Streaming belief propagation for community detection

Yuchen Wu, Jakab Tardos, MohammadHossein Bateni, Andre Linhares, Filipe Miguel Goncalves de Almeida, Andrea Montanari, Ashkan Norouzi-Fard

NEURAL INFORMATION PROCESSING SYSTEMS

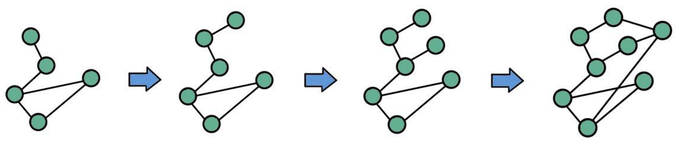
Introduction

We consider community detection on dynamic networks. From a modeling perspective, we propose the **streaming stochastic block model (StSBM)** to model the evolving real-word networks. In terms of algorithms, we restrict to the family of **local streaming algorithms**, which is a variant of local algorithms on dynamic graphs [1]. We prove that in the sparse graph setting, local streaming algorithms can not outperform random guess. On the other side, we propose the **streaming belief propagation algorithm (StreamBP)** which achieves optimal estimation accuracy given a small amount of side information. We evaluate the performance of proposed algorithms on both synthetic and real-world datasets.

Notations

- Denote the number of vertices by n.
- At time $t = 1, \dots, n$, denote the graph at time t by $G(t) = (V(t), E(t))$.
- Let k be the number of communities, and $\tau = [k]^n$ be the vector of true labels. Let $\bar{\tau} = [k]^n$ be the vector of noisy labels (side information).
- Let $B_R^t = (V_R^t, E_R^t)$ denote the ball of radius R in $G(t)$ centered at v.

Streaming stochastic block model



Evolving process of StSBM.

- Reveal one vertex at a time: $V(t+1) = V(t) \cup \{v(t+1)\}$, $|V(t)| = t$.
- StSBM(n, k, p, W, α): draw τ i.i.d. from distribution p. For all vertex v set $\bar{\tau}(v) = \tau(v)$ with probability $1 - \alpha$, and $\bar{\tau}(v) \sim Unif\ ([k] \setminus \{\tau(v)\})$ otherwise. Generate the edges independently with $P((u,v) \in E(n) | \tau, \bar{\tau}) = W_{\tau(u)\tau(v)}$. Finally, generate the graph sequence by choosing a uniformly random permutation over the vertices $\{v(1), v(2), \dots, v(n)\}$.
- **Symmetric StSBM (StSSBM(n,k,a,b,α))**: taking $p = (1/k, \dots, 1/k)$, W having diagonal elements a/n and off-diagonal elements b/n.

Local streaming algorithms

An R-local streaming algorithm keeps a graph $\mathcal{G}_v^t = (\mathcal{V}_v^t, \mathcal{E}_v^t)$ at each vertex v, with initialization $\mathcal{G}_v^0 = (\{v\}, \emptyset)$. At time $t+1$, conduct the following updates:

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

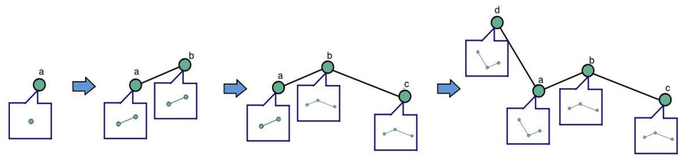
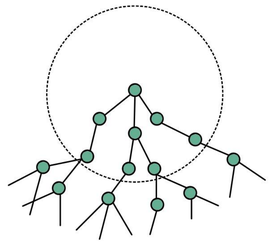


Illustration of updating $\mathcal{G}_v^t$ for 1-local streaming algorithms, where we assume the order of revealing is a -> b -> c -> d.

Theorem. Under StSSBM(n, k, a, b) with no side information, no local streaming algorithn can outperform random guessing.

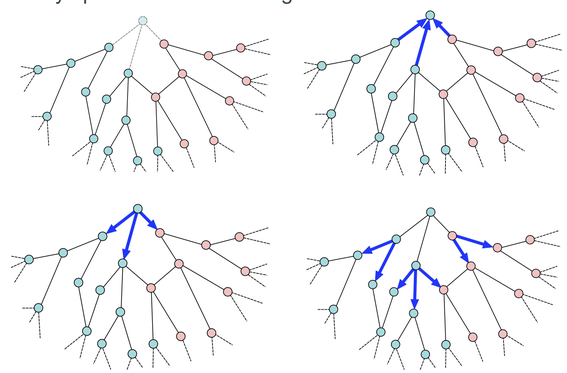
Proof idea:
1. Show that with high probability, $\mathcal{G}_v^n$ is a subgraph of $B_r^n(v)$ for some fixed r. Therefore, streaming local algorithms are essentially local.
2. By [2], local algorithms can not do better than random guessing on sparse graphs. In fact, any bounded radius neighborhood is tree-like.



Any bounded radius neighborhood can be coupled with a Galton-Watson branching tree which is independent of the label.

Streaming belief propagation

StreamBP: a local streaming algorithm that achieves information-theoretically optimal reconstruction given a small amount of side information.



Update schedule of StreamBP. Upon the arrival of a new vertex (shown in the first figure), StreamBP performs the belief propagation updates corresponding to the blue edges in the three other figures.

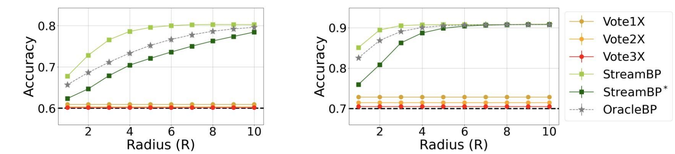
Empirical evaluation

	$\|V\|$	$\|E\|$	k	a	b
Citeseer	3,264	4,536	6	11.47	0.89
Cora	2,708	5,278	7	17.62	0.90
Polblogs	1,490	16,715	2	40.69	4.23
Synthetic	[10,000–50,000]	[20,000–700,000]	[2–5]	[2.5–18]	[0.05–1]

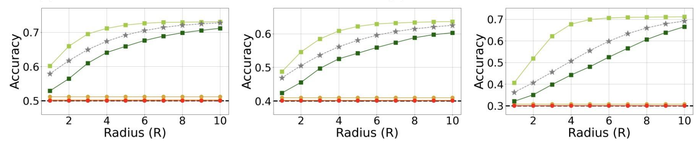
Statistics of the synthetic and real-world datasets.

- **StreamBP*:** a more robust version of StreamBP.
- **Vote1X,Vote2X,Vote3X:** voting algorithms.
- **OracleBP:** offline benchmark.

Synthetic datasets



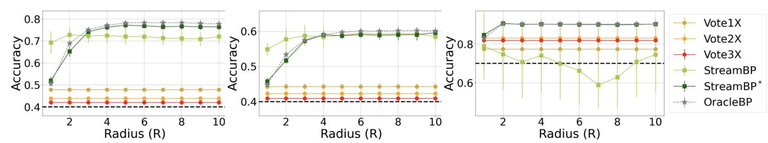
(a) $k = 2, a = 3, b = .1, \alpha = .4$. (b) $k = 2, a = 5, b = .5, \alpha = .3$.



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

Results of the experiments on synthetic datasets with 50,000 vertices. The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely 1 − α.

Real-world datasets



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

Results of experiments on real-world datasets. The black dashed line represents the accuracy of the noisy side information (without using the graph at all), namely 1 − α.

References

[1] Jukka Suomela (2013). “ Survey of local algorithms.” In: ACM Computing Surveys (CSUR), 45(2):1–40.

[2] Varun Kanade (2016) et al. “Global and local information in clustering labeled block models.”. In: IEEE Transactions on Information Theory, 62(10):5906–5917.