Deep Learning based Scalable Inference of Uncertain Opinions

IEEE ICDM 2018

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

November 18, 2018

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

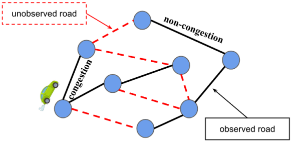
Outline

- Motivation
- Research Problem & Challenge
- Related Work
- Graph Convolutional Networks
- Proposed Approach
- Experimental Results
- Conclusion & Future Work

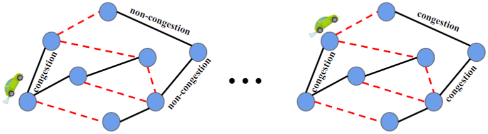
Motivation

- How do we make decisions with subjective, uncertain opinions?
- **Applications**
  - 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)?



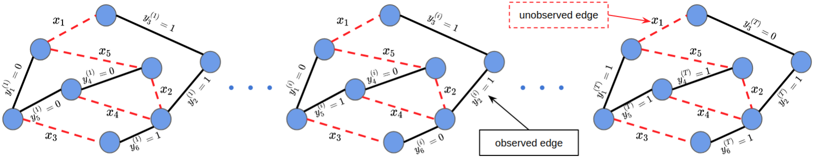
What if we have so many observations?



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 $\mathbf{x}$.



1st observation i observation T observation

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

Contributions

Research goal: Develop a scalable, effective Deep Learning (DL)-based opinion inference algorithm for a large, heterogeneous, uncertain network data.

Key Contributions:

1. Combined non-parametric DL-based algorithm with an opinion formalism of SL to deal with uncertainty of subjective opinions while maximizing prediction accuracy.
2. Proposed a DL-based opinion inference algorithm characterizing uncertainty based on a set of **heterogeneous** belief and uncertainty in a large-scale network data while maximizing prediction accuracy with minimum computation time by leveraging GCN and VAE technology.
3. Validated the proposed DL-based opinion inference algorithm via extensive simulation experiments using real-world datasets.

Binomial 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

Binomial 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

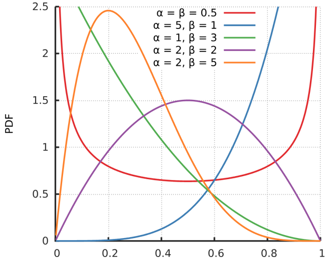
$$b + d + u = 1 \tag{2}$$

SL's Binomial Opinion with Beta Distribution

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

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

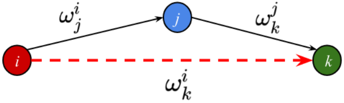
where α is the number of positive evidence and β is the number of negative evidence.



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

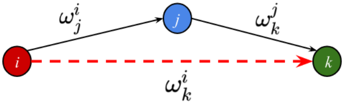
Fusion Operators with Uncertain Opinions in SL

- **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., $w_k^i = w_j^i \otimes w_k^j$

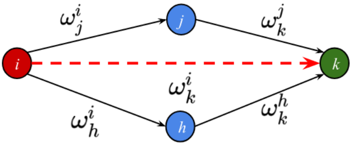


Fusion Operators with Uncertain Opinions in SL

- **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., $w_k^i = w_j^i \otimes w_k^j$



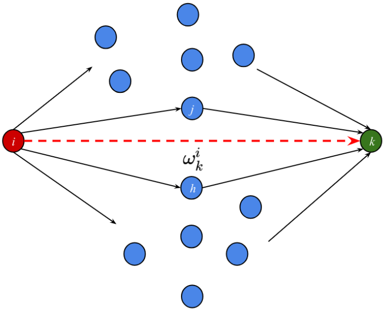
- **Consensus operator,** $\oplus$: Find a consensus between two opinions where two entities observe a same entity, e.g., $w_k^i = (w_j^i \otimes w_k^j) \oplus (w_h^i \otimes w_k^h)$



[Jøsang, Springer 2016]

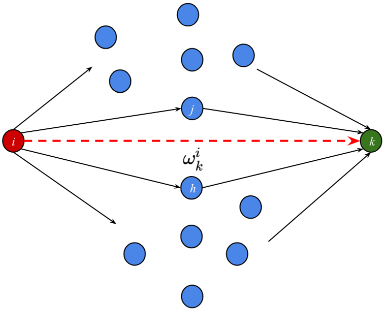
Scalability Issue in Subjective Logic

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

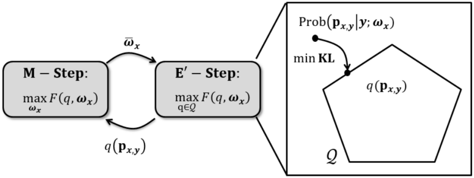
SL's operators are good for fusing two opinions in dyadic relationships; **not scalable for multiple opinions with large network data**.

Collective Subjective Logic (CSL)

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

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

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

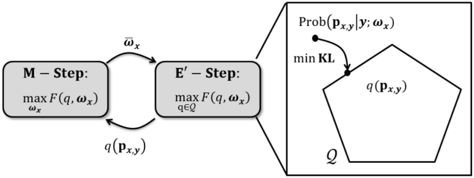


Collective Subjective Logic (CSL)

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

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

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



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?

Both SL and CSL are:

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

How to Solve These Challenge?

Why Deep Learning?

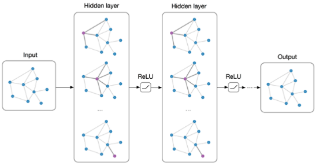
Both SL and CSL are:

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

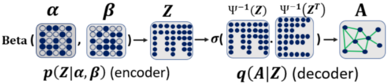
How to Solve These Challenge?

Graph Convolutional Network and **Variational Autoencoder** can provide solutions for

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



[Kipf & Welling, ICLR 2017]



[Kingma & Welling, ICLR 2014]

Graph Convolutional Networks (GCN)

What capability can GCN offer?

- **node level prediction** (regression or classification)

Graph Convolutional Networks (GCN)

What capability can GCN offer?

- **node level prediction** (regression or classification)

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

Graph Convolutional Networks (GCN)

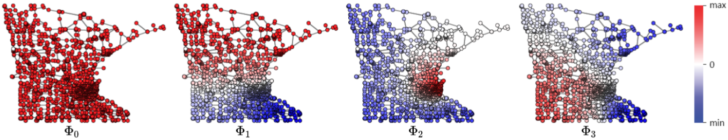
What capability can GCN offer?

- **node level prediction** (regression or classification)

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

Graph Fourier Transform:

- Euclidean spaces: $\mathbf{r} = \sum_{k \geq 0} \hat{\mathbf{r}}_k e^{ik}$
- 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 **r** and **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 **r** and **b** on graph, **graph convolution**

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

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

- **Graph convolutional layer**

$$g_\theta \star \mathbf{r} = \Phi g_\theta \Phi^T \mathbf{r}. \quad (5)$$

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

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

Graph Convolution

- Given two signals **r** and **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 **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

How to use GCN model to inference opinion?

- How many convoluation layers could be sufficinet?

Proposed Approach: GCN-based Uncertain Opinion Prediction

How to use GCN model to inference opinion?

- How many convoluation layers could be sufficinet?
  Two convolution layer have less risk of overfitting.

Proposed Approach: GCN-based Uncertain Opinion Prediction

How to use GCN model to inference opinion?

- How many convoluation layers could be sufficinet?
  Two convolution layer have less risk of overfitting.
- How to model the heterogeneous dependencies among Belief and Uncertainty?

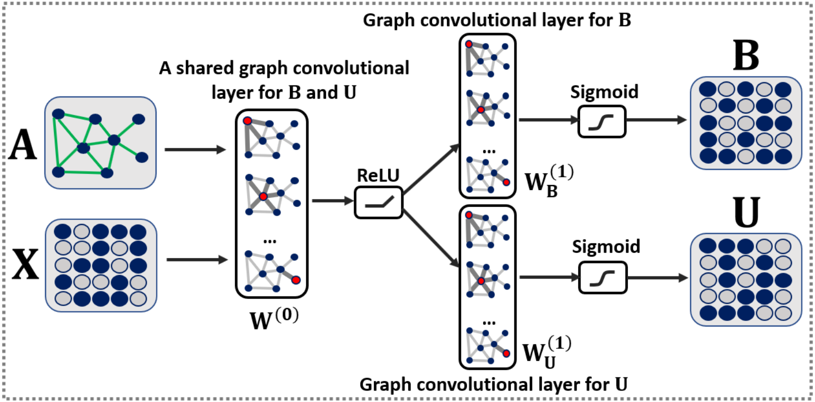
Proposed Approach: GCN-based Uncertain Opinion Prediction

How to use GCN model to inference opinion?

- How many convoluation layers could be sufficinet?
  Two convolution layer have less risk of overfitting.
- How to model the heterogeneous dependencies among Belief and Uncertainty?
  - heterogeneous structural dependencies among node-level belief and uncertainty,
  - inherent relational dependencies between belief and uncertainty.

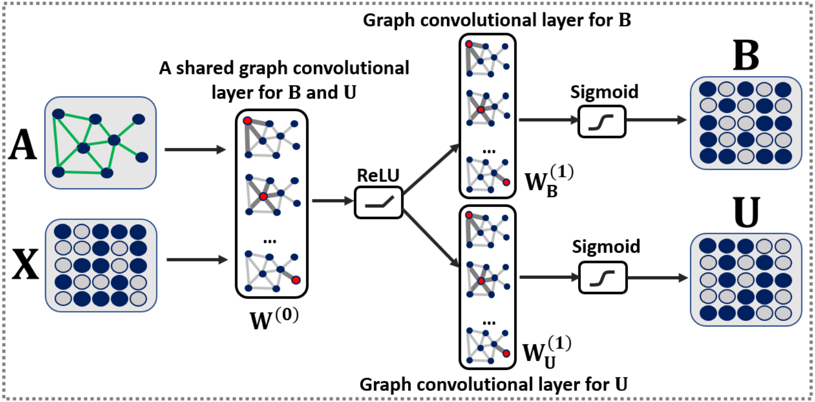
GCN-based opinion model

- Model higher order shared structural information for **B** and **U**.
- Capture their own heterogeneous structural dependencies.
- Loss function: $\mathcal{L}(\theta) = \mathcal{L}_{\mathbf{B}(\theta)} + \mathcal{L}_{\mathbf{U}(\theta)}$



GCN-based opinion model

- Model higher order shared structural information for **B** and **U**.
- Capture their own heterogeneous structural dependencies.
- Loss function: $\mathcal{L}(\theta) = \mathcal{L}_{\mathbf{B}(\theta)} + \mathcal{L}_{\mathbf{U}(\theta)}$



Lack of effectively handling the inherent relational dependencies between **B** and **U**

VAE-based opinion model

- **Transform** the combinations (opinions) of **B** and **U** to their equivalent **Beta PDF**
- Latent probability variables for each node $i \in \mathbb{V}$ that are sampled from the Beta PDF: $z_{i,j} \sim \text{Beta}(\alpha_i, \beta_i),\ j = 1, \cdots, P$.

VAE-based opinion model

- **Transform** the combinations (opinions) of **B** and **U** to their equivalent **Beta PDF**
- Latent probability variables for each node $i \in \mathbb{V}$ that are sampled from the Beta PDF: $z_{i,j} \sim \text{Beta}(\alpha_i, \beta_i),\ j = 1, \cdots, P$.

Encoder: graph structural information encoded by node-level opinions

$$q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} \prod_{j=1}^{P} \text{Beta}(z_{i,j}|\alpha_i, \beta_i),$$

VAE-based opinion model

- **Transform** the combinations (opinions) of **B** and **U** to their equivalent **Beta PDF**
- Latent probability variables for each node $i \in \mathbb{V}$ that are sampled from the Beta PDF: $z_{i,j} \sim \mathrm{Beta}(\alpha_i, \beta_i),\ j = 1, \cdots, P$.

Encoder: graph structural information encoded by node-level opinions

$$q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} \prod_{j=1}^{P} \mathrm{Beta}(z_{i,j}|\alpha_i, \beta_i),$$

Decoder: use the latent variables **Z** to recover the structural information in adjacency matrix **A**

$$p(\mathbf{A}|\mathbf{Z}) = \prod_{i=1}^{N} \prod_{j=\mathcal{N}_i} p(A_{i,j}|\mathbf{z}_i, \mathbf{z}_j),$$

where $p(A_{i,j} = 1|\mathbf{Z}) = \sigma([\psi^{-1}(\mathbf{z}_i)]^T[\psi^{-1}(\mathbf{z}_j)])$, $\psi^{-1}(\cdot)$ is the reverse CDF that converts a probability to a real value.

VAE-based opinion model

- **Transform** the combinations (opinions) of **B** and **U** to their equivalent **Beta PDF**
- Latent probability variables for each node $i \in \mathbb{V}$ that are sampled from the Beta PDF: $z_{i,j} \sim \text{Beta}(\alpha_i, \beta_i),\ j = 1, \cdots, P$.

Encoder: graph structural information encoded by node-level opinions

$$q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} \prod_{j=1}^{P} \text{Beta}(z_{i,j}|\alpha_i, \beta_i),$$

Decoder: use the latent variables **Z** to recover the structural information in adjacency matrix **A**

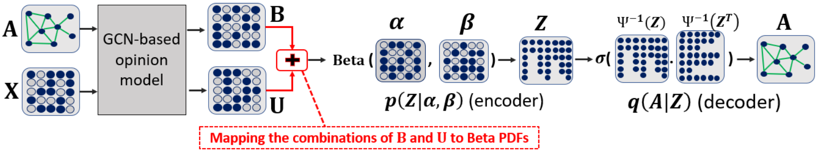
$$p(\mathbf{A}|\mathbf{Z}) = \prod_{i=1}^{N} \prod_{j=\mathcal{N}_i} p(A_{i,j}|\mathbf{z}_i, \mathbf{z}_j),$$

where $p(A_{i,j} = 1|\mathbf{Z}) = \sigma([\psi^{-1}(\mathbf{z}_i)]^T [\psi^{-1}(\mathbf{z}_j)])$, $\psi^{-1}(\cdot)$ is the reverse CDF that converts a probability to a real value.

Negative variational lower bound $\mathcal{L}$:

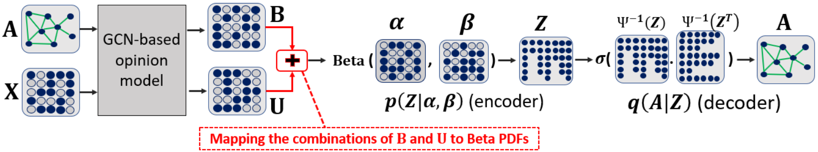
$$\tilde{\mathcal{L}} = -\mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{A})}\Big[\log p(\mathbf{A}|\mathbf{Z})\Big] + \text{KL}\Big[q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \| p(\mathbf{Z})\Big]$$

GCN-VAE based inference model



- Combine GCN-based opinion model and VAE-based opinion model,

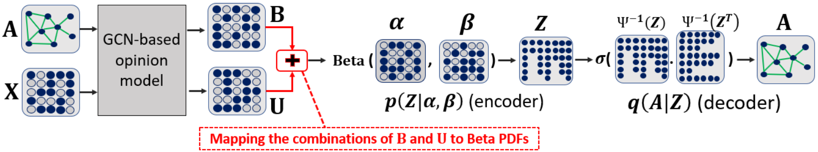
GCN-VAE based inference model



- Combine GCN-based opinion model and VAE-based opinion model,
- Jointly optimize the loss functions of these two model,

$$\min_{\theta} \lambda \tilde{\mathcal{L}}(\theta) + \mathcal{L}_{\mathbf{B}(\theta)} + \mathcal{L}_{\mathbf{U}(\theta)} \tag{8}$$

GCN-VAE based inference model



- Combine GCN-based opinion model and VAE-based opinion model,
- Jointly optimize the loss functions of these two model,

$$\min_{\theta} \lambda\tilde{\mathcal{L}}(\theta) + \mathcal{L}_{\mathbf{B}(\theta)} + \mathcal{L}_{\mathbf{U}(\theta)} \tag{8}$$

- *"reparameterization trick"*: Beta distribution not differentiable. Instead, *Kumaraswamy* distribution:

$$\tilde{q}(z|\alpha,\beta) = \alpha\beta(z)^{\alpha-1}(1-z^{\alpha})^{\beta-1} \tag{9}$$

Reparameterization: $z \sim (1-u^{1/\beta})^{1/\alpha}$, where $u \sim \text{Unif}(0,1)$

Datasets & Experimental Setting

- Road traffic datasets:

Dataset	# nodes	# edges	# weeks	# snapshots in total
Epinions	47,676	477,468	-	-
D.C.	1,383	1,878	43	3440
Philadelphia	603	708	43	3440

Datasets & Experimental Setting

- Road traffic datasets:

Dataset	# nodes	# edges	# weeks	# snapshots in total
Epinions	47,676	477,468	-	-
D.C.	1,383	1,878	43	3440
Philadelphia	603	708	43	3440

- Parameter settings:
  - Time window size: $T \in \{3, 6, 8, 11, 38\}$,
  - Uncertainty mass values: $u \in \{40\%, 25\%, 20\%, 15\%, 5\%\}$
  - Test Ratio: $TR \in \{20\%, 40\%, 60\%, 80\%\}$

Datasets & Experimental Setting

- Road traffic datasets:

Dataset	# nodes	# edges	# weeks	# snapshots in total
Epinions	47,676	477,468	-	-
D.C.	1,383	1,878	43	3440
Philadelphia	603	708	43	3440

- Parameter settings:
  - Time window size: $T \in \{3, 6, 8, 11, 38\}$,
  - Uncertainty mass values: $u \in \{40\%, 25\%, 20\%, 15\%, 5\%\}$
  - Test Ratio: $TR \in \{20\%, 40\%, 60\%, 80\%\}$
- Performance metrics:

$$\text{B-MSE}(\boldsymbol{\omega}_{\mathbb{V}\setminus\mathbb{L}}) = \frac{1}{N} \sum\nolimits_{i \in \mathbb{V}\setminus\mathbb{L}} |b_i - b_i^\star| \quad (10)$$

$$\text{U-MSE}(\boldsymbol{\omega}_{\mathbb{V}\setminus\mathbb{L}}) = \frac{1}{N} \sum\nolimits_{i \in \mathbb{V}\setminus\mathbb{L}} |u_i - u_i^\star| \quad (11)$$

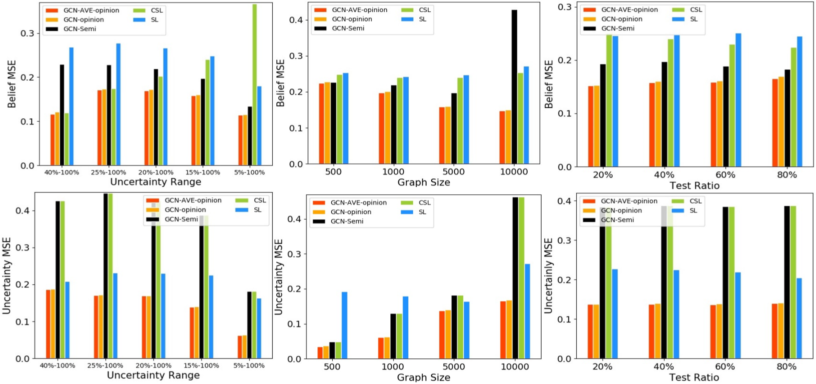
- Computation time metric: seconds

Comparing Schemes

Comparison Methods:

- Our proposed: **GCN-opinion** and **GCN-AVE-opinion**
- GCN-Semi: Semi-supervised node classification
- CSI: combining Probabilistic Soft Logic (PSL) and Markov Random Fields (MRFs) with SL
- SL: Subjective Logic inference based on Discount and Consensus operator

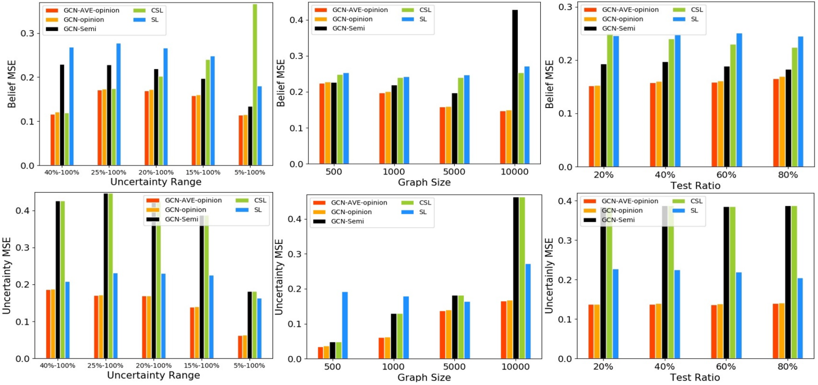
Results with Epinions Dataset



Effect of Uncertainty:

- Belief-MSE: **GCN-AVE** > GCN > CSL > GCN-Semi > SL
- Uncertainty-MSE: **GCN-AVE** ≈ GCN > SL > GCN-Semi ≈ CSL
- GCN-AVE and GCN **better** performance under **larger ranges of uncertainties.**

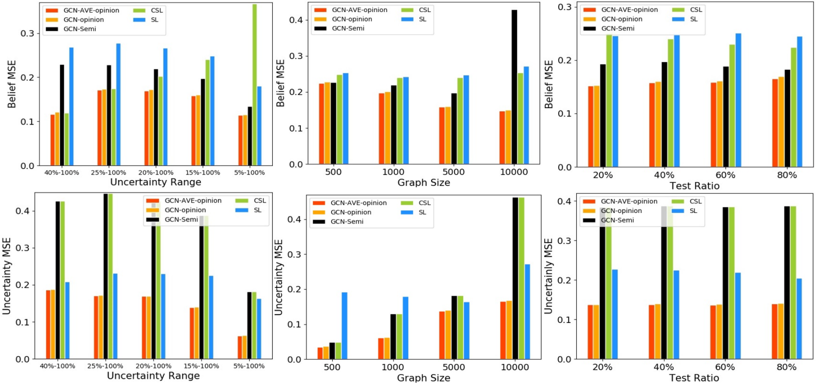
Results with Epinions Dataset



Effect of Graph Size:

- Belief-MSE: **GCN-AVE** > GCN > GCN-Semi > CSL > SL
- Uncertainty-MSE: **GCN-AVE** > GCN > GCN-Semi ≈ CSL > SL

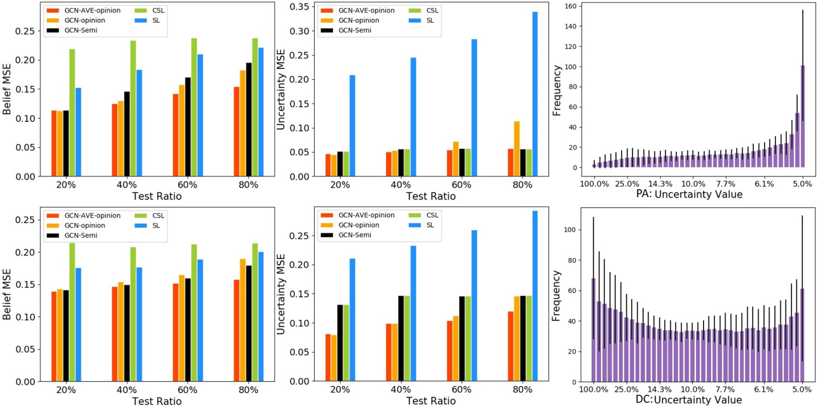
Results with Epinions Dataset



Effect of Test Ratio:

- Belief-MSE: **GCN-AVE** > GCN > GCN-Semi > CSL > SL
- Uncertainty-MSE: **GCN-AVE** ≈ GCN > SL > GCN-Semi ≈ CSL
- GCN-AVE and GCN **less sensitivity** under different test ratios

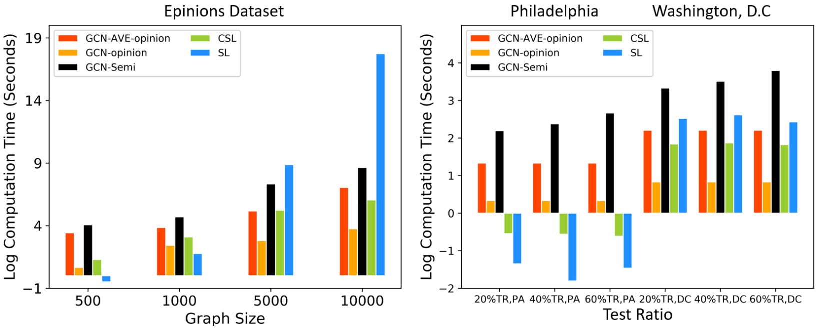
Result on Traffic Dataset



Effect of Test Ratio:

- Belief-MSE: **GCN-AVE** > GCN-Semi > GCN > SL > CSL
- Uncertainty-MSE: **GCN-AVE** > GCN > GCN-Semi ≈ CSL > SL

Computation Time Analysis



- **Computation order**: **GCN** $<$ CSL $<$ GCN-AVE $<$ GCN-Semi $<$ SL
- SL increases in an **exponential order** while others(GCN-AVE, GCN, CSL) increase in a **linear order**.

Conclusion

1. GCN-AVE method **outperforms** among all in both B-MSE and U-MSE.
2. GCN-AVE method shows **less sensitivity** over a wide range of test ratios.
3. GCN-AVE performed better than GCN because GCN-AVE integrates an VAE-based opinion model to consider the **inherent relational dependencies** between beliefs and uncertainties.
4. GCN-AVE scales almost **linearly** in proportion to the network size and is scalable for large-scale network data.

Thank You!

Questions?

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