Heterogeneous Graph Structure Learning through the Lens of Data-generating Processes

Keyue Jiang[1], Bohan Tang[2], Xiaowen Dong[2] and Laura Toni[1]

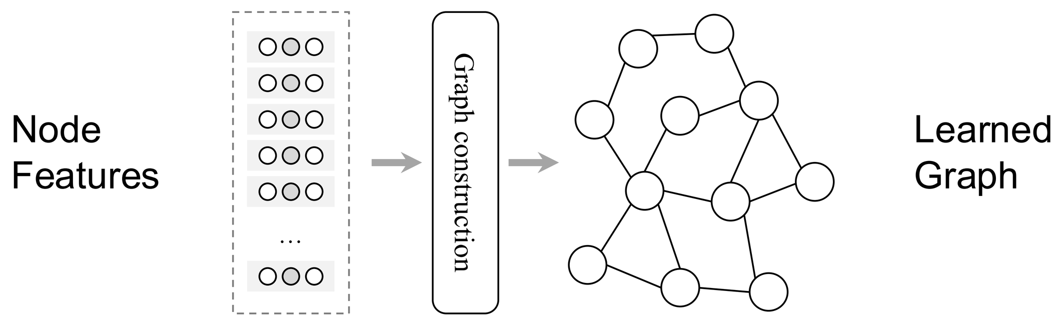
1. University College London, London (UK) 2. University of Oxford, Oxford (UK)

UNIVERSITY OF OXFORD

UCL

Introduction

- Graphs can represent complex data in network systems.
- However, a meaningful graph is not always readily available from the data. Learning the underlying graph from observed data is important in graph machine learning (GML) and Graph Signal Processing (GSP).



Hidden Markov Networks for Graphs

- **A graph** $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ has nodes $v \in \mathcal{V}$ and edges $e \in \mathcal{E}$. Node features $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times K}$. The graph adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$.
- **Hidden Markov Networks:** With the **graphical model** (figure 1a),when there is a latent variable $\boldsymbol{Y}$, we have the PDF:

$$p(\mathbf{W}, \boldsymbol{X}, \boldsymbol{Y}) = p(\boldsymbol{X} \mid \boldsymbol{Y}) p(\boldsymbol{Y} \mid \boldsymbol{W}) p(\boldsymbol{W})$$

$p(\boldsymbol{Y} \mid \boldsymbol{W}) = \prod_v \psi_1(v) \prod_{\{u,v\} \in \mathcal{E}} \psi_2(u, v)$ (node-wise potential $\psi_1(v)$ and edge-wise potential $\psi_2(u, v)$) and $p(\boldsymbol{X} \mid \boldsymbol{Y}) = \mathcal{N}(\boldsymbol{V}\boldsymbol{y}_v, \boldsymbol{\Sigma}_x)$

So we get, $p(\mathbf{W}, \boldsymbol{X}) = p(\boldsymbol{X} \mid \mathbf{W}) p(\mathbf{W}) = \int p(\mathbf{W}, \boldsymbol{X}, \boldsymbol{Y}) \boldsymbol{dY}$

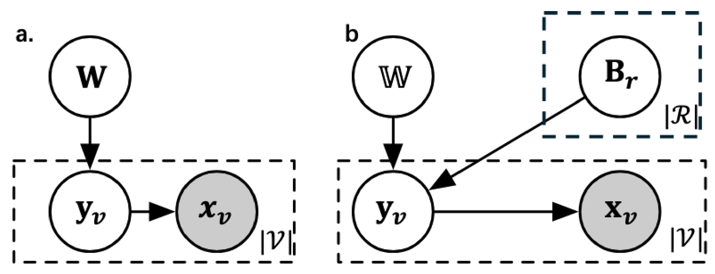


Figure 1: The graphical models for a) HMN and b) our H2MN. The shadowed variable is observable.

Algorithm Design

- To make the optimization problem convex, we use low-rank approximation: $\boldsymbol{e}_r \boldsymbol{e}_r^\top \approx \boldsymbol{U}_{\phi(u)}^\top \boldsymbol{B}_r \boldsymbol{U}_{\phi(v)}$.
- The optimization problem is decomposed into the **Graph Structure Learning step** and the **relation embedding update step**.
- The general optimization objective:

$$\operatorname*{argmin}_{\mathbf{w}, \mathbf{e}} \sum_r w_{uvr} \parallel \boldsymbol{e}_r \circ (\boldsymbol{x}_u - \boldsymbol{x}_v) \parallel^2 + \Omega(\mathbf{W}) + \Omega(\boldsymbol{e}_r)$$

We choose $\log p(\mathbf{W}) = \mathbf{1}^\top \log((\sum_r \mathbf{W}_{::r}) \cdot \mathbf{1})$ and an elastic norm on $\boldsymbol{e}_r$

- **Graph Structure Learning Step**: with reparameterization: $\mathbf{w} \in \mathbb{R}^{|\mathcal{V}| \cdot (|\mathcal{V}|-1) \cdot |\mathcal{R}|/2}$, $\boldsymbol{e}_r \circ (\boldsymbol{x}_u - \boldsymbol{x}_v) \longrightarrow \mathbf{z}$, we get

$$\operatorname*{argmin}_{\mathbf{w}} \parallel \mathbf{w} \circ \mathbf{z} \parallel_F^2 + \mathcal{I}(\mathbf{w} > 0) - \alpha \mathbf{1}^\top \log\left(\left(\sum_r \mathbf{W}_{::r}\right) \cdot \mathbf{1}\right) + \beta \parallel \mathbf{w} \parallel_1$$

which we adapt PDS and ADMM to optimize.

- **Relation Embedding Update Step:**

$$\boldsymbol{e}_{r,k}^{t+1} = \frac{\lambda_2}{\lambda_1}\left(\frac{1}{\lambda_1}\left(\sum_{\{u,v,r\} \in \mathcal{E}_r} w_{uvr}^t \boldsymbol{x}_{v,k} \cdot \boldsymbol{x}_{u,k} - 2\right)\right)$$

$$= \lambda_1' \sum_{\{u,v,r\} \in \mathcal{E}_r} w_{uvr}^t \boldsymbol{x}_{v,k} \cdot \boldsymbol{x}_{u,k} - \lambda_2'$$

which is an analytical solution to the sub optimization problem.

Heterogeneous Graph Structure Learning (HGSL)

- **Heterogeneous Graphs (HGs)**: the graphs with multiple node types and edge types. Each node v has type $\phi(v)$ and edge e has type $r = \phi(e)$. The graph structure is represented by tensor $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{R}|}$.

They widely exist in recommendation systems, social networks, academic networks, etc.

- **Hidden Markov Networks for HGs (H2MN)** (Figure 1b)
  - The node-potential incorporates the impact of type $\phi(v)$: $p(\boldsymbol{X} \mid \boldsymbol{Y}) = \mathcal{N}(\boldsymbol{V}_{\phi(v)} \boldsymbol{y}_v, \boldsymbol{\Sigma}_x)$.
  - For edge types, the edge-wise potential: $\psi_r(u, v) = \exp(w_{uvr} \boldsymbol{y}_u^T \boldsymbol{B}_r \boldsymbol{y}_v)$

By marginalizing out the latent variable $\boldsymbol{Y}$, we obtain

$$p(\boldsymbol{X} \mid \mathbf{W}, \{\boldsymbol{B}_r\}) = \prod_v \psi_1(v) \prod_{\{u,v,r\} \in \mathcal{E}} \psi_2(u, v)$$

With $\psi_1(v) = \exp(-(v + d_v) \left|\boldsymbol{U}_{\phi(v)} \boldsymbol{x}_v\right|^2)$ modulate by node type;

and $\psi_r(u, v) = \exp(w_{uvr} \left|\boldsymbol{x}_u \boldsymbol{U}_{\phi(u)}^\top \boldsymbol{B}_r \boldsymbol{U}_{\phi(v)} \boldsymbol{x}_v\right|^2)$ modulated by relation type.

- **HGSL Problem Formulation:**

Given nodes $v \in \mathcal{V}$ with type $\phi(v)$ and features $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times K}$, together with a relation type set $\mathcal{R}$, learn a weighted and undirected graph represented by tensor $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{R}|}$ that encodes the weights and edge types.

- HGSL formulation through H2MN:

Maximum a-posterior estimation:

$$\mathbf{W}^*, \{\boldsymbol{B}_r^*\} = \operatorname{argmax}_{\mathbf{W}, \{\boldsymbol{B}_r\}} p(\mathbf{W}, \{\boldsymbol{B}_r\} \mid \boldsymbol{X})$$

$$= \operatorname{argmax}_{\mathbf{W}, \{\boldsymbol{B}_r\}} \log p(\boldsymbol{X} \mid \mathbf{W}, \{\boldsymbol{B}_r\}) + \log p(\mathbf{W}) + \log p(\{\boldsymbol{B}_r\})$$

However, the problem is not jointly convex in $\mathbf{W}, \{\boldsymbol{B}_r\}$.

Experiment Results

Results in graph structure learning:

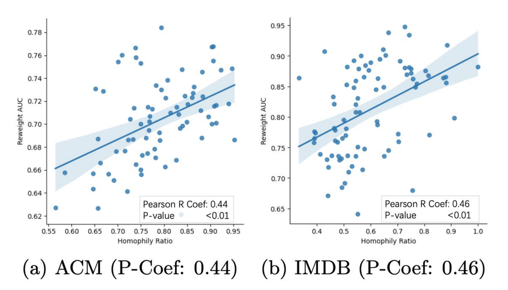
- HGSL compared to Graphical Lasso variants and vanilla GSL algorithms

Model	Synthetic Dataset		IMDB (HR: 0.51)		ACM (HR: 0.64)	
	AUC	GMSE	AUC	GMSE	AUC	GMSE
GGL (Jacob et al., 2009)	0.63 ± 0.03	0.06 ± 0.02	0.60 ± 0.06	0.5 ± 0.02	0.68 ± 0.05	**0.02 ± 0.01**
FGL (Danaher et al., 2014)	0.62 ± 0.03	0.05 ± 0.01	0.73 ± 0.06	**0.03 ± 0.03**	0.59 ± 0.03	0.03 ± 0.00
GSL (Dong et al., 2016)	0.61 ± 0.04	0.04 ± 0.00	0.75 ± 0.06	0.30 ± 0.04	0.56 ± 0.06	0.09 ± 0.08
GSL (Kalofolias, 2016a)	0.58 ± 0.02	0.04 ± 0.02	0.74 ± 0.08	0.27 ± 0.10	0.57 ± 0.04	0.14 ± 0.03
GSL (Pu et al., 2021a)	0.65 ± 0.01	0.03 ± 0.00	0.74 ± 0.04	0.29 ± 0.08	0.65 ± 0.06	0.07 ± 0.10
HGSL-IR (Ours)	**0.83 ± 0.04**	**0.02 ± 0.01**	**0.81 ± 0.07**	0.07 ± 0.05	**0.73 ± 0.02**	0.12 ± 0.07

* Experimental results are evaluated over 30 trials and the mean/standard deviation is calculated.

Robustness Analysis:

- The Impact of homophily ratio and signal smoothness to HGSL.



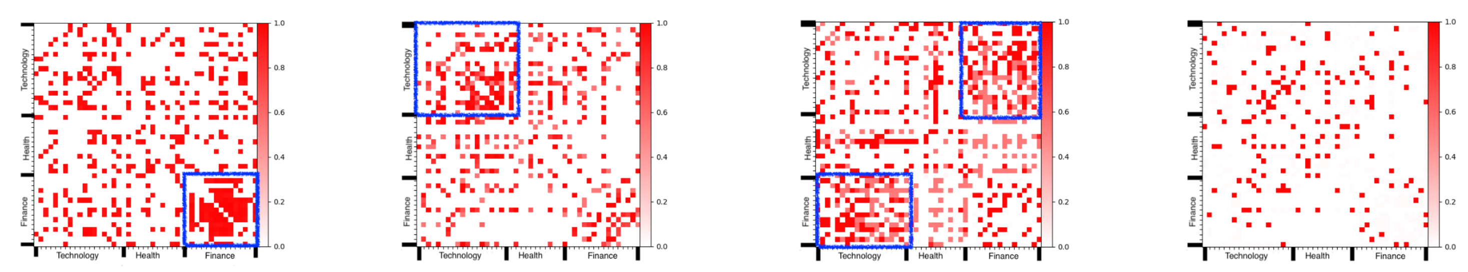
(a) ACM (P-Coef: 0.44) (b) IMDB (P-Coef: 0.46)

Figure 2: The Pearson correlation test between relaxed homophily ratio and AUC.

Figure 3: The SDOR and model performance.

Real-world Analysis

- HGSL applied to financial relation mining among companies.



(a) Connections with type finance-to-finance. (b) Connections with type technology-to-technology. (c) Connections with type technology-to-finance. (d) Connections with type health-to-technology.

Contact: keyue.jiang.18@ucl.ac.uk

UKRI Engineering and Physical Sciences Research Council