Uncertainty-Based Opinion Inference on Network Data Using Graph Convolutional Neural Networks

IEEE MILCOM 2018

Xujiang Zhao[1], Feng Chen[1], Jin-Hee Cho[2]

October 30, 2018

[1]University at Albany - SUNY, [2]Virginia Tech

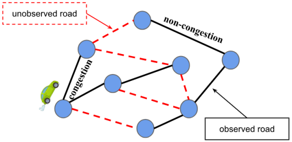
Outline

- Motivation
- Research Problem & Challenge
- Related Work
- Graph Convolutional Networks
- Proposed approach
- Experimental results
- Conclusion & Future Work

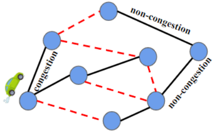
Motivation

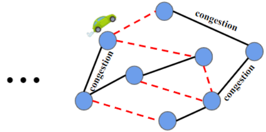
- Decision Making with Uncertain Opinions
- **When Useful?**
- Trust in social networks
- Opinion diffusion
- Graph summarization.

In a traffic network, how can we predict the traffic condition of unobserved roads (e.g., congested vs. non-congested)?



What if we have so many observations?



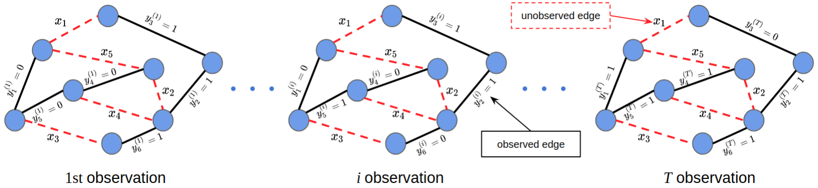


Research Problem & Challenges

Given

- $\mathcal{G} = (\mathbb{V}, \mathbb{E} = \mathbb{Y} \cup \mathbb{X}, f)$, an input network;
- $\{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(T)}\}$, the **observations** of a vector of input Boolean variables and $\boldsymbol{\omega}_{\mathbf{y}} = (\omega_{y_1}, \cdots, \omega_{y_M})$, the subjective opinions on $\mathbf{y}$.

Predict $\boldsymbol{\omega}_{\mathbf{x}}$, the unknown opinion on the vector of target Boolean variables $\mathbf{x}$.



How can we accurately and efficiently predict unknown opinions with a large, heterogeneous, uncertain network data?

Contributions

Research goal: Accurately and efficiently predict unknown opinions with a large, heterogeneous, uncertain network data.
Key Contribution:

1. The proposed GCN-based framework is the **first** deep learning framework that is capable of predicting the opinions of multiple nodes in a network collectively.
2. The proposed GCN-based method achieves both **efficiency** and **effectiveness** by leveraging the GCN to model heterogeneous dependencies and *knowledge distillation* to transfer the heterogeneous dependencies into the prediction of opinions.
3. We validate the performance of our proposed approach through two road traffic datasets.

Uncertain, Subjective Opinion in Subjective Logic (SL)

- A binomial opinion is defined in terms of belief, disbelief, and uncertainty towards a given proposition. An opinion ω is represented by

$$\omega = (b, d, u, a) \tag{1}$$

  where

  - b: belief (e.g., agree)
  - d: disbelief (e.g., disagree)
  - u: uncertainty (i.e., ignorance, vacuity, or lack of evidence)
  - a: a base rate, a prior, general knowledge upon no commitment

Uncertain, Subjective Opinion in Subjective Logic (SL)

- A binomial opinion is defined in terms of belief, disbelief, and uncertainty towards a given proposition. An opinion ω is represented by

$$\omega = (b, d, u, a) \tag{1}$$

where

- b: belief (e.g., agree)
- d: disbelief (e.g., disagree)
- u: uncertainty (i.e., ignorance, vacuity, or lack of evidence)
- a: a base rate, a prior, general knowledge upon no commitment

and

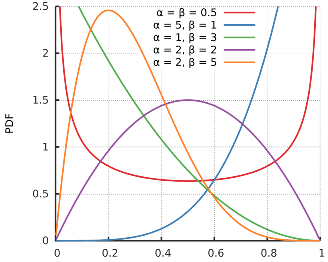
$$b + d + u = 1 \tag{2}$$

SL's Binomial Opinion with Beta Distribution

- A binomial opinion follows a Beta PDF, denoted by,

$$\text{Beta}(p|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{3}$$

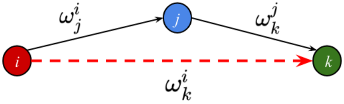
where α is the number of positive evidence and β is the number of negative evidence.



- $\omega = (\alpha,\beta)$, which can be translated to $\omega = (b,d,u,a)$.

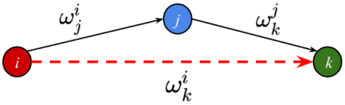
Operators with Uncertain Opinions in SL

- **Discount operator,** $\otimes$: Discount trust of an entity one wants to interact when it does not have any direct interaction with the entity, e.g., $w_k^i = w_j^i \otimes w_k^j$

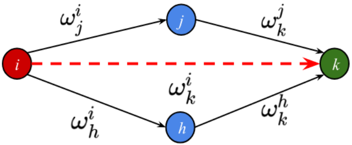


Operators with Uncertain Opinions in SL

- **Discount operator,** $\otimes$: Discount trust of an entity one wants to interact when it does not have any direct interaction with the entity, e.g., $w^i_k = w^i_j \otimes w^j_k$



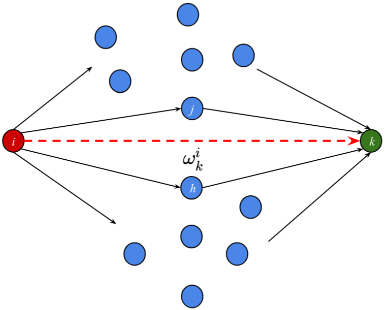
- **Consensus operator,** $\oplus$: Find a consensus between two opinions where two entities observe a same entity, e.g., $w^i_k = (w^i_j \otimes w^j_k) \oplus (w^i_h \otimes w^h_k)$



[Jøsang, Springer 2016]

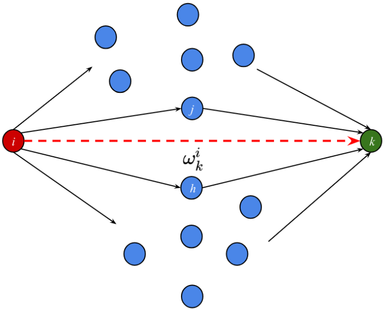
Scalability Issue in Subjective Logic

When a network is large, there are too many paths to consider for fusing them.



Scalability Issue in Subjective Logic

When a network is large, there are too many paths to consider for fusing them.



Limitation

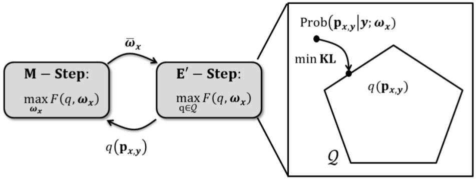
SL's operators are good for fusing two opinions in dyadic relationships; **not scalable for multiple opinions with large network data**.

Collective Subjective Logic (CSL)

A variant of SL, combining Probabilistic Soft Logic (PSL) and Markov Random Fields (MRFs) with SL

$$\max_{\omega_x, \xi \geq 0} \mathcal{L}(\omega_x) = \max_{\omega_x, \xi \geq 0} \log \text{Prob}(y; \omega_x, \omega_y)$$

$$s.t. \rho_i \mathbb{E}_{\text{Prob}(p_{x,y}|y;\omega_x,\omega_y)}[1 - r_i(p_{x,y})] \leq \xi_i, \|\xi\|_\beta \leq \epsilon, i = 1, \cdots, k$$

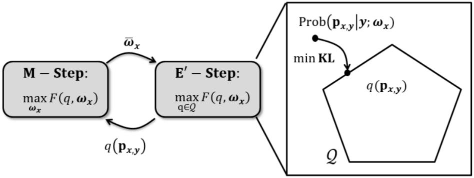


Collective Subjective Logic (CSL)

A variant of SL, combining Probabilistic Soft Logic (PSL) and Markov Random Fields (MRFs) with SL

$$\max_{\omega_\mathrm{x}, \xi \geq 0} \mathcal{L}(\omega_\mathrm{x}) = \max_{\omega_\mathrm{x}, \xi \geq 0} \log \mathbf{Prob}(\mathrm{y}; \omega_\mathrm{x}, \omega_\mathrm{y})$$

$$s.t. \rho_i \mathbb{E}_{\mathrm{Prob}(\mathrm{p}_{\mathrm{x},\mathrm{y}} | \mathrm{y}; \omega_\mathrm{x}, \omega_\mathrm{y})}[1 - r_i(\mathrm{p}_{\mathrm{x},\mathrm{y}})] \leq \xi_i, \|\xi\|_\beta \leq \epsilon, i = 1, \cdots, k$$



Limitation

The assumption of distribution based on MRFs limits its capability to deal with, large-scale, **heterogeneous** network data that may be lossy, noisy, incomplete, and/or missing.

[Chen, Wang & Cho, Bigdata 2017]

Why Deep Learning Needed?

Both SL and CSL are:

- not scalable.
- not effectively dealing with heterogeneous data.

How to Solve These Challenge?

Why Deep Learning Needed?

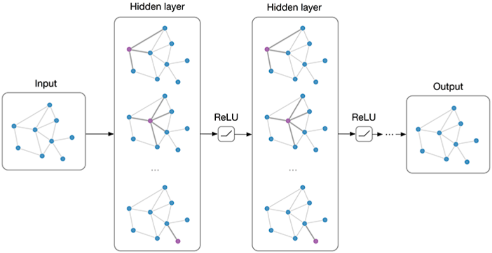
Both SL and CSL are:

- not scalable.
- not effectively dealing with heterogeneous data.

How to Solve These Challenge?

Graph Convolutional Network can provide solutions for

- dealing with graph network data
- modeling heterogeneous dependency
- processing large-scale data (i.e., scalability)



[Kipf & Welling, ICLR 2017]

Graph Convolutional Networks (GCN)

What capability can GCN offer?

- node classification
- graph classification

Graph Convolutional Networks (GCN)

What capability can GCN offer?

- node classification
- graph classification

How to use the convolution operator on graph data *effectively* and *efficiently*?

Graph Convolutional Networks (GCN)

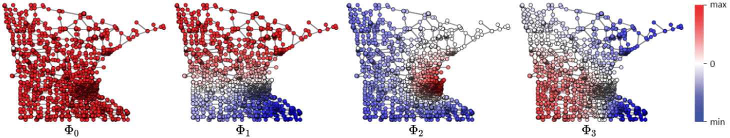
What capability can GCN offer?

- **node classification**
- graph classification

How to use the **convolution operator** on graph data *effectively* and *efficiently*?

Graph Fourier Transform:

- on Euclidean spaces: $\mathbf{r} = \sum_{k \geq 0} \hat{\mathbf{r}}_k e^{ik}$
- on non-Euclidean spaces: $\mathbf{r} = \sum_{k \geq 0} \hat{\mathbf{r}}_k \phi_k = \phi^T \phi \mathbf{r}$
  where $L = \Phi \Lambda \Phi^T$, L is the Graph Laplacian matrix,
  $\Phi = (\phi_1, \cdots, \phi_n)$ is the orthonormal **eigenvectors** and
  $\Lambda = diag(\lambda_1, \cdots, \lambda_n)$ is the diagonal matrix of **eigen values**.



Graph Convolution

- Given two signals $\mathbf{r}$ and $\mathbf{b}$ on graph, graph convolution

$$\mathbf{r} \star \mathbf{b} = \Phi^T(\Phi^T\mathbf{r}) \circ (\Phi^T\mathbf{b}) = \Phi diag(\hat{r}_1, \cdots, \hat{r}_n)\hat{\mathbf{b}}, \tag{4}$$

  convolution on Fourier domain is element-wise product of their Fourier transformations
- Graph convolutional layer

$$g_\theta \star \mathbf{r} = \Phi g_\theta \Phi^T \mathbf{r}. \tag{5}$$

Graph Convolution

- Given two signals $\mathbf{r}$ and $\mathbf{b}$ on graph, **graph convolution**

$$\mathbf{r} \star \mathbf{b} = \Phi^T(\Phi^T\mathbf{r}) \circ (\Phi^T\mathbf{b}) = \Phi diag(\hat{r}_1, \cdots, \hat{r}_n)\hat{\mathbf{b}}, \tag{4}$$

  convolution on Fourier domain is **element-wise product** of their Fourier transformations
- **Graph convolutional layer**

$$g_\theta \star \mathbf{r} = \Phi g_\theta \Phi^T \mathbf{r}. \tag{5}$$

  While **computationally expensive** of Φ is $O(n^2)$. $g_\theta(\Lambda)$ can be well approximated by Chebyshev polynomials

$$g_\theta(\Lambda) \approx \sum_{k=1}^{K} \theta_k T_k(\tilde{\Lambda}), T_k(r) = 2xT_{k-1}(r) - T_{k-2}(r) \tag{6}$$

- **Graph Convolution** of a signal $\mathbf{r}$ with a filter g_θ approximated by

$$g_\theta \star \mathbf{r} \approx \sum_{k=1}^{K} \theta_k T_k(\tilde{L})\mathbf{r}. \tag{7}$$

Proposed Approach: GCN-based Uncertain Opinion Prediction

Graph Convolutional Networks can:

- dealing with graph data (road traffic networks, social networks)
- consider a graph convolution layer that models heterogeneous dependency
- provide high efficiency with low complexity (i.e., linear time complexity) based on the Chebyshev approximated

Proposed Approach: GCN-based Uncertain Opinion Prediction

Graph Convolutional Networks can:

- dealing with graph data (road traffic networks, social networks)
- consider a graph convolution layer that models heterogeneous dependency
- provide high efficiency with low complexity (i.e., linear time complexity) based on the Chebyshev approximated

However, GCN **cannot be directly** applied to predict ω_X because it cannot model the opinions directly.

A Probabilistic Model of Uncertain Opinions in SL

- Following Bayesian distributions:

$$y_i \sim \text{Bern}(y_i; p_{y_i}); p_{y_i} \sim \text{Beta}(p_{y_i}; \omega_{y_i}) \quad (8)$$

A Probabilistic Model of Uncertain Opinions in SL

- Following Bayesian distributions:

$$y_i \sim \text{Bern}(y_i; p_{y_i}); p_{y_i} \sim \text{Beta}(p_{y_i}; \omega_{y_i}) \tag{8}$$

- The PDF of y_i based on its opinion ω_{y_i} can be calculated as

$$q(y_i; \omega_{y_i}) = \int \text{Beta}(p_{y_i}; \omega_{y_i})\text{Bern}(y_i; p_{y_i})dp_{y_i} = \text{Bern}(y_i; \frac{\alpha_{y_i}}{\alpha_{y_i} + \beta_{y_i}}) \tag{9}$$

where $\omega_{y_i} = (\alpha_{y_i}, \beta_{y_i})$.

A Probabilistic Model of Uncertain Opinions in SL

- Following Bayesian distributions:

$$y_i \sim \text{Bern}(y_i; p_{y_i}); p_{y_i} \sim \text{Beta}(p_{y_i}; \omega_{y_i}) \tag{8}$$

- The PDF of y_i based on its opinion ω_{y_i} can be calculated as

$$q(y_i; \omega_{y_i}) = \int \text{Beta}(p_{y_i}; \omega_{y_i})\text{Bern}(y_i; p_{y_i})dp_{y_i} = \text{Bern}(y_i; \frac{\alpha_{y_i}}{\alpha_{y_i} + \beta_{y_i}}) \tag{9}$$

where $\omega_{y_i} = (\alpha_{y_i}, \beta_{y_i})$.
- The **joint PDF** function of $\mathbf{x}$ and $\mathbf{y}$:

$$q(\mathbf{x}, \mathbf{y}; \omega_{\mathbf{x}}, \omega_{\mathbf{y}}) = \prod_{i=1}^{N} q(x_i; \omega_{x_i}) \prod_{j=1}^{M} q(y_j; \omega_{y_j}) \tag{10}$$

A GCN Model

- A GCN model defines a conditional PDF $p(\mathrm{x}, \mathrm{y}|\mathrm{r}; \theta)$ by using a softmax output layer that produces a $((N + M) \times 2)$-dimensional soft prediction matrix $p_{\mathrm{x},\mathrm{y}} \in [0, 1]^{(N+M)\times 2}$ as defined below,

$$p_{\mathrm{x},\mathrm{y}} = g(\mathrm{r}; A, \theta) : \mathbb{R}^{M+N} \to [0, 1]^{(M+N)\times 2}, \tag{11}$$

A GCN Model

- A GCN model defines a conditional PDF $p(\mathrm{x}, \mathrm{y}|\mathrm{r}; \theta)$ by using a softmax output layer that produces a $((N+M) \times 2)$-dimensional soft prediction matrix $p_{\mathrm{x,y}} \in [0,1]^{(N+M)\times 2}$ as defined below,

$$p_{\mathrm{x,y}} = g(\mathrm{r}; A, \theta) : \mathbb{R}^{M+N} \rightarrow [0,1]^{(M+N)\times 2}, \tag{11}$$

- The conditional PDF function $p(\mathrm{x}, \mathrm{y}|\mathrm{r}; \theta)$ has the form as

$$p(\mathrm{x}, \mathrm{y}|\mathrm{r}; \theta) = \prod_{i=1}^{N} p(x_i|r_i; \theta) \prod_{j=1}^{M} p(y_i|r_{N+j}; \theta) \tag{12}$$

where $p(x_i|r_i; \theta) = \prod_{k=1}^{2} [g_{i,k}(r, A; \theta)]^{x_{i,k}}$ and $p(y_i|r_{N+j}; \theta) = \prod_{k=1}^{2} [g_{i+N,k}(r, A; \theta)]^{y_{i,k}}$.

Bridging the Two Models

- Probabilistic Model $q(\mathsf{x}, \mathsf{y}; \omega_{\mathsf{x}}, \omega_{\mathsf{y}})$ **cannot model the heterogeneous dependencies**.

Bridging the Two Models

- Probabilistic Model $q(\mathsf{x}, \mathsf{y}; \omega_{\mathsf{x}}, \omega_{\mathsf{y}})$ cannot model the heterogeneous dependencies.
- GCN model $p(\mathsf{x}, \mathsf{y}|\mathsf{r}; \theta)$ is cannot model the opinions directly.

Bridging the Two Models

- Probabilistic Model $q(\mathbf{x}, \mathbf{y}; \omega_{\mathbf{x}}, \omega_{\mathbf{y}})$ **cannot model the heterogeneous dependencies**.
- GCN model $p(\mathbf{x}, \mathbf{y}|\mathbf{r}; \theta)$ is **cannot model the opinions directly**.

Transferring of the dependency information from the GCN model to the probabilistic model of opinions for predicting opinions $\boldsymbol{\omega}_{\mathbf{x}}$.

$$\min_{\boldsymbol{\omega}_{\mathbf{x}}} \mathrm{KL}\Big(\prod_{t=1}^{T} q(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}; \boldsymbol{\omega}_{\mathbf{x}}, \boldsymbol{\omega}_{\mathbf{y}}) || \prod_{t=1}^{T} p(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}|\mathbf{r}^{(t)}; \theta^{(\ell)}) \Big), \tag{13}$$

Bridging the Two Models

- Probabilistic Model $q(\mathrm{x}, \mathrm{y}; \omega_{\mathrm{x}}, \omega_{\mathrm{y}})$ cannot model the heterogeneous dependencies.
- GCN model $p(\mathrm{x}, \mathrm{y}|\mathrm{r}; \theta)$ is cannot model the opinions directly.

Transferring of the dependency information from the GCN model to the probabilistic model of opinions for predicting opinions $\boldsymbol{\omega}_{\mathrm{x}}$.

$$\min_{\boldsymbol{\omega}_{\mathrm{x}}} \mathrm{KL}\Big(\prod_{t=1}^{T} q(\mathrm{x}^{(t)}, \mathrm{y}^{(t)}; \boldsymbol{\omega}_{\mathrm{x}}, \boldsymbol{\omega}_{\mathrm{y}}) || \prod_{t=1}^{T} p(\mathrm{x}^{(t)}, \mathrm{y}^{(t)}|\mathrm{r}^{(t)}; \theta^{(\ell)})\Big), \tag{13}$$

Estimation of θ for the GCN model based on **feedback from the predicted opinions $\boldsymbol{\omega}_{\mathrm{x}}^{(\ell)}$** at iteration ℓ.

$$\theta^{(\ell+1)} = \arg\min_{\theta} \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{j=1}^{2} \pi y_{i,j}^{(t)} \cdot \log g_{i+N,j}(r^{(t)}, A; \theta)$$

Bridging the Two Models

- Probabilistic Model $q(\mathbf{x}, \mathbf{y}; \omega_\mathbf{x}, \omega_\mathbf{y})$ **cannot model the heterogeneous dependencies**.
- GCN model $p(\mathbf{x}, \mathbf{y}|\mathbf{r}; \theta)$ is **cannot model the opinions directly**.

Transferring of the dependency information from the GCN model to the probabilistic model of opinions for predicting opinions $\boldsymbol{\omega}_\mathbf{x}$.

$$\min_{\boldsymbol{\omega}_\mathbf{x}} \mathrm{KL}\Big(\prod_{t=1}^{T} q(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}; \boldsymbol{\omega}_\mathbf{x}, \boldsymbol{\omega}_\mathbf{y}) || \prod_{t=1}^{T} p(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}|\mathbf{r}^{(t)}; \theta^{(\ell)})\Big), \tag{13}$$

Estimation of θ for the GCN model based on **feedback from the predicted opinions** $\boldsymbol{\omega}_\mathbf{x}^{(\ell)}$ at iteration ℓ.

$$\theta^{(\ell+1)} = \arg\min_{\theta} \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{2} \pi y_{i,j}^{(t)} \cdot \log g_{i+N,j}(r^{(t)}, A; \theta) + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{2} (1-\pi)\bar{x}_{i,j} \log g_{i,j}(r^{(t)}, A; \theta) \tag{14}$$

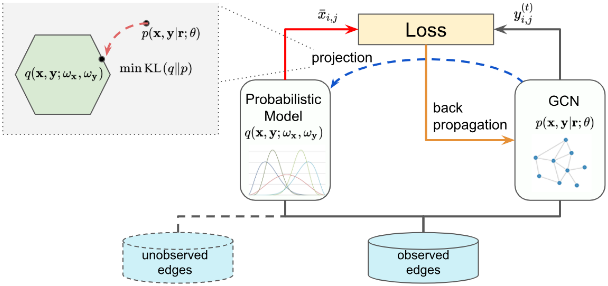
predicted opinions $\boldsymbol{\omega}_\mathbf{x}^{(\ell)}$ are given

Bridging the Two Models

$$\min_{\boldsymbol{\omega}_{\mathbf{x}}} \mathrm{KL}\left(\prod_{t=1}^{T} q(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}; \boldsymbol{\omega}_{\mathbf{x}}, \boldsymbol{\omega}_{\mathbf{y}}) \| \prod_{t=1}^{T} p(\mathbf{x}^{(t)}, \mathbf{y}^{(t)} | \mathbf{r}^{(t)}; \theta^{(\ell)})\right)$$

θ ω_x

$$\theta^{(\ell+1)} = \arg\min_{\theta} \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{2} \pi y_{i,j}^{(t)} \cdot \log g_{i+N,j}(r^{(t)}, A; \theta) + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{2} (1-\pi)\bar{x}_{i,j} \log g_{i,j}(r^{(t)}, A; \theta)$$



Datasets & Experimental Setting

- Road traffic datasets:

Dataset name	# nodes	# edges	# weeks	# snapshots in total
D.C.	1,383	1,878	43	3440
Philadelphia	603	708	43	3440

Datasets & Experimental Setting

- Road traffic datasets:

Dataset name	# nodes	# edges	# weeks	# snapshots in total
D.C.	1,383	1,878	43	3440
Philadelphia	603	708	43	3440

- Parameter settings
  - Time window size: $T \in \{2, 3, 6, 8, 11\}$,
  - Uncertainty mass values: $u \in \{50\%, 40\%, 25\%, 20\%, 15\%\}$
  - Test Ratio: $TR \in \{10\%, 30\%\}$

Datasets & Experimental Setting

- Road traffic datasets:

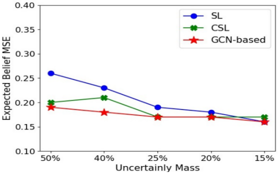
Dataset name	# nodes	# edges	# weeks	# snapshots in total
D.C.	1,383	1,878	43	3440
Philadelphia	603	708	43	3440

- Parameter settings
  - Time window size: $T \in \{2, 3, 6, 8, 11\}$,
  - Uncertainty mass values: $u \in \{50\%, 40\%, 25\%, 20\%, 15\%\}$
  - Test Ratio: $TR \in \{10\%, 30\%\}$
- Performance metrics:

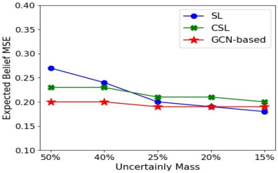
$$\text{EB-MSE}(\boldsymbol{\omega}_{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N} \Big| \underbrace{\frac{a_{x_i}}{a_{x_i} + b_{x_i}}}_{\text{prediction}} - \underbrace{\frac{a^\star_{x_i}}{a^\star_{x_i} + b^\star_{x_i}}}_{\text{ground truth}} \Big| \quad (15)$$

- Computation time metric: seconds

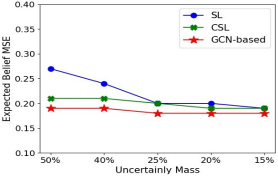
Result and Analysis



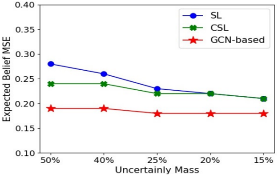
(a) 10% Test Ratio, PA Dataset



(b) 30% Test Ratio, PA Dataset

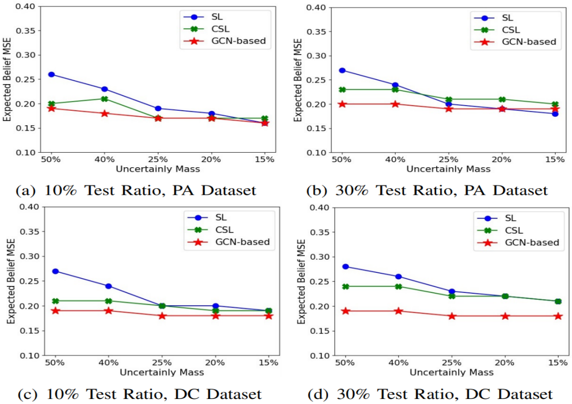


(c) 10% Test Ratio, DC Dataset



(d) 30% Test Ratio, DC Dataset

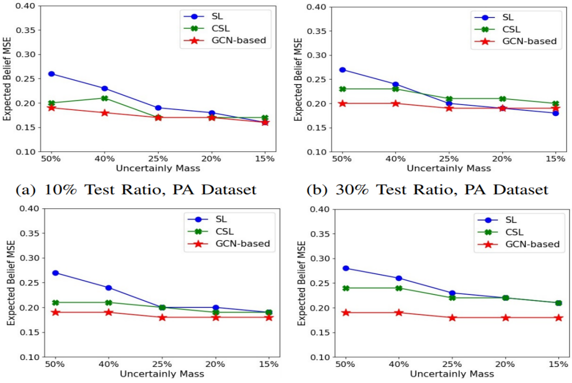
Result and Analysis



(a) 10% Test Ratio, PA Dataset
(b) 30% Test Ratio, PA Dataset
(c) 10% Test Ratio, DC Dataset
(d) 30% Test Ratio, DC Dataset

- GCN-based outperforms among all methods
- As uncertainty mass, u, increases, **high resilience (low sensitivity) with GCN-based** is observed
- As the test ratio increases, the performance of all methods becomes low

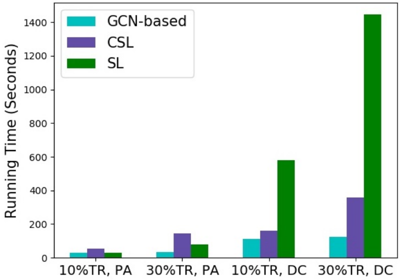
Result and Analysis



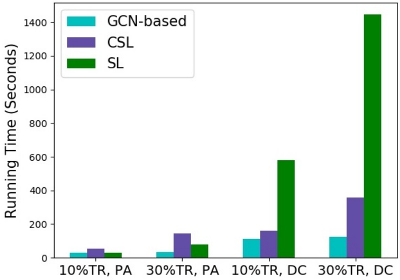
(a) 10% Test Ratio, PA Dataset (b) 30% Test Ratio, PA Dataset

(c) 10% Test Ratio, DC Dataset (d) 30% Test Ratio, DC Dataset

- As the network size grows, the performance of all methods become worse except **GCN-based**
- **GCN-based method outperformed** with **heterogeneous** data (high uncertainty and test ratio).


GCN-based
CSL
SL
Running Time (Seconds)
1400
1200
1000
800
600
400
200
0
10%TR, PA
30%TR, PA
10%TR, DC
30%TR, DC

Result and Analysis



- When the network size increases, the time complexity of SL increases in an **exponential order** while those of GCN-based approach and CSL increase in a **linear order**.
- **GCN-based method outperforms SL and CSL without experiencing performance degradation** from test ratio 10% to 30% since the GCN model is semi-supervised learning, non-sensitive with the growth of the test ratio size.

Conclusion

1. **GCN-based method outperforms** with heterogeneous data that can be effectively handled by the graph convolution.
2. **GCN-based method shows less sensitivity** over a wide range of the uncertainty mass, implying **high resilience**, compared to CSL and SL.
3. The performance order in running time follows: **GCN-based** $>$ CSL $>$ SL, where the running time complexity of the GCN-based model is linear.

Thank You!

Questions?

Reach Xujiang Zhao at
xzhao8@albany.edu
UAB 401, 1215 Western Ave, Albany, NY, USA
University at Albany, SUNY