Fast and Scalable Loopy Belief Propagation on Real-World Graphs

DATA MINING LAB, DEPT. CSE, SNU

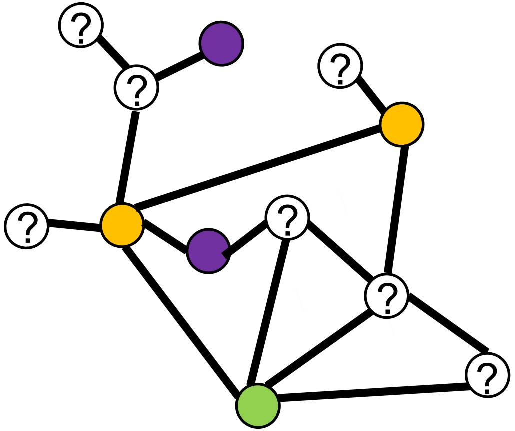
Saehan Jo, Jaemin Yoo, U Kang

11th ACM International Conference on Web Search and Data Mining (WSDM) 2018

Problem Definition - General

Question) Inference Problem on Large Graphs

Given large graphs with partial information of some vertices, how can we make inference on other unobserved vertices?



Important Topic! Applications:

- Recommendation
- Link Prediction
- Anomaly Detection (Malware, Fraud)

Challenges

We can use **Loopy Belief Propagation** on a distributed environment to solve the Inference Problem on Large Graphs.

However! Loopy Belief Propagation on a distributed environment still suffers from

- Power-law degree distribution of real-world graphs
- Burdensome iterative computations
- High communication overhead

Solution?
Distributed Loopy Belief Propagation (Our Method)

Proposed Method – Contributions

Distributed Loopy Belief Propagation (DLBP)

1. Correct Convergence Criterion
2. Minimizing Numerical Computations
3. Minimizing Network Communication

Proposed Method –
1. Correct Convergence Criterion

Main Idea: ***Using "Belief" as the convergence criterion***

Message convergence criterion does not guarantees the
- Lemma 1. Convergence of beliefs
- Lemma 2. Convergence of messages in the next iteration

Belief convergence criterion guarantees the
- Lemma 3. Convergence of beliefs
- Lemma 4. Convergence of messages between two converged vertices

Proof: We use Linearized Belief Propagation (VLDB'15) to prove the lemmas

Linearization of update equations of LBP

Proposed Method – 2. Minimizing Numerical Computations

Main Idea: ***Skipping of Converged Vertices***

Belief convergence criterion guarantees the convergence of messages between two converged vertices (Lemma 3).

- Thus, DLBP skips the computation of outgoing messages from vertices that have converged at the previous iteration.

Advantages:

- Omits redundant computations
- Reduces the number of iterations until convergence

Proposed Method – 3. Minimizing Network Communication

Main Idea: *Hub-Oriented Scheduling*

- Focus on achieving the convergence of high-degree vertices

Preprocessing Stage

- Divide the vertices into hubs (high-degree) and spokes (low-degree)

Main Stage

- Iterate super-steps until convergence

Advantages: (Refer to paper for detailed analysis)

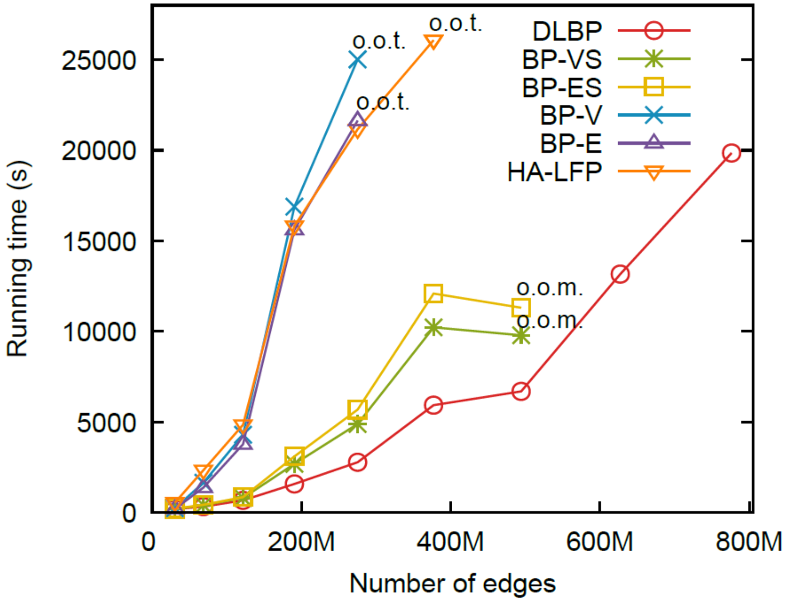
- Reduce the amount of shuffled data
- Lower time complexity
- Lower memory usage

Result - Timing

Q1. Timing: How fast is DLBP compared to standard LBP on a distributed environment?

Result

- **DLBP is up to <u>10.7x faster</u> than standard BP**
- **DLBP is up to <u>10.0x faster</u> than HA-LFP**
- **DLBP shows <u>additional 2.0x improvement</u> and <u>better scalability</u> by applying *Hub-Oriented Scheduling***



Result - Accuracy

Q2. Accuracy: How does the label classification of DLBP perform compare to that of standard LBP?

Result

- **The difference between the accuracies of any two methods is always <u>less than 0.14%</u>**

Dataset	BP-E	BP-V	BP-ES	BP-VS	DLBP
Campaigns	89.36%	89.36%	89.31%	89.31%	89.31%
PolBlogs	95.62%	95.62%	95.62%	95.62%	95.62%
PubMed	82.65%	82.65%	82.79%	82.79%	82.65%

Conclusion

DLBP enhances the standard LBP by overcoming the challenges associated with large real-world graphs by

1. Using a convergence criterion better suited for real-world graphs
2. Skipping redundant message computations
3. Carefully scheduling the sub-iterations to minimize the network communication