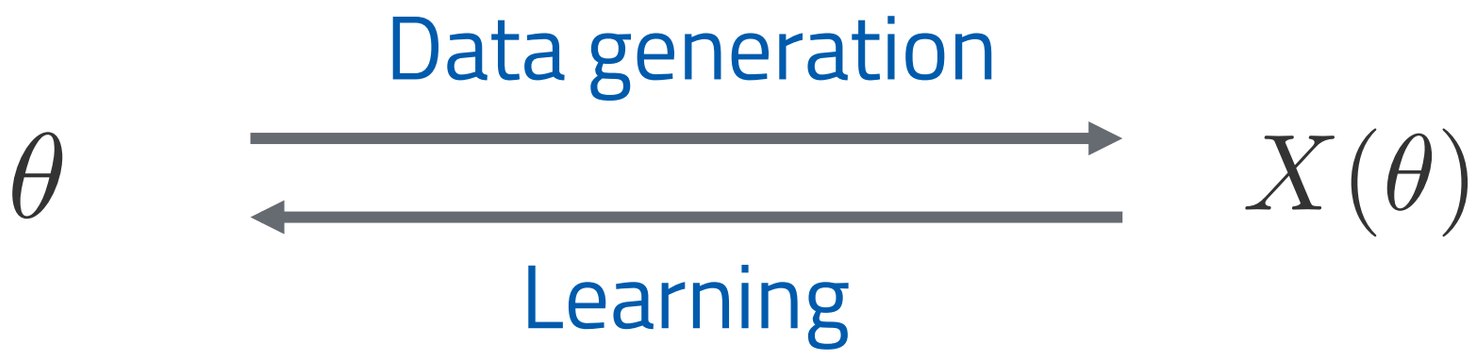
Learning



- 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

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 $\uparrow \dim(\theta)$

Unstructured Probability

θ  X

Discrete probability distribution (support over M outcomes)

N i.i.d. samples $N \sim \text{Poisson}$ (frequency counts over M outcomes)

p

$B = \text{Poisson}(Np)$

Unstructured Probability

θ ⇌ X

Discrete probability distribution (support over M outcomes)

N i.i.d. samples $N \sim \text{Poisson}$ (frequency counts over M outcomes)

p

.40
.20
.15
.15
.10

$M = 5$

$B = \text{Poisson}(Np)$

9
4
4
3
0

$N = 20$

Unstructured Probability

θ ⇌ X

Discrete probability distribution (support over M outcomes)

N i.i.d. samples $N \sim \text{Poisson}$ (frequency counts over M outcomes)

p

.40
.20
.15
.15
.10

$M = 5$

$B = \text{Poisson}(Np)$

9
4
4
3
0

$N = 20$

✔ **Goal:** find $\widehat{p}$ such that $\|\widehat{p} - p\|_1 \leq \epsilon$

N (↑ M) **sample complexity:** upper / lower bound

Structured Probability

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)

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

Structured Probability

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 **low rank** $\mathbb{B} = pp^\top + qq^\top$

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

Structured Probability

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 **low rank** $\mathbb{B} = pp^\top + qq^\top$

.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
.20
.15
.15
.10

p

.15
.40
.15
.10
.20

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

$N = 20$

Structured Probability

θ $\rightleftharpoons$ 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 **low rank** $\mathbb{B} = pp^\top + qq^\top$

.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
.20
.15
.15
.10

p

.15
.40
.15
.10
.20

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

$N = 20$

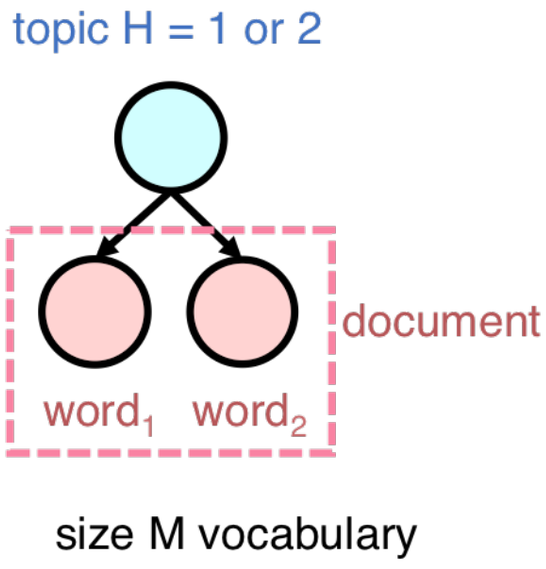
?

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

N (↑ M) **sample complexity:** upper / lower bound

Connection to Learning Mixture Models

Topic model



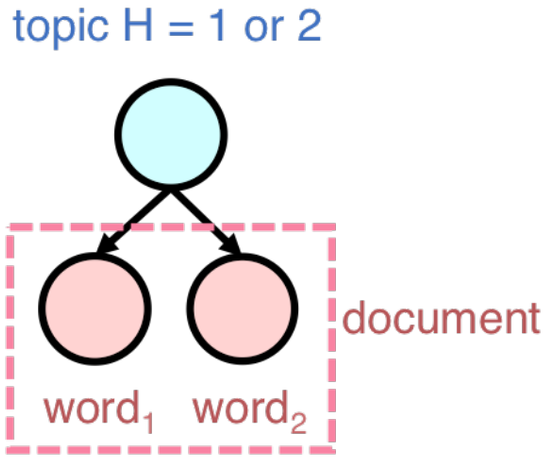
$$\Pr(\text{word}_1, \text{word}_2 | \text{topic} = T_1) = pp^\top$$

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

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

Connection to Learning Mixture Models

Topic model

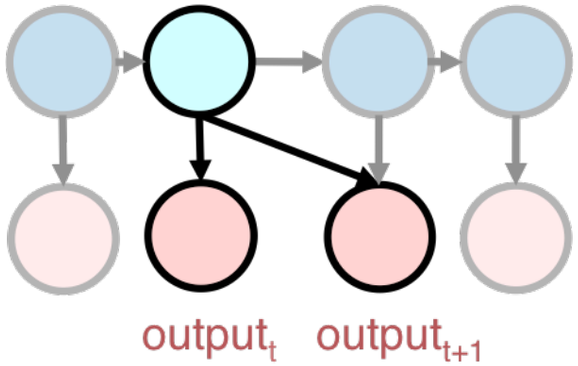


$$\Pr(\text{word}_1, \text{word}_2 | \text{topic} = T_1) = pp^\top$$

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

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

HMM

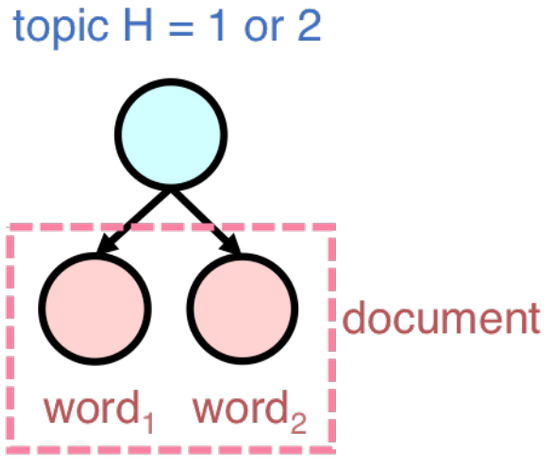


$$\Pr(\text{output}_1, \text{output}_2 | \text{state} = S_i) = O_i(OQ_i)^\top$$

$\mathbb{B}$ distribution of past and future outputs

Connection to Learning Mixture Models

Topic model

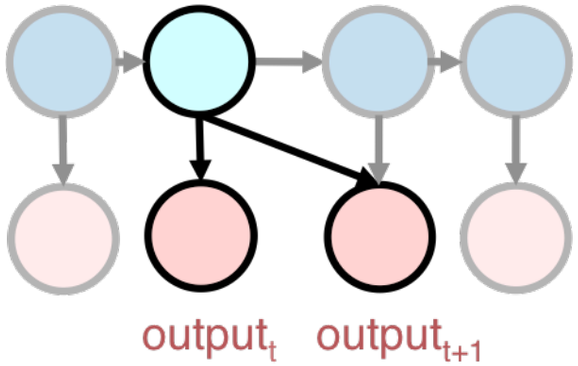


$$\Pr(\text{word}_1, \text{word}_2 | \text{topic} = T_1) = pp^\top$$

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

$\mathbb{B}$ 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 past and future outputs

Spectral Algorithm:

N data samples → empirical dsitribution B → find low rank $\widehat{B}$ close to $\mathbb{B}$ → Estimate model parameters

Structured Distribution Learning

θ  X

Discrete probability distribution

i.i.d. samples counts

Sample complexity	**Unstructured**	**Low rank structure**
Estimation	Linear	**?**
Property Testing	Sub-Linear	**?**

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

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

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

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

MLE is non-convex optimization ☹

let's try something intuitive ☺

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

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

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

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

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

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

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

- 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 $N = \Omega(M^2 \log M)$

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

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

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

- 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 $N = \Omega(M)$

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

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

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

- Set $\widehat{B}$ to be the rank 2 truncated SVD of $\frac{1}{N}B$
- To achieve accuracy $\|\widehat{B} - \mathbb{B}\|_1 \le \epsilon$ 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 Results

- 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 samples
  - ✓ Runtime $O(M^3)$

Main Results

- 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 samples
  - ✓ Runtime $O(M^3)$

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

Main Results

- 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 samples
  - ✓ Runtime $O(M^3)$

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

Sample complexity	**Unstructured**	**Low rank structure**
Estimation	Linear	Linear
Property Testing	Sub-Linear	Linear

Algorithmic Idea

We capitalize the idea of community detection in sparse random network, SBM is a special case of our problem formulation with homogeneous nodes

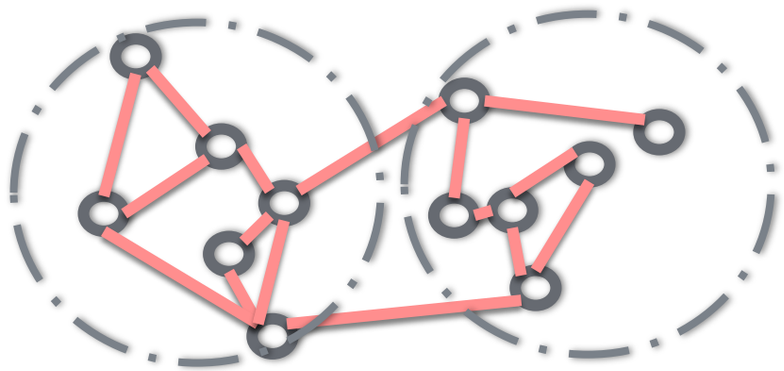
Algorithmic Idea

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

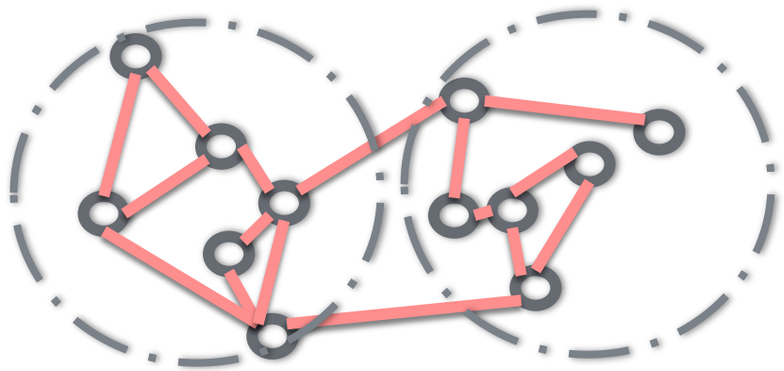
Algorithmic Idea

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

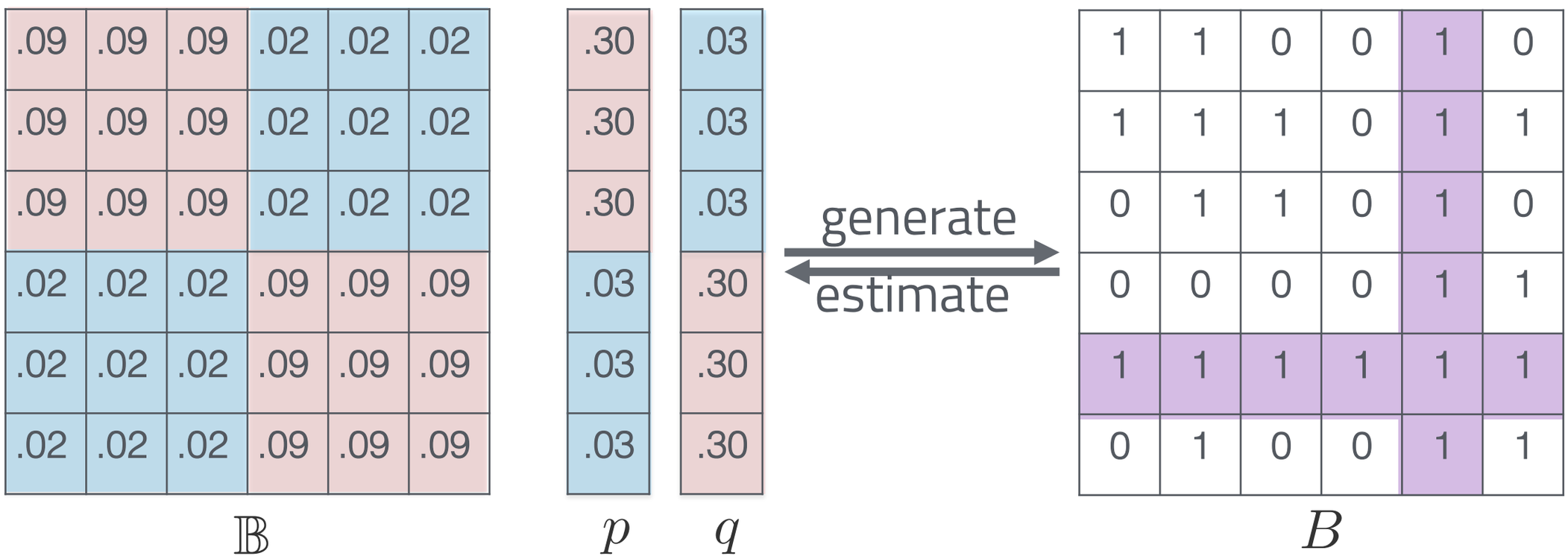
Algorithmic Idea

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



Algorithmic Idea

We capitalize the idea of community detection in sparse random network,
SBM is a special case of our problem 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	$\mathbb{B} = \rho\rho^\top + \Delta\Delta^\top$
	Sample counts	$B = \text{Poisson}(N\mathbb{B})$

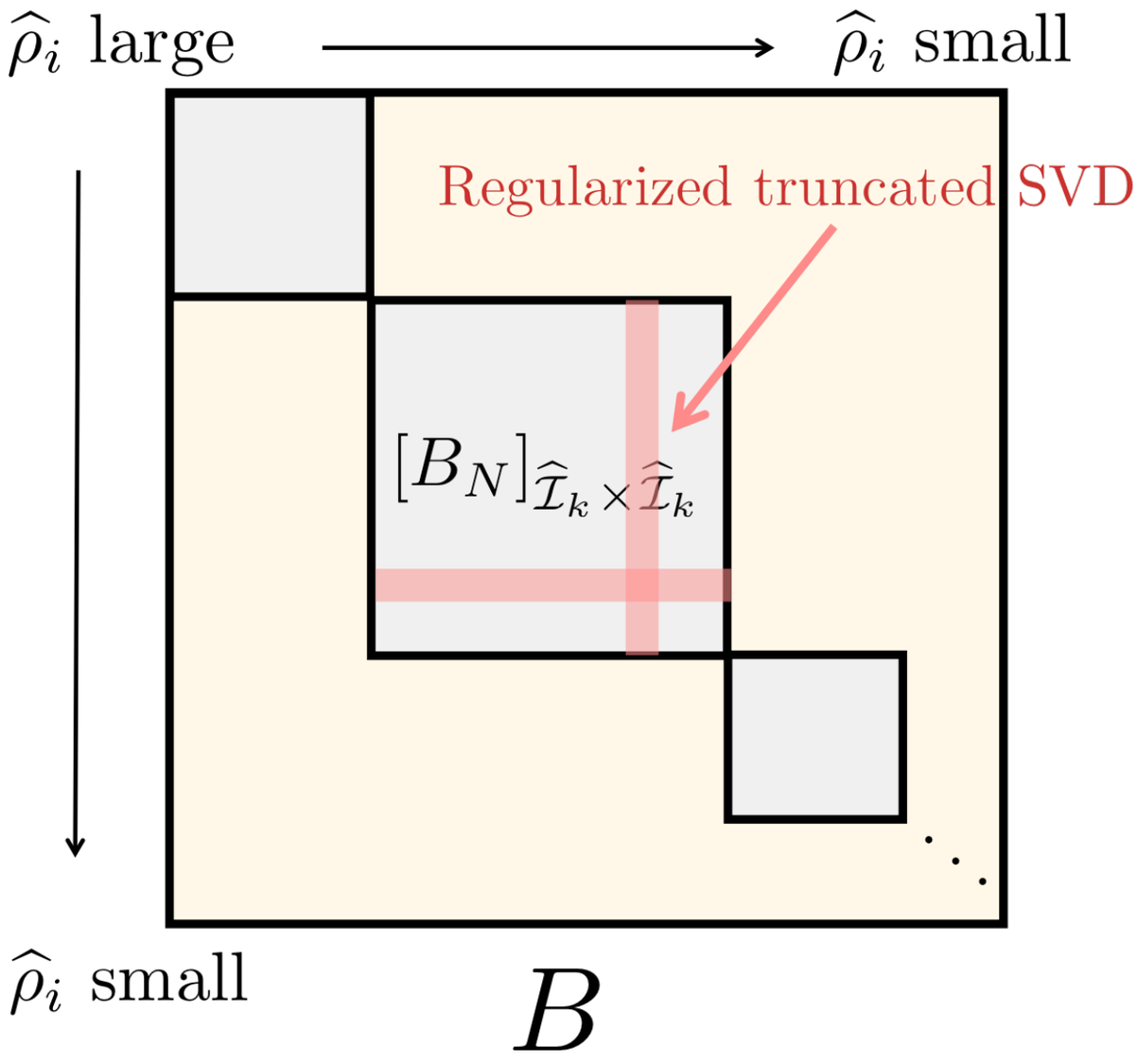
Key Challenge:
In our general setup, we have heterogeneous nodes/ marginal probabilities

Algorithmic Idea 1, **Binning**

We group the M nodes according to the empirical marginal probability, divide the matrix into 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

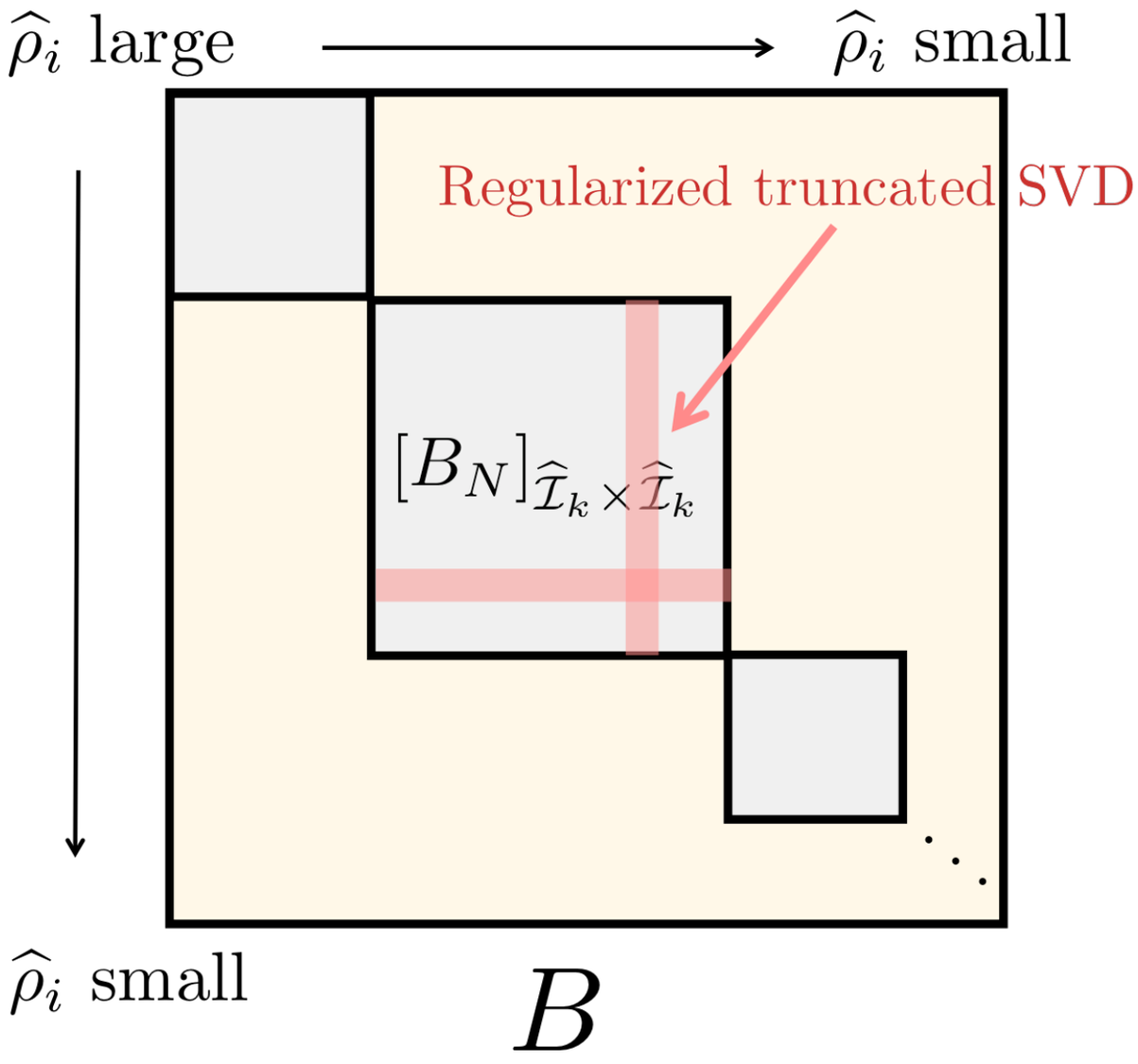


Algorithmic Idea 1, **Binning**

We group the M nodes according to the empirical marginal probability, divide the matrix into 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!

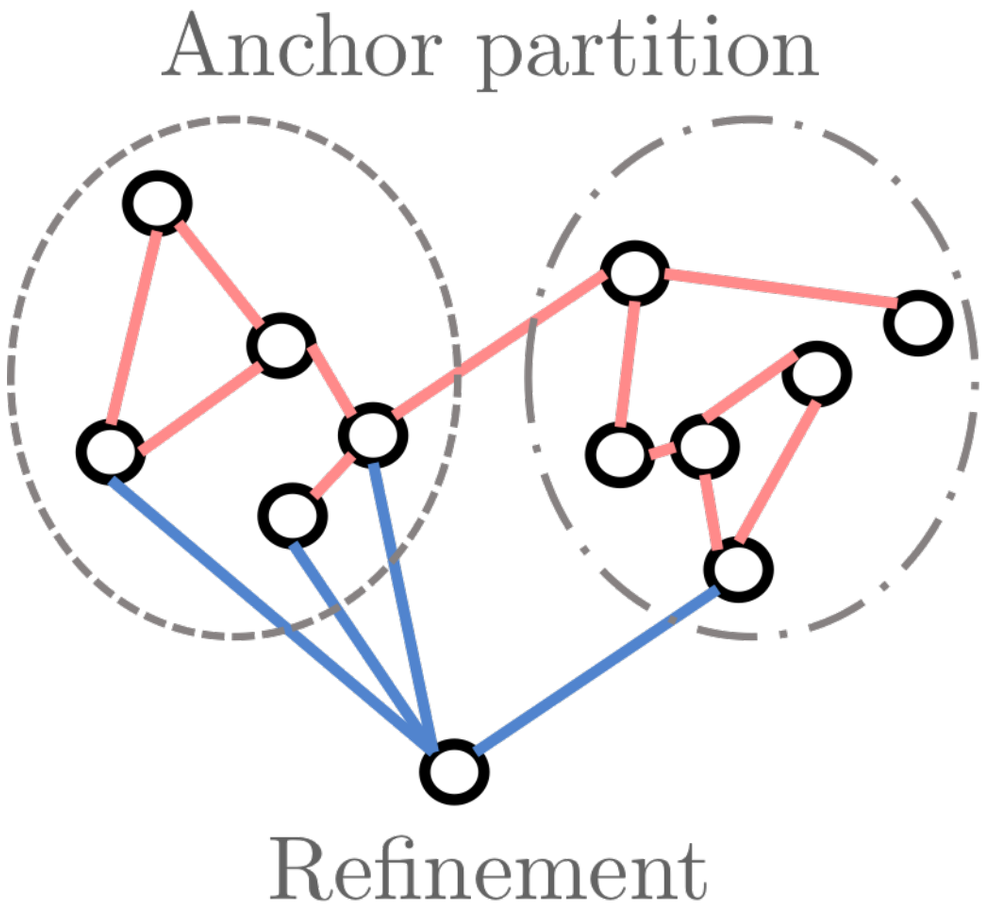
Algorithmic Idea 2, **Refinement**

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

Phase 2

1. Refine the estimate for each node
   use linear regression

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



Take-Away Message

- We identify a problem that lies at the core of many learning problems
- Spectral algorithm is not solving for the non-convex MLE, we need carefully designed algorithm to improve its statistical efficiency
- Coming soon: *estimation / approximation / property testing of structured probability distribution*