Streaming Belief Propagation for Community Detection

Yuchen Wu

Department of Statistics, Stanford University

18th October 2021

Jakab Tardos

MohammadHossein Bateni

André Linhares

Filipe Miguel Gonçalves de Almeida

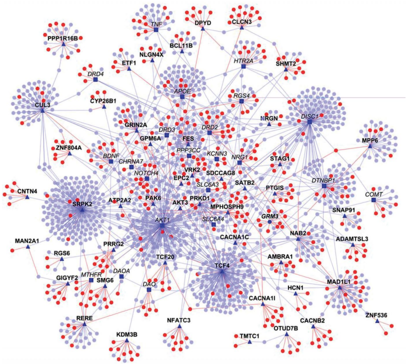
Andrea Montanari

Ashkan Norouzi-Fard

Community detection

Data: $G = (V, E)$.

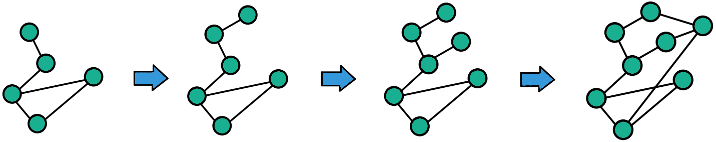
Social networks.



Protein interaction.

Dynamic networks

- Network structure evolves over time



- Streaming local algorithms:
  1. Understand the fundamental statistical limits
  2. Design efficient streaming local algorithms

Streaming stochastic block model (StSBM)

Streaming stochastic block model (StSBM)

- StSBM$(n, k, p, W, \alpha \leq \frac{k}{k-1})$

Streaming stochastic block model (StSBM)

- StSBM$(n, k, p, W, \alpha \leq \frac{k}{k-1})$
- For $v \in V$,

$$\mathbb{P}(\tau(v) = s) = p_s, \qquad \mathbb{P}(\tilde{\tau}(v) = s_1 \mid \tau(v) = s_0) = \begin{cases} 1 - \alpha & \text{if } s_1 = s_0, \\ \alpha/(k-1) & \text{if } s_1 \neq s_0. \end{cases}$$

Streaming stochastic block model (StSBM)

- StSBM$(n, k, p, W, \alpha \leq \frac{k}{k-1})$
- For $v \in V$,

$$\mathbb{P}(\tau(v) = s) = p_s, \qquad \mathbb{P}(\tilde{\tau}(v) = s_1 \mid \tau(v) = s_0) = \begin{cases} 1 - \alpha & \text{if } s_1 = s_0, \\ \alpha/(k-1) & \text{if } s_1 \neq s_0. \end{cases}$$

- Conditioning on $(\tau, \tilde{\tau})$, for all $u, v \in V$,

$$\mathbb{P}((u, v) \in E \mid \tau, \tilde{\tau}) = W_{\tau(u), \tau(v)}.$$

Streaming stochastic block model (StSBM)

- StSBM$(n, k, p, W, \alpha \leq \frac{k}{k-1})$
- For $v \in V$,

$$\mathbb{P}(\tau(v) = s) = p_s, \qquad \mathbb{P}(\tilde{\tau}(v) = s_1 \mid \tau(v) = s_0) = \begin{cases} 1 - \alpha & \text{if } s_1 = s_0, \\ \alpha/(k-1) & \text{if } s_1 \neq s_0. \end{cases}$$

- Conditioning on $(\tau, \tilde{\tau})$, for all $u, v \in V$,

$$\mathbb{P}((u, v) \in E \mid \tau, \tilde{\tau}) = W_{\tau(u), \tau(v)}.$$

- At time t reveal $v(t)$, $G(t) = (V(t), E(t))$, $V(t) = \{v(1), \cdots, v(t)\}$, $E(t) = \{(u, v) \in E, u, v \in V(t)\}$

- StSBM$(n, k, p, W, \alpha \leq \frac{k}{k-1})$
- For $v \in V$,

$$\mathbb{P}(\tau(v) = s) = p_s, \qquad \mathbb{P}(\tilde{\tau}(v) = s_1 \mid \tau(v) = s_0) = \begin{cases} 1 - \alpha & \text{if } s_1 = s_0, \\ \alpha/(k-1) & \text{if } s_1 \neq s_0. \end{cases}$$

- Conditioning on $(\tau, \tilde{\tau})$, for all $u, v \in V$,

$$\mathbb{P}((u, v) \in E \mid \tau, \tilde{\tau}) = W_{\tau(u),\tau(v)}.$$

- At time t reveal $v(t)$, $G(t) = (V(t), E(t))$, $V(t) = \{v(1), \cdots, v(t)\}$, $E(t) = \{(u, v) \in E, u, v \in V(t)\}$
- StSSBM(n, k, a, b, α): $p = (1/k, \cdots, 1/k)$, W having diagonal elements a/n and non-diagonal elements b/n.

Local streaming algorithm

Local streaming algorithm

- $B_R^t(v) = (V_R^t(v), E_R^t(v))$

Local streaming algorithm

- $B_R^t(v) = (V_R^t(v), E_R^t(v))$
- $\mathcal{G}_v^t = (\mathcal{V}_v^t, \mathcal{E}_v^t)$: subgraph accessible to node v at time t, $\mathcal{G}_v^0 = (\{v\}, \emptyset)$

Local streaming algorithm

- $B_R^t(v) = (V_R^t(v), E_R^t(v))$
- $\mathcal{G}_v^t = (\mathcal{V}_v^t, \mathcal{E}_v^t)$: subgraph accessible to node v at time t, $\mathcal{G}_v^0 = (\{v\}, \emptyset)$
- At time t

$$\mathcal{V}_v^t = \begin{cases} \cup_{v' \in V_R^t(v(t))} \mathcal{V}_{v'}^{t-1} & \text{for } v \in V_R^t(v(t)), \\ \mathcal{V}_v^{t-1} & \text{for } v \notin V_R^t(v(t)). \end{cases}$$

Local streaming algorithm

- $B_R^t(v) = (V_R^t(v), E_R^t(v))$
- $\mathcal{G}_v^t = (\mathcal{V}_v^t, \mathcal{E}_v^t)$: subgraph accessible to node v at time t, $\mathcal{G}_v^0 = (\{v\}, \emptyset)$
- At time t

$$\mathcal{V}_v^t = \begin{cases} \cup_{v' \in V_R^t(v(t))} \mathcal{V}_{v'}^{t-1} & \text{for } v \in V_R^t(v(t)), \\ \mathcal{V}_v^{t-1} & \text{for } v \notin V_R^t(v(t)). \end{cases}$$

- At time t, output cluster estimate $\hat{\tau}(v)$ based on $\mathcal{V}_v^t$

Local streaming algorithm

- $B_R^t(v) = (V_R^t(v), E_R^t(v))$
- $\mathcal{G}_v^t = (\mathcal{V}_v^t, \mathcal{E}_v^t)$: subgraph accessible to node v at time t, $\mathcal{G}_v^0 = (\{v\}, \emptyset)$
- At time t

$$\mathcal{V}_v^t = \begin{cases} \cup_{v' \in V_R^t(v(t))} \mathcal{V}_{v'}^{t-1} & \text{for } v \in V_R^t(v(t)), \\ \mathcal{V}_v^{t-1} & \text{for } v \notin V_R^t(v(t)). \end{cases}$$

- At time t, output cluster estimate $\hat{\tau}(v)$ based on $\mathcal{V}_v^t$

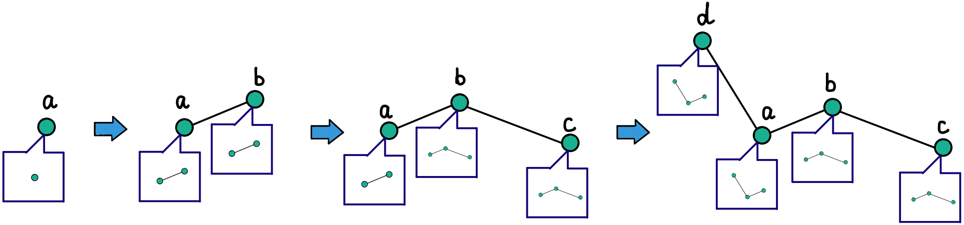


Illustration of a local streaming algorithm with $R = 1$.

Fundamental limit of local streaming algorithm

For a clustering algorithm $\mathcal{A}$, define its estimation accuracy as

$$Q_n(\mathcal{A}) = \mathbb{E}\left[\max_{\pi \in \mathfrak{S}_k} \frac{1}{n} \sum_{v \in V} \mathbb{1}\{\mathcal{A}(v) = \pi \circ \tau(v)\}\right].$$

$\mathfrak{S}_k$: set of permutations over $\{1, 2, \cdots, k\}$.

Theorem 1

Under StSSBM(n, k, a, b) with no side information, no R-local streaming algorithm $\mathcal{A}$ can achieve non-trivial estimation accuracy. That is, $\lim_{n \to \infty} Q_n(\mathcal{A}) = 1/k$.

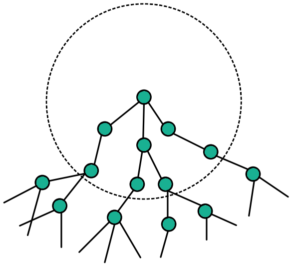
Proof idea of Theorem 1

Step 1: local streaming algorithms are essentially local algorithms

For all $\epsilon > 0$, there exists $R_\epsilon \in \mathbb{N}^+$, independent of n, such that

$$\liminf_{n\to\infty} \mathbb{P}(\mathcal{V}_v^n \in V_{R_\epsilon}^n(v)) \geq 1 - \epsilon.$$

Step 2: bounded radius neighborhood is a Galton-Watson branching tree



any bounded radius neighborhood can be coupled with a Galton-Watson branching tree.

Streaming belief propagation

- StSSBM(n, k, a, b, α), $\alpha < (k-1)/k$

Streaming belief propagation

- StSSBM(n, k, a, b, α), $\alpha < (k-1)/k$
- $(u, v) \in E$, $\mathbf{m}_{u \to v}$, $\mathbf{m}_{v \to u}$

$$\mathbf{m}_{u \to v} = (m_{u \to v}(1), m_{u \to v}(2), \cdots, m_{u \to v}(k)) \in \Delta_k.$$

Streaming belief propagation

- StSSBM(n, k, a, b, α), $\alpha < (k-1)/k$
- $(u, v) \in E$, $\mathbf{m}_{u\to v}$, $\mathbf{m}_{v\to u}$

$$\mathbf{m}_{u\to v} = (m_{u\to v}(1), m_{u\to v}(2), \cdots, m_{u\to v}(k)) \in \Delta_k.$$

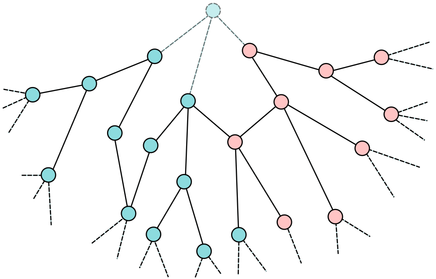
- $\mathbf{BP}_0(\tilde{\tau})(s) = (\alpha + (k-1-k\alpha)\mathbb{1}_{\tilde{\tau}=s})/(k-1)$

$$\mathbf{BP}(\{\mathbf{m}_i\}_{i\le l}; \tilde{\tau})(s) = \frac{\mathbf{BP}_0(\tilde{\tau})(s)}{Z}\prod_{i=1}^{l}(b + (a-b)m_i(s)).$$

Streaming belief propagation

Algorithm 1 Streaming R-local belief propagation

for $t = 1, 2, \ldots, n$ **do**
  // Update the incoming messages:
  for $w \in D_1^t(v(t))$ **do**
    $\boldsymbol{m}_{w \to v(t)} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to w} : u \in \partial w \setminus \{v(t)\}\}; \tilde{\tau}(w))$
  end for
  // Update the outgoing messages:
  for $r = 1, 2, \ldots, R$ **do**
    for $v \in D_r^t(v(t))$ **do**
      Let $v' \in D_1^t(v)$ on a shortest path connecting v and $v(t)$.
      $\boldsymbol{m}_{v' \to v} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to v'} : u \in \partial v' \setminus \{v\}\}; \tilde{\tau}(v'))$
    end for
  end for
end for
for $u \in V$ **do**
  $\boldsymbol{m}_u \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{v \to u}\}_{v \in \partial u}; \tilde{\tau}(u))$
  Output $\hat{\tau}(u) := \arg\max_{s \in [k]} m_u(s)$ as an estimate for $\tau(u)$.
end for



Streaming belief propagation

Algorithm 1 Streaming R-local belief propagation

for $t = 1, 2, \ldots, n$ **do**

    // Update the incoming messages:

    for $w \in D_1^t(v(t))$ **do**

        $\boldsymbol{m}_{w \to v(t)} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to w} : u \in \partial w \setminus \{v(t)\}\}; \tilde{\tau}(w))$

    end for

    // Update the outgoing messages:

    for $r = 1, 2, \ldots, R$ **do**

        for $v \in D_r^t(v(t))$ **do**

            Let $v' \in D_1^t(v)$ on a shortest path connecting v and $v(t)$.

            $\boldsymbol{m}_{v' \to v} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to v'} : u \in \partial v' \setminus \{v\}\}; \tilde{\tau}(v'))$

        end for

    end for

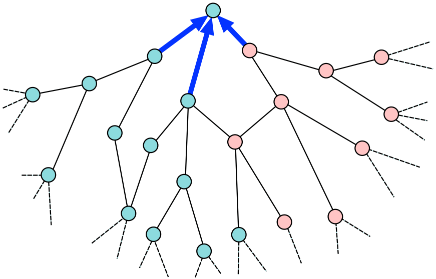
end for

for $u \in V$ **do**

    $\boldsymbol{m}_u \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{v \to u}\}_{v \in \partial u}; \tilde{\tau}(u))$

    Output $\hat{\tau}(u) := \arg\max_{s \in [k]} m_u(s)$ as an estimate for $\tau(u)$.

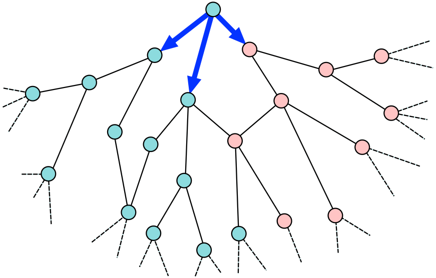
end for



Streaming belief propagation

Algorithm 1 Streaming R-local belief propagation

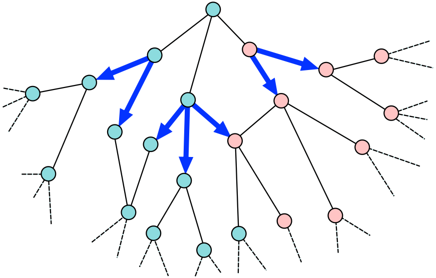
for $t = 1, 2, \ldots, n$ **do**
    // Update the incoming messages:
    for $w \in D_1^t(v(t))$ **do**
        $\boldsymbol{m}_{w \to v(t)} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to w} : u \in \partial w \setminus \{v(t)\}\}; \tilde{\tau}(w))$
    end for
    // Update the outgoing messages:
    for $r = 1, 2, \ldots, R$ **do**
        for $v \in D_r^t(v(t))$ **do**
            Let $v' \in D_1^t(v)$ on a shortest path connecting v and $v(t)$.
            $\boldsymbol{m}_{v' \to v} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to v'} : u \in \partial v' \setminus \{v\}\}; \tilde{\tau}(v'))$
        end for
    end for
end for
for $u \in V$ **do**
    $\boldsymbol{m}_u \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{v \to u}\}_{v \in \partial u}; \tilde{\tau}(u))$
    Output $\hat{\tau}(u) := \arg\max_{s \in [k]} m_u(s)$ as an estimate for $\tau(u)$.
end for



Streaming belief propagation

Algorithm 1 Streaming R-local belief propagation

for $t = 1, 2, \ldots, n$ **do**
  // Update the incoming messages:
  for $w \in D_1^t(v(t))$ **do**
    $\boldsymbol{m}_{w \to v(t)} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to w} : u \in \partial w \setminus \{v(t)\}\}; \tilde{\tau}(w))$
  end for
  // Update the outgoing messages:
  for $r = 1, 2, \ldots, R$ **do**
    for $v \in D_r^t(v(t))$ **do**
      Let $v' \in D_1^t(v)$ on a shortest path connecting v and $v(t)$.
      $\boldsymbol{m}_{v' \to v} \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{u \to v'} : u \in \partial v' \setminus \{v\}\}; \tilde{\tau}(v'))$
    end for
  end for
end for
for $u \in V$ **do**
  $\boldsymbol{m}_u \leftarrow \mathrm{BP}(\{\boldsymbol{m}_{v \to u}\}_{v \in \partial u}; \tilde{\tau}(u))$
  Output $\hat{\tau}(u) := \arg\max_{s \in [k]} m_u(s)$ as an estimate for $\tau(u)$.
end for



Streaming belief propagation

Theorem 2

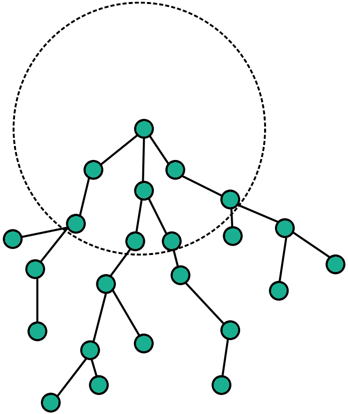
Suppose one of the following conditions hold (for a sufficiently large absolute constant C) in the two-group symmetric model StSSBM$(n, k = 2, a, b, \alpha)$:

1. $|a - b| < 2$ *and* $\alpha \in (0, 1/2)$;
2. $(a - b)^2 > C(a + b)$ *and* $\alpha \in (0, 1/2)$;
3. $\alpha \in (0, 1/C)$.

Then R-local streaming belief propagation achieves optimal estimation accuracy:

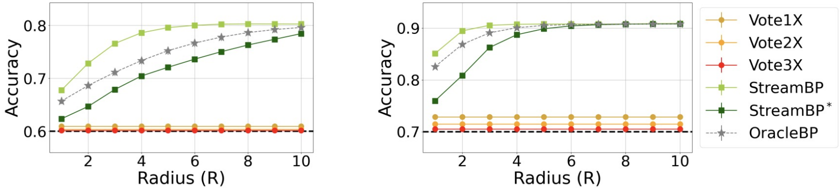
$$\limsup_{R\to\infty} \limsup_{n\to\infty} \left(\sup_{\mathcal{A}} Q_n(\mathcal{A}) - Q_n(\mathcal{A}_R) \right) = 0.$$

Proof idea of Theorem 2

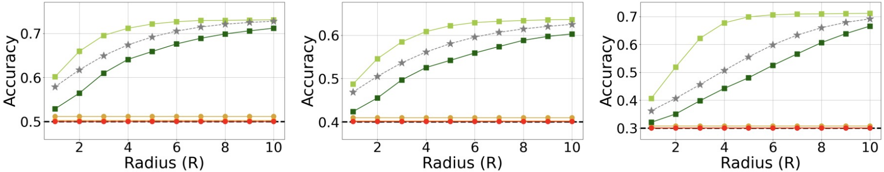


$\mathcal{G}_v^t,\ R = 2$

Simulation

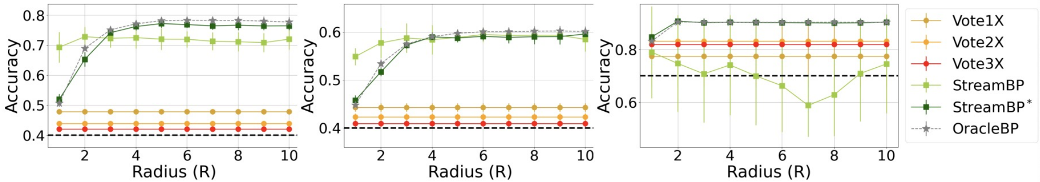


(a) $k = 2, a = 3, b = .1, \alpha = .4.$ (b) $k = 2, a = 5, b = .5, \alpha = .3.$



(c) $k = 3, a = 4, b = .05, \alpha = .5.$ (d) $k = 4, a = 3, b = .1, \alpha = .6.$ (e) $k = 5, a = 8, b = .1, \alpha = .7.$

Simulation



(a) Cora ($k = 7$), $\alpha = 0.6$. (b) Citeseer ($k = 6$), $\alpha = 0.6$. (c) Polblogs ($k = 2$), $\alpha = 0.3$.

Thank You!