Supervised Belief Propagation: Scalable Supervised Inference on Attributed Networks

Jaemin Yoo, Saehan Jo, and U Kang

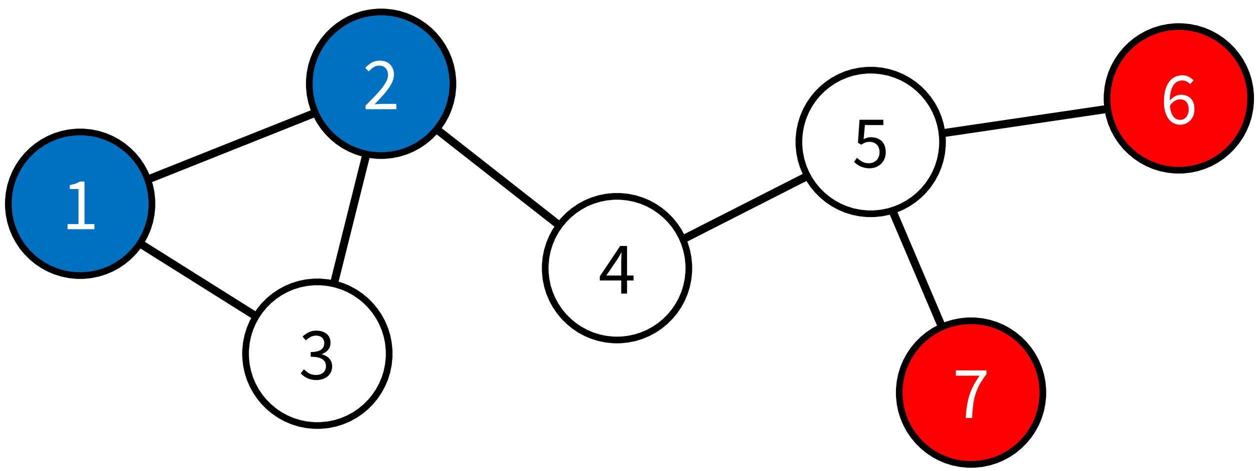
Seoul National University

Outline

1. **Introduction**
2. Proposed Method
3. Experiments
4. Conclusion

Network of Political Blogs

- We are given a network of political blogs
- Each blog is either **Liberal** or **Conservative**
- Only some blogs have been observed



Node Classification

- How can we classify unobserved nodes?
- This problem is called **node classification**

- **Real-world applications:**
- Fraud detection in auction networks
- User labeling in social networks
- Malware detection in file-machine networks

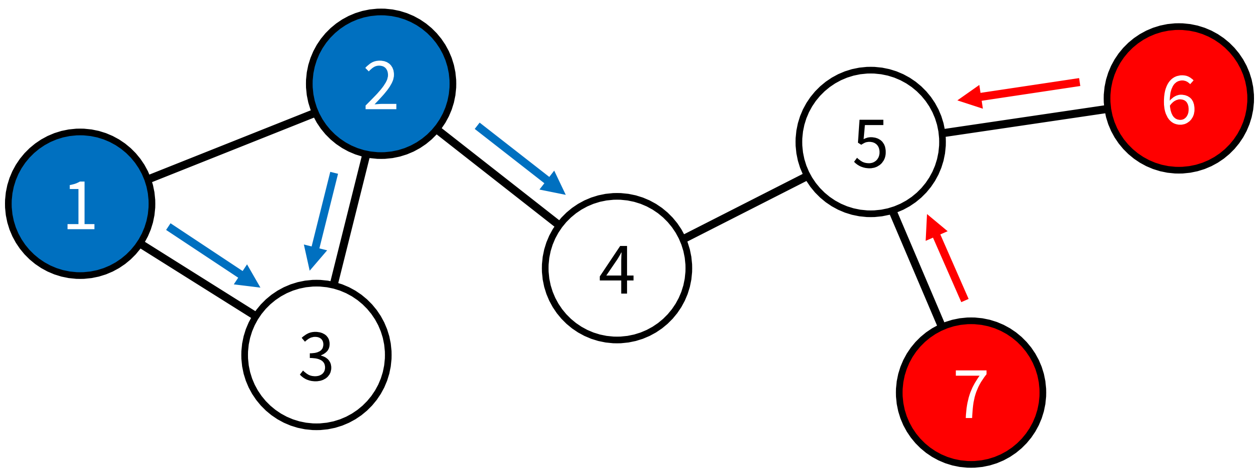
Loopy Belief Propagation

- Widely used method for node classification
- Uses an **affinity table** to model the network
- For any pair of connected nodes
  - the probability of having a same label is ϵ
  - the probability of having different labels is $1 - \epsilon$

	Label 1	**Label 2**
Label 1	ϵ	$1 - \epsilon$
Label 2	$1 - \epsilon$	ϵ

LBP in the Network of Blogs

- LBP propagates information when $\epsilon > 0.5$
- This is called the property of **homophily**
- Nodes 3 and 4 as liberal, 5 as conservative



Two Assumptions of LBP

- LBP assumes **homophily** and **uniformity**
- Homophily ($\epsilon > 0.5$):
  - Adjacent nodes are likely to have a same label
  - This is called *guilt-by-association* in literature
- Uniformity (ϵ is globally applied):
  - All edges have a same **propagation strength**

Attributed Network

- Let's say the network is **attributed:**
  - Each edge (i, j) is attached with a feature vector θ_{ij}
- Then, these assumptions may break

Node 1	Node 2	Features
1	3	$(15,3,0)$
1	3	$(5,-1,0)$
	$\dots$	
5	7	$(0,0,10)$

LBP in Attributed Networks

- Edges have distinct and different features
- Are the edges uniform?
  - **No.** Their strengths depend on their features
- Do the nodes represent homophily?
  - **Not sure.** Some can even represent heterophily
- Moreover, it is difficult to choose a proper ϵ

- **Not appropriate for attributed networks!**

Research Question

- **Question.** How can we classify nodes in an attributed network with high accuracy?
- **Challenges:**
  - Domain knowledge is not given
  - Different values of ϵ should be modeled
  - Edges may follow the heterophily

- We propose **Supervised belief propagation!**

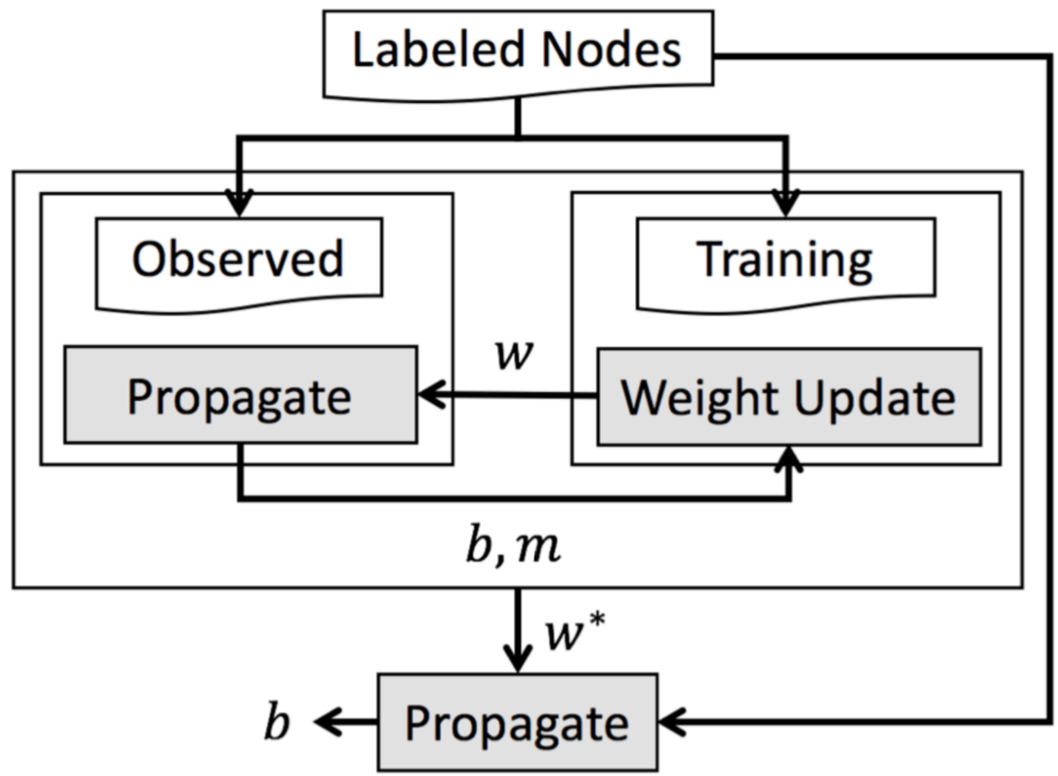
Outline

1. Introduction
2. **Proposed Method**
3. Experiments
4. Conclusion

Supervised Belief Propagation

- Node classification method for attr. networks
- Learns the **weight vector** w as a parameter
- Alternates the following steps until w converges:
  - Propagation step
  - Weight update step
- Then, uses the optimal w in final classification

Flowchart of SBP



Alternating Updates

- The process is similar to an EM algorithm
- **In the propagation step:**
  - SBP uses the current w in classification
  - This is similar to the **expectation** in EM
- **In the weight update step**:
  - SBP updates w toward a local optimum
  - This is similar to the **maximization** in EM

Return Values of SBP

- We assume a network of binary labels
  - Name one label as **pos.**, the other as **neg.**
- SBP returns the belief b_i of every node i:
  - The probability of node i for being positive
- Then we classify node i based on b_i:

$$\text{Label of node } i = \begin{cases} \text{pos.} & \text{if } b_i > 0.5 \\ \text{neg.} & \text{if } b_i < 0.5 \end{cases}$$

Key Ideas of SBP

- **Idea 1.** Model learnable prop. strengths
- **Idea 2.** Introduce a differentiable loss
- **Idea 3.** Iteratively update the parameter

Idea 1. Prop. Strength

- To model the strength ϵ_{ij} as a function:

$$\epsilon_{ij} = \left(1 + \exp\left(-\theta_{ij}^{\mathrm{T}} w\right)\right)^{-1}$$

where

- θ_{ij} is the feature vector of edge (i, j)
- w is the **weight vector** to be optimized

Idea 2. Differentiable Loss

- To use a differentiable loss $E(w)$:

$$E(w) = \lambda \|w\|_2^2 + \sum_p \sum_n h\left(b_n - b_p\right)$$

where

- $h = (1 + \exp(-x/d))^{-1}$ is an increasing func.
- p and n are positive and negative nodes, resp.
- b_i is the belief of node i for being positive

Idea 3. Gradient Update

- To update w with the gradient descent:

$$w \leftarrow w - \mathrm{clip}\left(\alpha \frac{\partial E(w)}{\partial w}; \beta\right)$$

where

- α is a step size parameter
- $\mathrm{clip}(\cdot\,; \beta)$ restricts the size of each update to β

Outline

1. Introduction
2. Proposed Method
3. **<u>Experiments</u>**
4. Conclusion

Experimental Settings

- MovieLens is a user-movie bipartite network
- Each edge has an integer rating of 1-5
- We classify the movies into “recomm.” and not

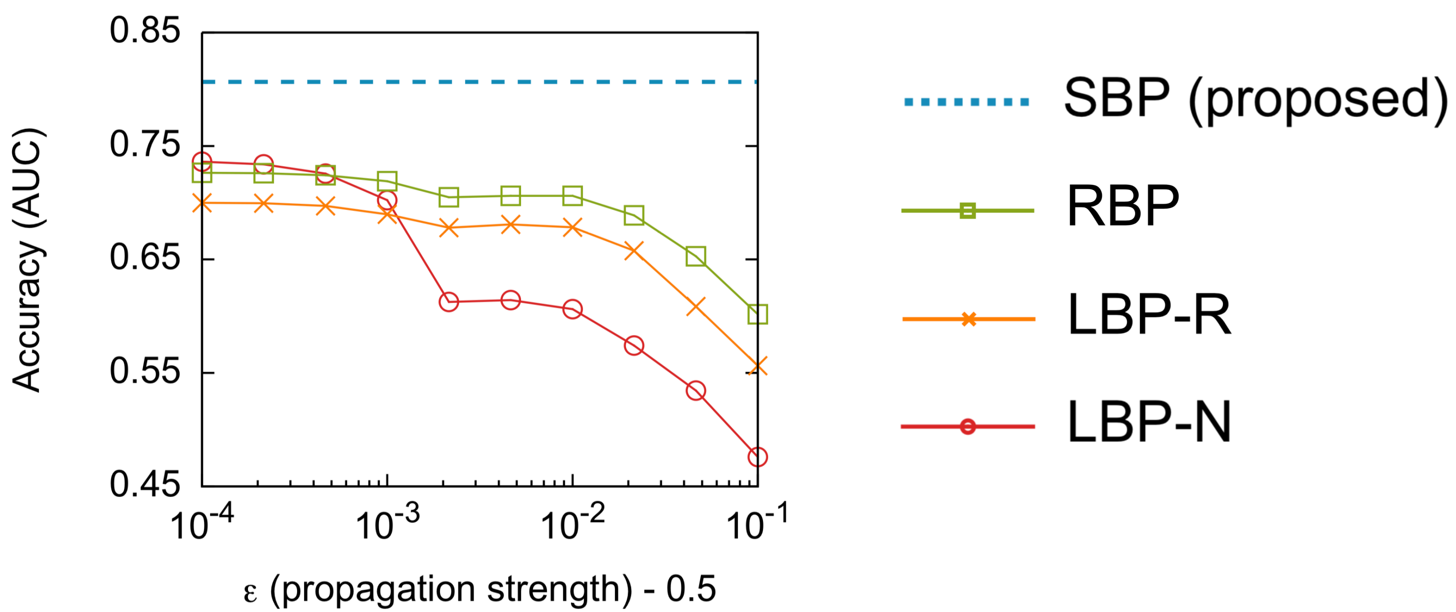
Dataset	Nodes	Edges	Attributes
Epinions-R[2]	189,028	1,152,005	ratings and trusts
Epinions-S[3]	131,828	841,372	signs (trusts or distrusts)
MovieLens[4]	9,940	1,000,209	ratings (1 to 5)

Experimental Questions

- **Q1.** How sensitive are previous methods to ϵ?
- **Q2.** How accurately does SBP classify nodes?
- **Q3.** How does $E(w)$ change during iterations?
- **Q4.** How does the running time scale?

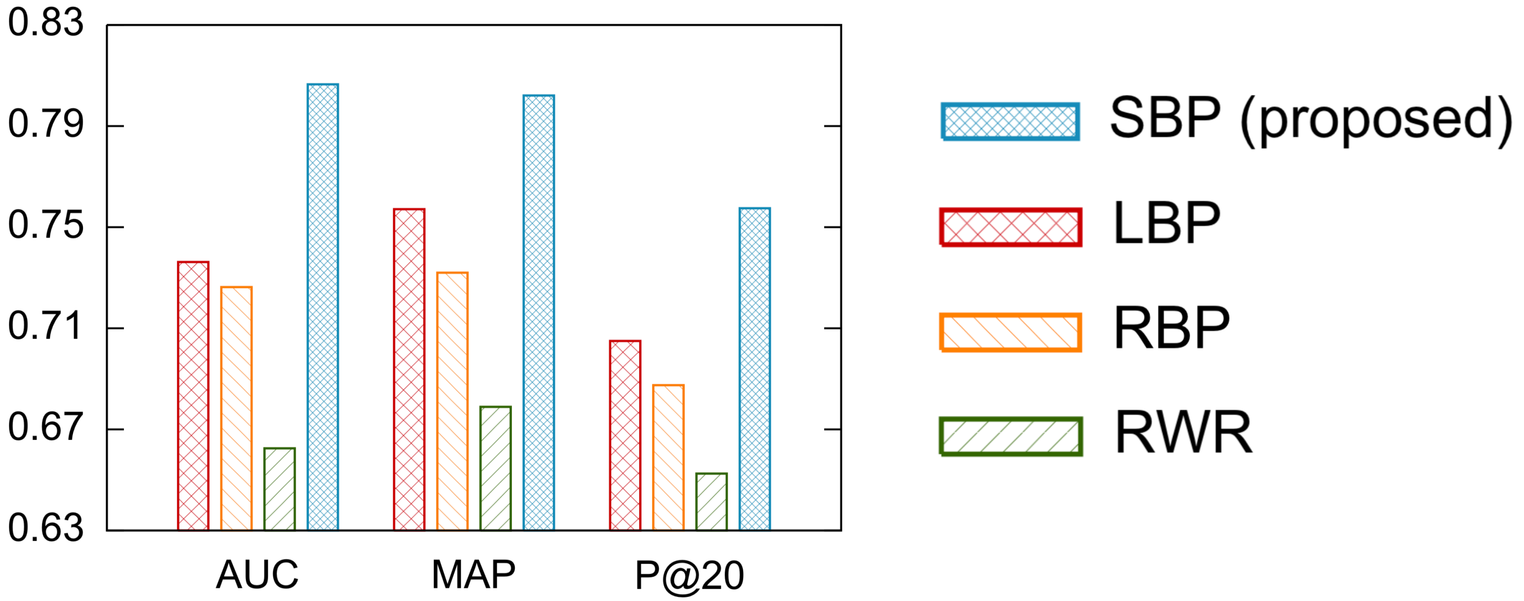
Sensitivity to Prop. Strength

- **Q1.** How sensitive are previous methods to ϵ?
- **Ans.** Previous methods highly depend on its val.



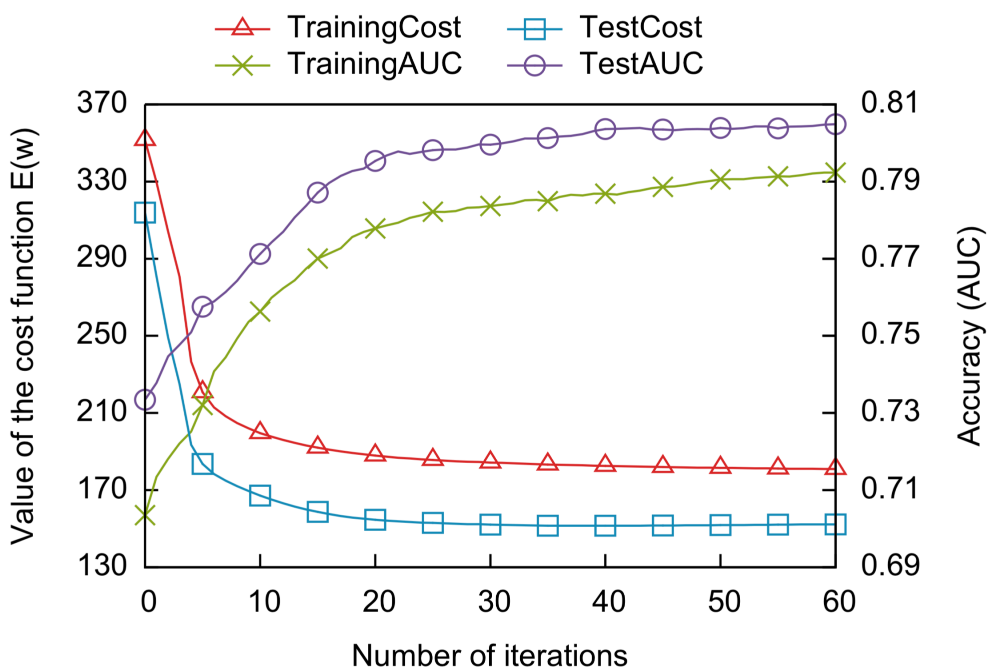
Classification Accuracy

- **Q2.** How accurately does SBP classify nodes?
- **Ans.** SBP shows the best AUC, MAP, and P@20



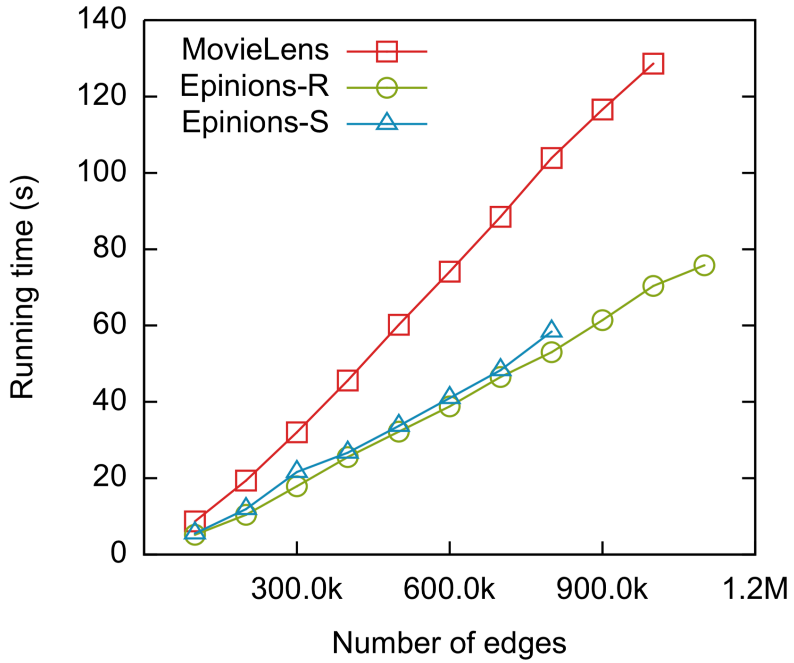
Cost Minimization

- **Q3.** How does $E(w)$ change during iterations?
- **Ans.** It is minimized for both train. and test sets



Linear Scalability

- **Q4.** How does the running time scale?
- **Ans.** It scales linearly with the number of edges



Outline

1. Introduction
2. Proposed Method
3. Experiments
4. **<u>Conclusion</u>**

Conclusion

- **Method: Supervised belief propagation**
- **Key ideas:**
  - Consider rich feature vectors in propagation
  - Learn the propagation strength of each edge
- **Contributions:**
  - Generalize previous LBP-based methods
  - Provide up to 15.6% higher AUC
  - Linearly scalable with the number of edges

Thank you!

Jaemin Yoo, Saehan Jo, and U Kang

https://datalab.snu.ac.kr/sbp/