Graph-Based Semi-Supervised Learning with Nonignorable Nonresponses

Fan Zhou
School of Statistics and Management
Shanghai University of Finance and Economics

Northeast Normal University

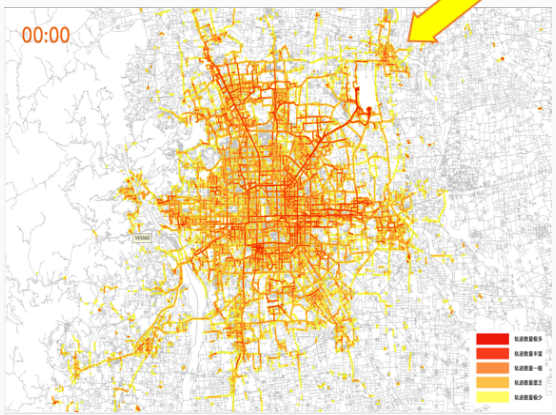
Oct 21st, 2019

Graph-Based Semi-Supervised Learning with Nonignorable Nonresponse

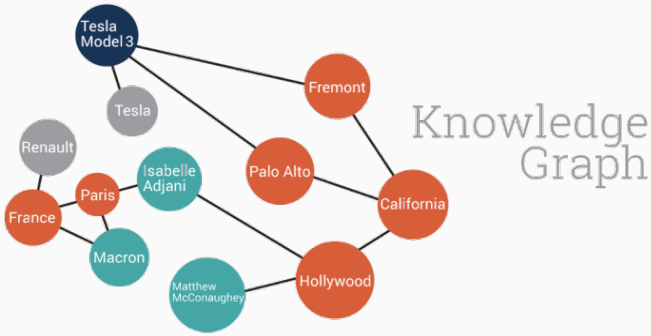
Graph-Based Data

Brain Connectivity

Transportation



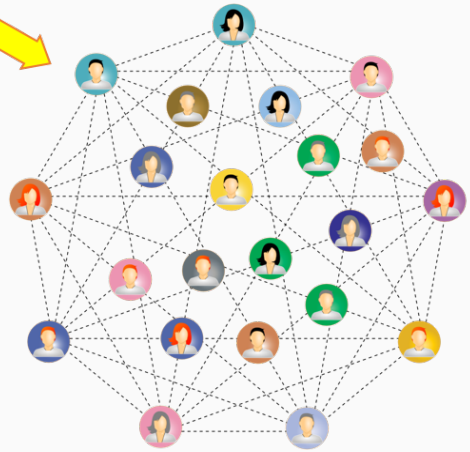
Graphs/Networks



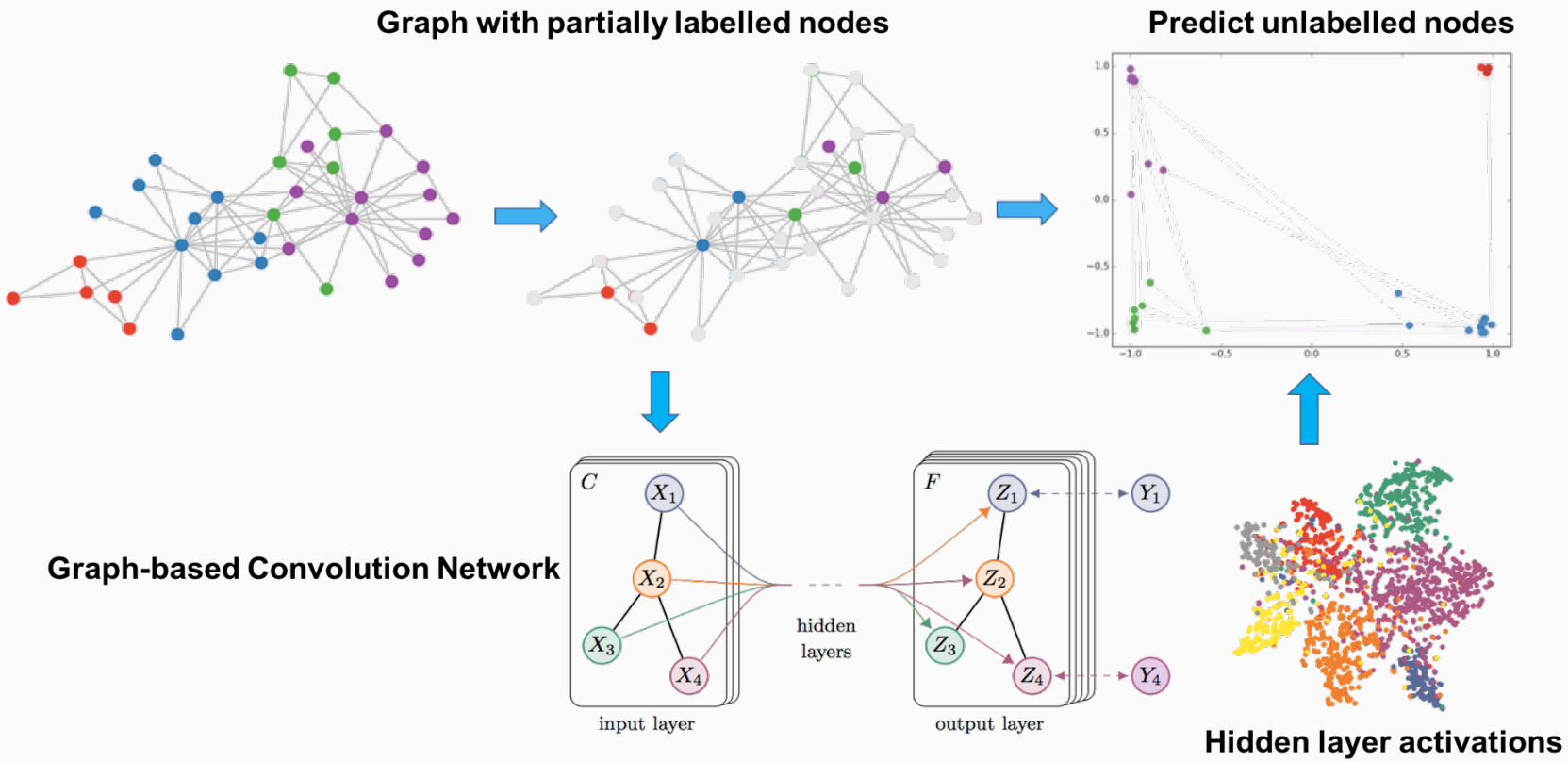
Knowledge Graph

Knowledge Graph

Social Network

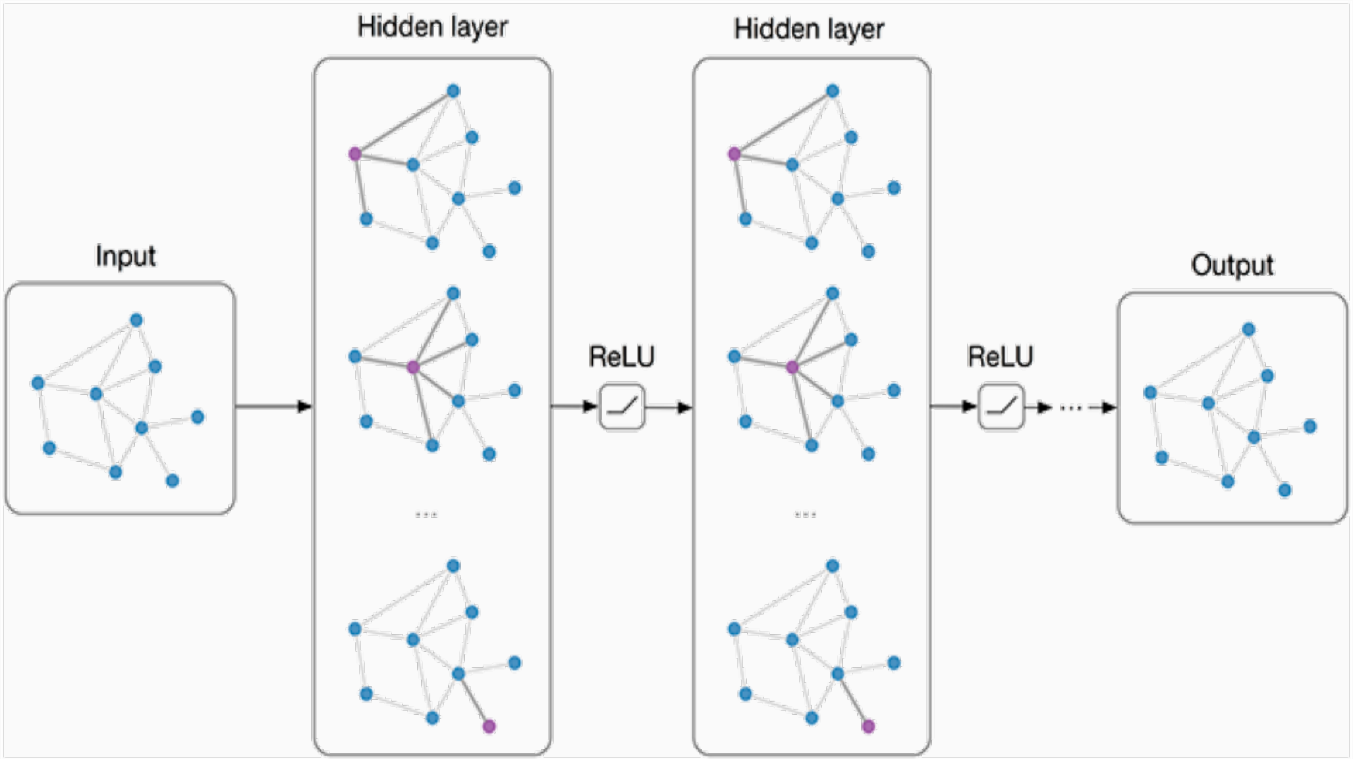


Graph-Based Semi-Supervised Learning



Graph Convolution Network

Thomas N. Kipf, Max Welling ICLR 2017



The layer-wise propagation of GCN is defined as

$$H^{(l+1)} = f(H^{(l)}, A) = \sigma(\widehat{D}^{-\frac{1}{2}}\widehat{A}\widehat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}), \quad (1)$$

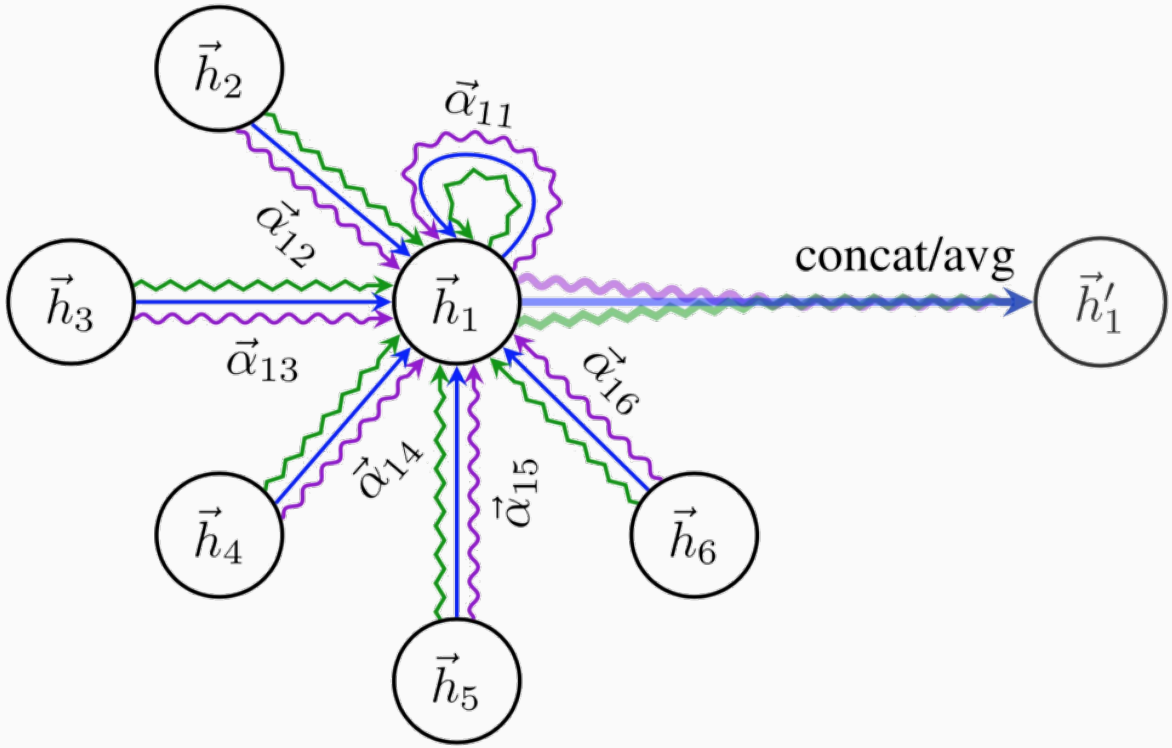
where $\widehat{A} = A + I$, and $\widehat{D}$ is the diagonal vertex degree matrix of $\widehat{A}$.

Graph Convolution Network

Semi-supervised classification with GCNs: Latent space dynamics for 300 training iterations. Labeled nodes are highlighted.

Graph Attention Network

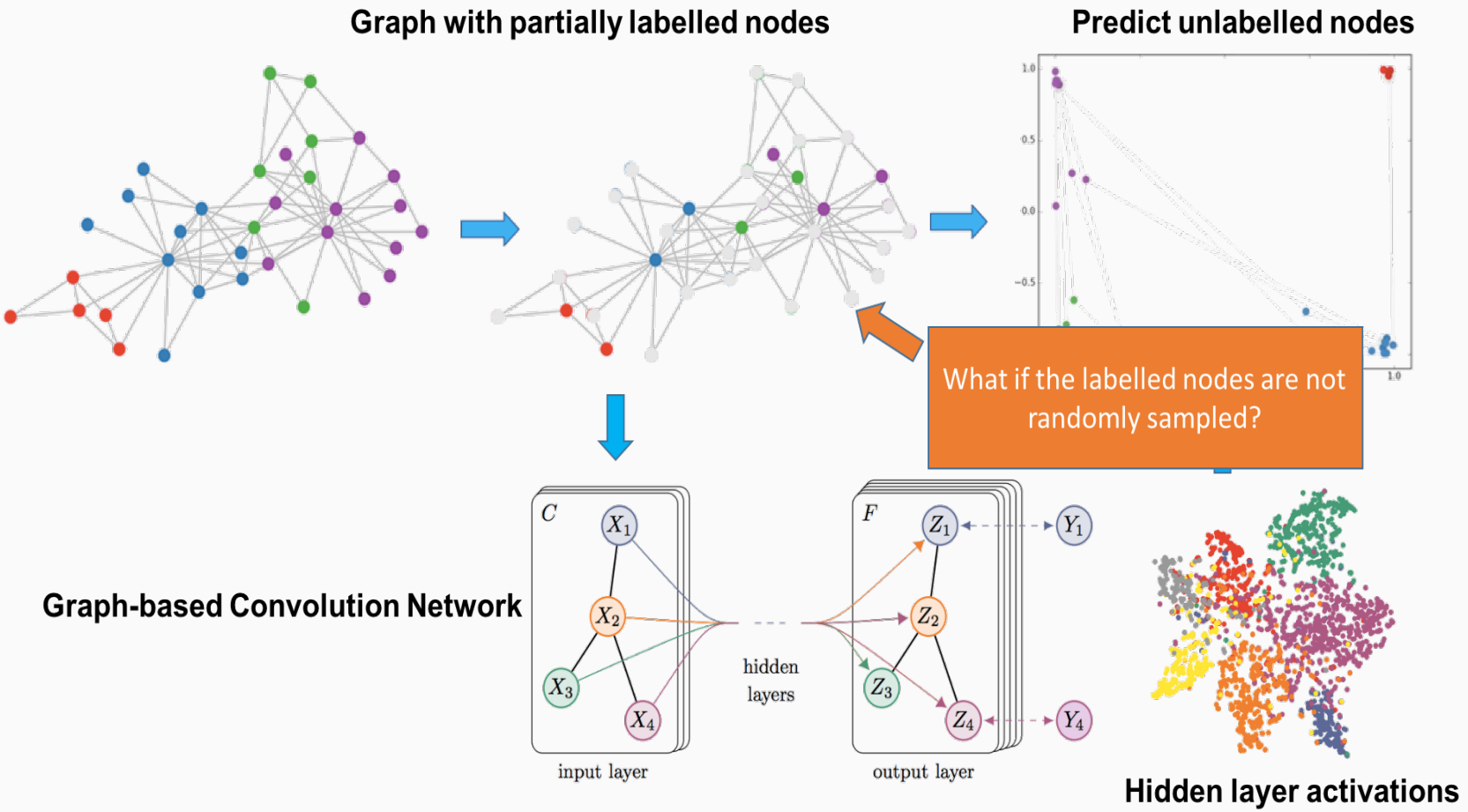
Velickovic et al., ICLR 2018



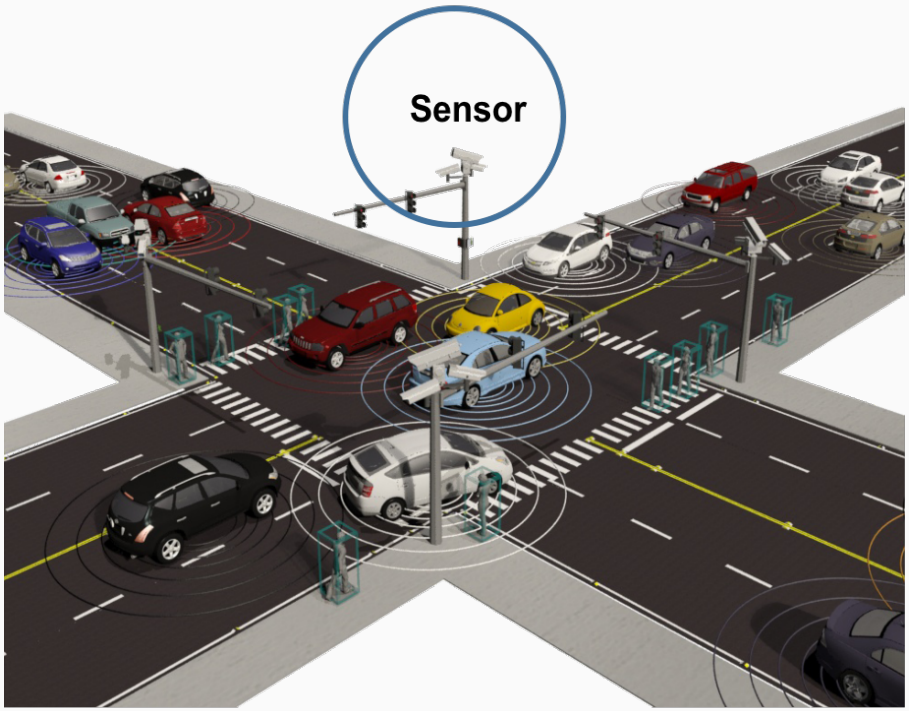
The coefficients is computed by the attention mechanism

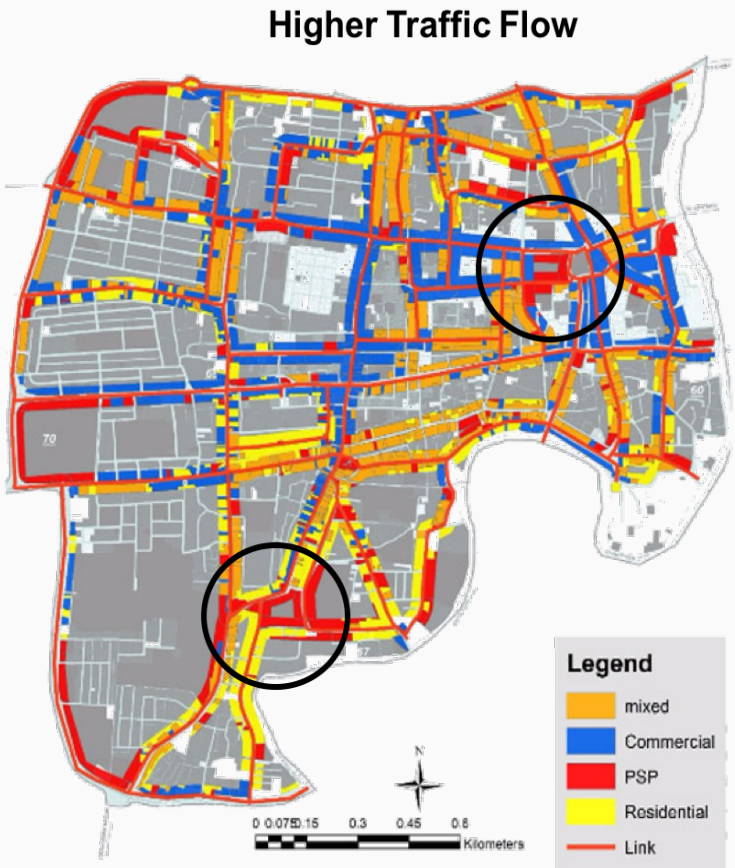
$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T[W\vec{h}_i^T || W\vec{h}_j]))}{\sum_{k\in N_i}\exp(\text{LeakyReLU}(\vec{a}^T[W\vec{h}_i^T || W\vec{h}_k]))} \quad (2)$$

Nonignorable Nonresponse

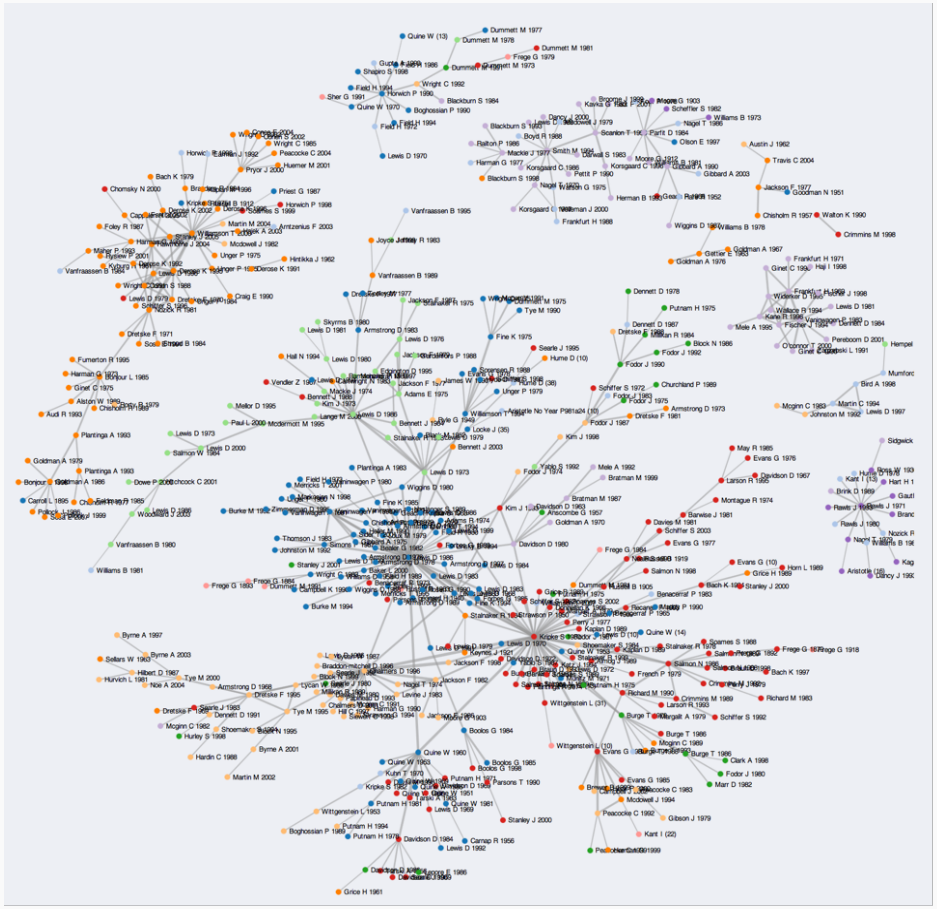


Nonignorable Nonresponse



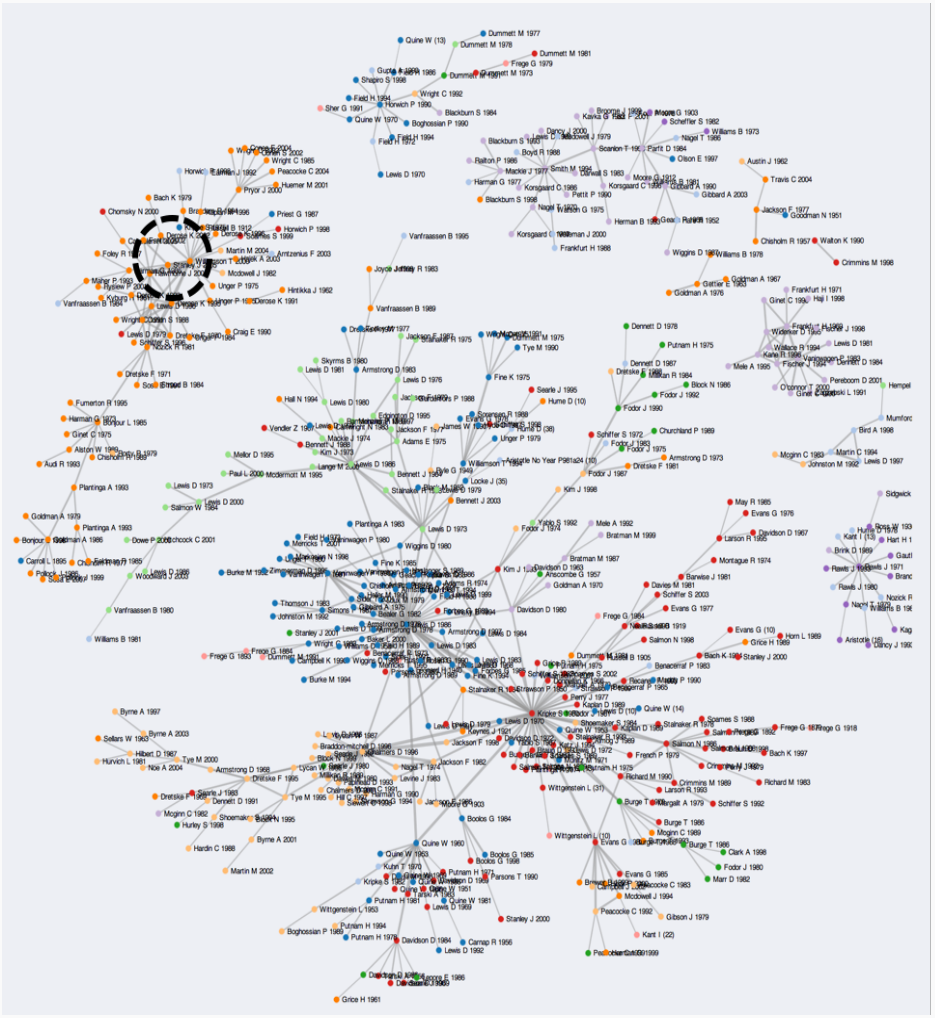


Nonignorable Nonresponse



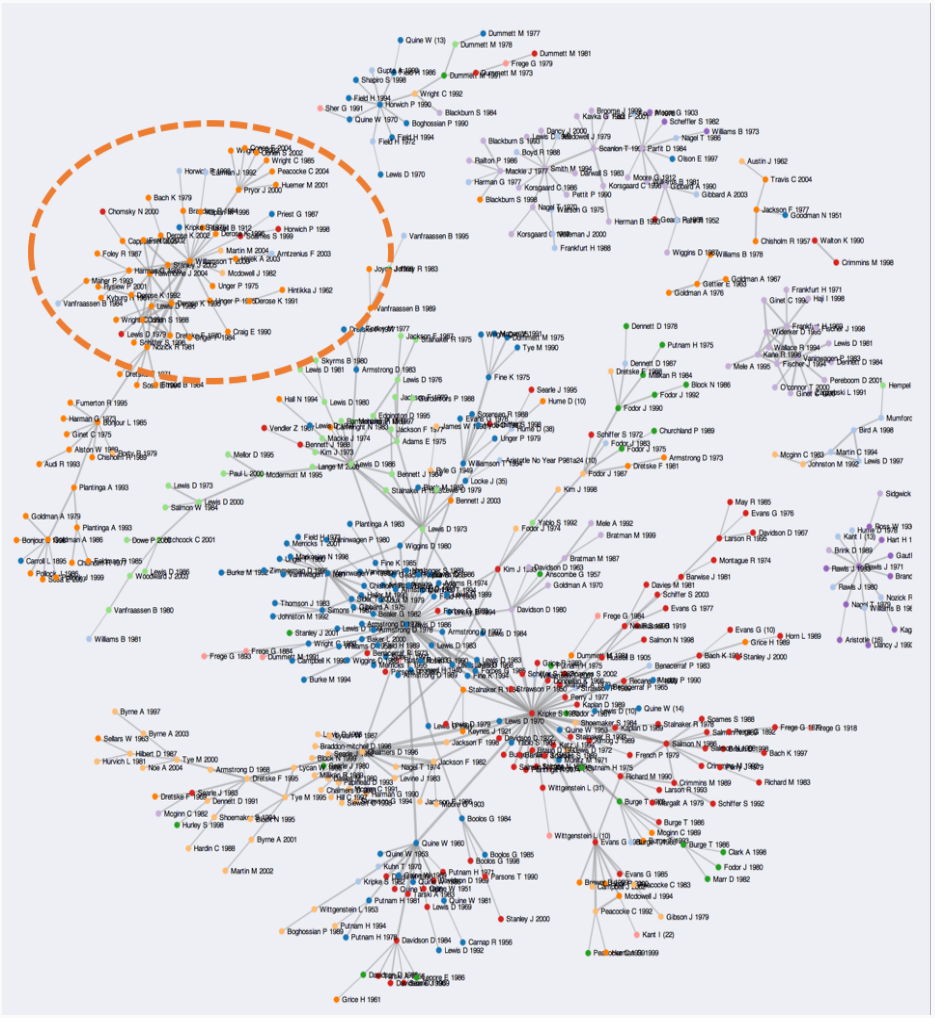
Citation Network of Philosophy

Nonignorable Nonresponse



Jason Stanley
Yale University

Nonignorable Nonresponse



Problem Statement

Target:

Our target is to propose a Graph-based joint model with Nonignorable Missingness (**GNM**) based on the observed data to predict all the unlabelled nodes.

Notations

Let $G = (V, E, A)$ be a weighted graph consists of $|V| = N$ vertexes; $A \in R^{N \times N}$ is the adjacency matrix. We introduce some important notations as follows:

(i). $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T \in R^{N \times p}$ is a fully observed input feature matrix of size $N \times p$.

(ii). $Y = (y_1, y_2, \ldots, y_N)^T$ is a vector of vertex responses, which is partially observed subject to missingness.

(iii). $A \in R^{N \times N}$ is the adjacency matrix (binary or weighted), which encodes node similarity and network connectivity.

(iv). $r_i \in \{0, 1\}$ is a "labeling indicator", that is y_i is observed if and only if $r_i = 1$. Let $R = \{1, \ldots, n\}$ denote the set of labelled vertexes and $R^c = \{n + 1, \ldots, N\}$ defines the subsample of non-respondents for which the vertex label is missing.

Model Description

We consider non-ignorable missingness, where the indicator variable r_i depends on y_i (which is unobserved when $r_i = 0$). It is assumed that r_i follows a Bernoulli distribution:

$$r_i|(y_i, h(x_i;\theta_h)) \sim \text{Bernoulli}(\pi_i), \tag{3}$$

where $\pi(y_i, h(x_i;\theta_h)) = P(r_i = 1|y_i, h(x_i;\theta_h))$ is the probability of missingness for y_i. Furthermore, an exponential tilting model is proposed for π_i as follows:

$$\pi(y_i, h(x_i;\theta_h)) = \pi(y_i, h(x_i;\theta_h);\alpha_r,\gamma,\phi) = \frac{\exp\{\alpha_r + \gamma^T h(x_i;\theta_h) + \phi y_i\}}{1+\exp\{\alpha_r + \gamma^T h(x_i;\theta_h) + \phi y_i\}}. \tag{4}$$

Our question of interest is to unbiasedly learn an outcome model $Y|\mathbf{x}$. Without loss of generality, when y is continuous, we consider a linear model given by

$$Y = \alpha + \mathscr{G}^A(\mathbf{x};\theta_g)\beta + \epsilon, \tag{5}$$

where $\mathscr{G}^A(\mathbf{x};\theta_g)$ denotes an unknown function of $\mathbf{x}$, which can be a deep neural network incorporating the network connectivity A.. When y is a discrete variable:

$$P(y_i = k|\mathscr{G}^A(\mathbf{x};\theta_g)_i;\alpha_k,\beta_k) = \exp(\alpha_k + \beta_k^T \mathscr{G}^A(\mathbf{x};\theta_g)_i)/\sum_{j=1}^{K}\exp(\alpha_j + \beta_j^T \mathscr{G}^A(\mathbf{x};\theta_g)_i) \tag{6}$$

Identifiability

The joint probability density function (pdf) of the observed data is given by

$$\prod_i [P(r_i = 1|y_i, h(x_i;\theta_h))f(y_i|\mathscr{G}^A(\mathbf{x};\theta_g)_i)]^{r_i}[1-\int P(r_i = 1|y, h(x_i;\theta_h))f(y|\mathscr{G}^A(\mathbf{x};\theta_g)_i)dy]^{1-r_i}. \tag{7}$$

The traditional identifiability fails when $h(x_i;\theta_h)$ or $\mathscr{G}^A(\mathbf{x};\theta_g)$ has ReLU structure.

Therefore, we introduce a novel identifiability. We call (θ_y, θ_r) is **equivalent** to (θ'_y, θ'_r), denoted by $(\theta_y, \theta_r) \sim (\theta'_y, \theta'_r)$, if

$$\gamma^T h(x;\theta_h) = \gamma'^T h(x;\theta'_h), \text{ and } \mathscr{G}^A(\mathbf{x};\theta_g)\beta = \mathscr{G}^A(\mathbf{x};\theta'_g)\beta'. \tag{8}$$

holds and $\alpha' = \alpha, \alpha'_r = \alpha_r$, $\phi' = \phi$, where $\theta_y = (\alpha, \beta, \theta_g), \theta_r = (\alpha_r, \gamma, \phi, \theta_h)$. The equivalence class of an element (θ_y, θ_r) is denoted by $[\![(\theta_y, \theta_r)]\!]$, defined as the set

$$[\![(\theta_y, \theta_r)]\!] = \{(\theta'_y, \theta'_r) \in \mathcal{D}(\theta_y) \otimes \mathcal{D}(\theta_r) | (\theta'_y, \theta'_r) \sim (\theta_y, \theta_r)\},$$

and the set of all equivalent classes is called the **Prediction-Equivalent Quotient** (PEQ) space, denoted by $S = \mathcal{D}(\theta_y) \otimes \mathcal{D}(\theta_r)/\sim$. The GNM model is called identifiable on the PEQ space iff that

$$f(y|\mathscr{G}^A(\mathbf{x})_i;\theta_y)P(r = 1|y, h(x_i);\theta_r) = f(y|\mathscr{G}^A(\mathbf{x})_i;\theta'_y)P(r = 1|y, h(x_i);\theta'_r)$$

holds for all $\mathbf{x}, y$ implies $(\theta_y, \theta_r) \sim (\theta'_y, \theta'_r)$.

Identifiability

An example to illustrate how the traditional identifiability fails with ReLu

$$\text{Logit}[P(r_i = 1|y_i, h(z_i; \beta_r)); \gamma] = \alpha_r + \gamma \text{Relu}(z_i \beta_r) + \phi y_i = \text{Logit}[P(r_i = 1|y_i, h(z_i; 2\beta_r)); \gamma/2].$$

Theorem

Assume three conditions as follows.

(A1) For all θ_g, there exist $(\mathbf{x}_1, \mathbf{x}_2)$ such that $\mathscr{G}^A(\mathbf{x}_1; \theta_g)_i \neq \mathscr{G}^A(\mathbf{x}_2; \theta_g)_i$ for each i; $\beta \neq 0$ holds.

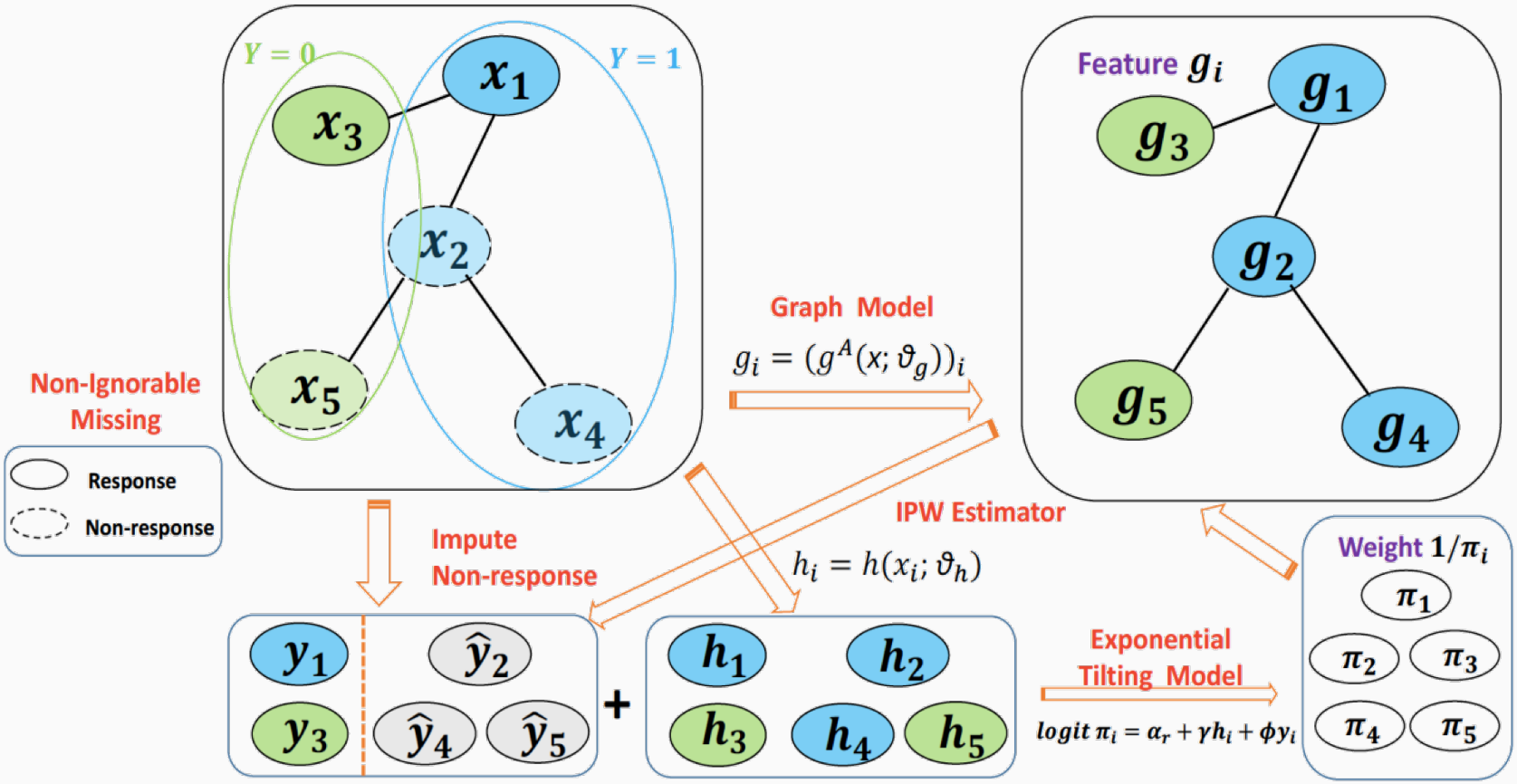
(A2) For all θ_g and $\mathbf{z}$, there exists $(\mathbf{u}_1, \mathbf{u}_2)$ such that $\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_g)_i \neq \mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_g)_i$ for each i; and $\beta \neq 0$ holds.

(A3) For all θ_h, there exists (z_1, z_2) such that $h(z_1; \theta_h) \neq h(z_2; \theta_h)$; and $\gamma \neq 0$ holds.

The GNM model (4) and (6) is identifiable on the PEQ space under Condition (A1). Suppose that there exists an instrumental variable $\mathbf{u}$ in $\mathbf{x} = [\mathbf{z}, \mathbf{u}]$ such that $f(y_i|\mathscr{G}^A(\mathbf{x})_i)$ depends on $\mathbf{u}$, whereas $P(r_i = 1|y_i, h(x_i))$ does not. Then the GNM model (4) and (5) is identifiable on the PEQ space under Conditions (A2) and (A3).

Estimation

General Picture of the Joint Estimation Approach



Estimation

We propose a doubly robust (DR) estimation approach to alternatively obtain the Inverse Probability Weighted Estimator (IPWE) of θ_y and imputation estimator of θ_r.

Inverse Probability Weighted Estimator (IPWE) of θ_y

With $\pi(y_i, h(x_i); \theta_r)$ estimated by $\pi(y_i, h(x_i); \widehat{\theta}_r)$, the Inverse Probability Weighted Estimator (IPWE) of θ_y can be obtained by minimizing the weighted cross-entropy loss

$$\mathscr{L}_1(\theta_y|\widehat{\theta}_r) = -\sum_i \frac{r_i}{\pi(y_i, h(x_i); \widehat{\theta}_r)} \sum_{k=1}^{K} 1(y_i = k)\log(P(y_i = k|\mathscr{G}^A(\mathbf{x})_i; \theta_y)) \quad (9)$$

or by minimizing the weighted mean squared error (MSE) when y is continuous.

Imputation estimator of θ_r

With the estimated $f(Y|\mathscr{G}^A(\mathbf{x}; \widehat{\theta}_g))$, we could obtain an estimator of θ_r by minimizing

$$\widetilde{\mathscr{L}_2}(\theta_r|\widehat{\theta}_y) = -\sum_{r_i=1} \ln(\pi(y_i, h(x_i); \theta_r)) - \sum_{r_i=0} \log(1 - B^{-1} \sum_{y_{ib} \sim f(y|\mathscr{G}^A(\mathbf{x})_i; \widehat{\theta}_y)} \pi(y_{ib}, h(x_i); \theta_r)), \quad (10)$$

where $\{y_{ib}\}_{b=1}^{B} \overset{iid}{\sim} f(y|\mathscr{G}^A(\mathbf{x})_i; \widehat{\theta}_y)$.

Estimation

The details of the algorithm are described in five steps as follows:

1. Determine the initial value of the response probability $\pi_i^{(0)}$ (or $\theta_r^{(0)}$). For example, we can let $\pi_i^{(0)} = 1$ for all the labelled vertexes ($r_i = 1$).
2. Let $e = 1$, where e represents the number of epoch. We update θ_y based on $\pi_i^{(0)}$ obtained from the previous epoch by minimizing the loss function in (9) using GD. At the i-th iteration within the e-th epoch, we update θ_y as follows:
$$\theta_y^{(e,i+1)} \leftarrow \theta_y^{(e,i)} - \gamma_0 \nabla_{\theta_y} \mathscr{L}_1(\theta_y | \theta_r^{(e-1)}), \tag{11}$$
3. Impute y_i for all the unlabelled nodes $r_i = 0$ using $y_i^{(e)} = \beta_0^{(e)} + \mathscr{G}^A(\mathbf{x}; \theta_g^{(e)})_i^T \beta_1^{(e)}$ for the continuous case and sampling $y_i^{(e)}$ from distribution $P(y_i | \mathscr{G}^A(\mathbf{x})_i; \theta_y^{(e)})$ otherwise.
4. We use GD to update θ_r. Specifically, at the j-th iteration, we have
$$\theta_r^{(e,j+1)} \leftarrow \theta_r^{(e,j)} - \gamma_1 \nabla_{\theta_r} \widetilde{\mathscr{L}_2}(\theta_r | \theta_y^{(e)}) \tag{12}$$
with the initial start $\theta_r^{(e,0)}$ equal to $\theta_r^{(e-1)}$. After convergence, we can get the estimate of θ_r denoted as $\theta_r^{(e)}$ and update the sampling weight $\pi_i^{(e)}$ based on $P(r_i = 1 | y_i, h(x_i); \theta_r^{(e)})$ for all labelled vertexes.
5. Stop once convergence achieved, otherwise let $e = e + 1$ and return to step 3.

Simulation

We consider a citation network 'Cora' generated by $|V| = 2708$ documents with a binary adjacency matrix A. The node response is simulated from the following model:

$$y_i = \beta_0 + \beta_1^T \mathscr{G}^A(\mathbf{x})_i + \epsilon_i, \tag{13}$$

where $\epsilon_i \sim N(0, \sigma^2)$ and $\mathscr{G}^A(\mathbf{x})$ is the output of a 2-layer GCN model. We let response probability π depend on the unobserved vertex response y only , such that

$$\pi_i \equiv P(r_i = 1|y_i) = \frac{\exp\{\alpha_r + \phi y_i\}}{1 + \exp\{\alpha_r + \phi y_i\}}. \tag{14}$$

$\bar{p}$	σ	Method	Metric	Mean	SD
4	0.5	SM	RMSE	1.1925	6.43e-1
			MAPE	0.2932	2.01e-1
		GNM	RMSE	0.6983	1.28e-2
			MAPE	0.1995	1.00e-2
	1	SM	RMSE	1.6185	8.58e-2
			MAPE	0.3104	4.73e-2
		GNM	RMSE	1.2103	4.81e-2
			MAPE	0.2263	2.28e-2
16	0.5	SM	RMSE	0.7923	9.94e-2
			MAPE	0.2014	2.42e-2
		GNM	RMSE	0.6015	2.17e-2
			MAPE	0.1672	1.90e-2
	1	SM	RMSE	1.4212	2.14e-1
			MAPE	0.2129	1.05e-2
		GNM	RMSE	1.1316	6.04e-2
			MAPE	0.1849	4.62e-3

Table 1: Mean RMSEs and MAPEs by GNM and SM based on simulated data sets

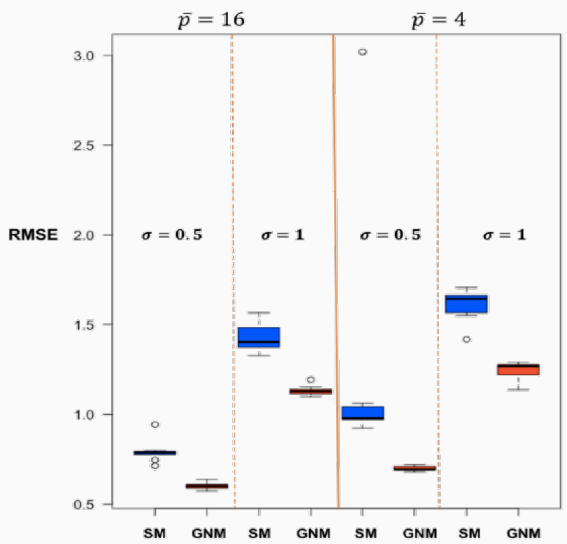


Figure 2: Boxplot of RMSEs in real data analysis

Real Analysis

We modify the Cora to a binary-class data by merging the six non-'Neural Network' classes together. The global prevalence of two new classes are $(0.698, 0.302)$ with $N_0 = \#\{y = 0\} = 1890$ and $N_1 = \#\{y = 1\} = 818$, respectively. Two missing mechanisms are considered. In the simple setup, r only depends on y. In a more complicated setup, the labelled nodes are generated based on

$$\pi_i \equiv P(r_i = 1 | y_i, h(x_i)) = \frac{\exp\{\alpha_r + \gamma^T h(x_i) + \phi y_i\}}{1 + \exp\{\alpha_r + \gamma^T h(x_i) + \phi y_i\}}, \tag{15}$$

where $h(x_i) = \exp(\sum_j x_{ij}/a_0 - a_1) - (\sum_j x_{ij} - a_2)/a_3$ with value range being $[0, 1]$

Results

		Accuracy	
λ	Method	Mean	SD
1	SM	0.8683	1.98e-2
	Rosset	0.8514	5.19e-2
	GNM	**0.8947**	**6.47e-3**
2	SM	0.8052	3.26e-2
	Rosset	0.8193	6.05e-2
	GNM	**0.8648**	**2.54e-2**

		Accuracy	
λ	Method	Mean	SD
1	SM	0.8663	1.21e-2
	GIM	0.8713	1.52e-2
	GNM	**0.8961**	**1.18e-2**
2	SM	0.8141	2.34e-2
	GIM	0.8291	2.79e-2
	GNM	**0.8669**	**1.63e-2**

Real Analysis

We add more experiments on some other dataset, such as 'Citeseer' and explore the performance of our model using other state-of-art architecture such as GAT. The two tables below compare all model settings on datasets 'Cora' and 'Citeseer', respectively.

			Accuracy	
(N_0/N_1)	λ	Method	Mean	SD
2.31	1	SM + GCN	0.8683	1.98e-2
		GNM + GCN	0.8947	6.47e-3
		SM + GAT	0.8771	1.51e-2
		GNM + GAT	0.8968	8.65e-3
	2	SM + GCN	0.8052	3.26e-2
		GNM + GCN	0.8648	2.54e-2
		SM + GAT	0.8015	3.85e-2
		GNM + GAT	0.8706	1.95e-2

Table 1: Prediction Accuracy for 'Cora'

			Accuracy	
(N_0/N_1)	λ	Method	Mean	SD
3.75	1	SM + GCN	0.8537	2.47e-2
		GNM + GCN	0.8981	7.95e-3
		SM + GAT	0.8076	7.98e-2
		GNM + GAT	0.8785	2.97e-2
	2	SM + GCN	0.5295	1.24e-1
		GNM + GCN	0.8325	7.09e-2
		SM + GAT	0.5898	1.42e-2
		GNM + GAT	0.8090	6.45e-2

Table 2: Prediction Accuracy for 'Citeseer'

Conclusion

Our contributions can be summarized as follows:

- We are the first to consider the graph-based semi-supervised learning problem in the presence of non-ignorable nonresponse
- We introduce a novel identifiability in prediction-equivalence quotient (PEQ) space for neural network architectures
- We propose a novel joint estimation approach by integrating the inverse weighting framework with a modified loss function based on the imputation of non-response
- We use gradient descent (GD) algorithm to learn all the parameters, which works for traditional regression model as well as for modern deep graphical neural networks.

Thank You!