An iterative clustering algorithm for the Contextual Stochastic Block Model with optimality guarantees

Work conducted under the supervision of Christophe Biernacki and Hemant Tyagi

Guillaume Braun

ICML

July 2022, Baltimore

Motivation: clustering graphs with side information

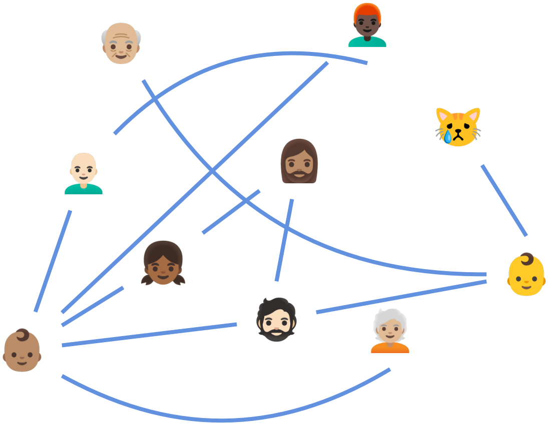

Figure: A social network with node features.

The Contextual Stochastic Block Model

- Adjacency matrix $A \in \{0,1\}^{n \times n} \sim SBM(n, K, \Pi)$ where:
  - n = number of nodes;
  - K = number of communities;
  - $Z \in \{0,1\}^{n \times K}$ partition matrix, $Z_{ik} = 1$ iff $i \in \mathcal{C}_k$;
  - $\Pi \in [0,1]^{n \times K}$ = connectivity matrix.
- Nodes features $(X_i)_{i \in [n]}$ generated independently of A and conditionally on Z by

$$X_i = \mu_{z_i} + \epsilon_i, \text{ where } \epsilon_i \overset{\text{ind.}}{\sim} \mathcal{N}(0, \sigma^2 I_d)$$

The Contextual Stochastic Block Model

- Adjacency matrix $A \in \{0,1\}^{n \times n} \sim SBM(n, K, \Pi)$ where:
  - n = number of nodes;
  - K = number of communities;
  - $Z \in \{0,1\}^{n \times K}$ partition matrix, $Z_{ik} = 1$ iff $i \in \mathcal{C}_k$;
  - $\Pi \in [0,1]^{n \times K}$= connectivity matrix.
- Nodes features $(X_i)_{i \in [n]}$ generated independently of A and conditionally on Z by

$$X_i = \mu_{z_i} + \epsilon_i, \text{ where } \epsilon_i \overset{\text{ind.}}{\sim} \mathcal{N}(0, \sigma^2 I_d)$$

The Contextual Stochastic Block Model

- Adjacency matrix $A \in \{0,1\}^{n \times n} \sim SBM(n, K, \Pi)$ where:
  - n = number of nodes;
  - K = number of communities;
  - $Z \in \{0,1\}^{n \times K}$ partition matrix, $Z_{ik} = 1$ iff $i \in \mathcal{C}_k$;
  - $\Pi \in [0,1]^{n \times K}$= connectivity matrix.
- Nodes features $(X_i)_{i \in [n]}$ generated independently of A and conditionally on Z by

$$X_i = \mu_{z_i} + \epsilon_i, \text{ where } \epsilon_i \overset{\text{ind.}}{\sim} \mathcal{N}(0, \sigma^2 I_d)$$

The Contextual Stochastic Block Model

- Adjacency matrix $A \in \{0,1\}^{n \times n} \sim SBM(n, K, \Pi)$ where:
  - n = number of nodes;
  - K = number of communities;
  - $Z \in \{0,1\}^{n \times K}$ partition matrix, $Z_{ik} = 1$ iff $i \in \mathcal{C}_k$;
  - $\Pi \in [0,1]^{n \times K}$= connectivity matrix.
- Nodes features $(X_i)_{i \in [n]}$ generated independently of A and conditionally on Z by

$$X_i = \mu_{z_i} + \epsilon_i, \text{ where } \epsilon_i \overset{\text{ind.}}{\sim} \mathcal{N}(0, \sigma^2 I_d)$$

The Contextual Stochastic Block Model

- Adjacency matrix $A \in \{0,1\}^{n\times n} \sim SBM(n, K, \Pi)$ where:
  - n = number of nodes;
  - K = number of communities;
  - $Z \in \{0,1\}^{n\times K}$ partition matrix, $Z_{ik} = 1$ iff $i \in \mathcal{C}_k$;
  - $\Pi \in [0,1]^{n\times K}$= connectivity matrix.
- Nodes features $(X_i)_{i\in[n]}$ generated independently of A and conditionally on Z by

$$X_i = \mu_{z_i} + \epsilon_i, \text{ where } \epsilon_i \overset{\text{ind.}}{\sim} \mathcal{N}(0, \sigma^2 I_d)$$

The Contextual Stochastic Block Model

- Adjacency matrix $A \in \{0,1\}^{n\times n} \sim SBM(n, K, \Pi)$ where:
  - n = number of nodes;
  - K = number of communities;
  - $Z \in \{0,1\}^{n\times K}$ partition matrix, $Z_{ik} = 1$ iff $i \in \mathcal{C}_k$;
  - $\Pi \in [0,1]^{n\times K}$= connectivity matrix.
- Nodes features $(X_i)_{i\in[n]}$ generated independently of A and conditionally on Z by

$$X_i = \mu_{z_i} + \epsilon_i, \text{ where } \epsilon_i \overset{\text{ind.}}{\sim} \mathcal{N}(0, \sigma^2 I_d)$$

The algorithm principle

1. Use a spectral method (random initialization sometimes also works) to get a first estimate $Z^{(0)}$ of Z.
2. For $t \leq T$ do:
   - Estimate the model parameters.
   - Refine the partition by solving a least square optimization problem that approximate the MAP for each node i.
3. Output the final partition $Z^{(T)}$

Advantages : the algorithm is fast and statistically optimal.

The algorithm principle

1. Use a spectral method (random initialization sometimes also works) to get a first estimate $Z^{(0)}$ of Z.
2. For $t \leq T$ do:
   - Estimate the model parameters.
   - Refine the partition by solving a least square optimization problem that approximate the MAP for each node i.
3. Output the final partition $Z^{(T)}$

Advantages : the algorithm is fast and statistically optimal.

The algorithm principle

1. Use a spectral method (random initialization sometimes also works) to get a first estimate $Z^{(0)}$ of Z.
2. For $t \leq T$ do:
   - Estimate the model parameters.
   - Refine the partition by solving a least square optimization problem that approximate the MAP for each node i.
3. Output the final partition $Z^{(T)}$

Advantages : the algorithm is fast and statistically optimal.

The algorithm principle

1. Use a spectral method (random initialization sometimes also works) to get a first estimate $Z^{(0)}$ of Z.
2. For $t \leq T$ do:
   - Estimate the model parameters.
   - Refine the partition by solving a least square optimization problem that approximate the MAP for each node i.
3. Output the final partition $Z^{(T)}$

Advantages : the algorithm is fast and statistically optimal.

The algorithm principle

1. Use a spectral method (random initialization sometimes also works) to get a first estimate $Z^{(0)}$ of Z.
2. For $t \leq T$ do:
   - Estimate the model parameters.
   - Refine the partition by solving a least square optimization problem that approximate the MAP for each node i.
3. Output the final partition $Z^{(T)}$

Advantages : the algorithm is fast and statistically optimal.

The algorithm principle

1. Use a spectral method (random initialization sometimes also works) to get a first estimate $Z^{(0)}$ of Z.
2. For $t \leq T$ do:
   - Estimate the model parameters.
   - Refine the partition by solving a least square optimization problem that approximate the MAP for each node i.
3. Output the final partition $Z^{(T)}$

Advantages : the algorithm is fast and statistically optimal.

The algorithm principle

1. Use a spectral method (random initialization sometimes also works) to get a first estimate $Z^{(0)}$ of Z.
2. For $t \leq T$ do:
   - Estimate the model parameters.
   - Refine the partition by solving a least square optimization problem that approximate the MAP for each node i.
3. Output the final partition $Z^{(T)}$

Advantages : the algorithm is fast and statistically optimal.

Main results

Theorem

The misclustering rate $r(Z^{(T)}, Z)$ satisfies

$$r(Z^{(T)}, Z) \lesssim e^{-(SNR_1 + SNR_2)}$$

where SNR_1 is the graph Signal-To-Noise Ratio (SNR) and SNR_2 is the features SNR.

- The rate of convergence is minimax optimal.
- In practice it also works on **weighted graphs** and one can sometimes use a **random initialization**.

Main results

Theorem

The misclustering rate $r(Z^{(T)}, Z)$ satisfies

$$r(Z^{(T)}, Z) \lesssim e^{-(SNR_1 + SNR_2)}$$

where SNR_1 is the graph Signal-To-Noise Ratio (SNR) and SNR_2 is the features SNR.

- The rate of convergence is minimax optimal.
- In practice it also works on **weighted graphs** and one can sometimes use a **random initialization**.