LEARNING HETEROGENEOUS GRAPHS WITH GENERALIZED SMOOTHNESS

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

[1]University College London, [2]University of Oxford

ABSTRACT

An important task in graph signal processing is learning the graph structure (when it is not readily available) from observed data such that it captures well the intrinsic relationships between the data entities. This has led to the development of multiple graph learning technologies mainly focusing on homogeneous graphs. Many real-world graphs, however, exhibit heterogeneous patterns, where nodes and edges have multiple types. This poses a challenge in graph learning, as node signals have more personalized information that does not contribute equally to the formation of edges. This paper introduces the first approach for heterogeneous graph structure learning (HGSL). To this end, we define a novel concept of generalized smoothness by considering the heterogeneity of nodes and edges and emphasizing the signal dimensions that contribute most to the smoothness. We then propose an optimization-based HGSL method that imposes such smoothness prior. Finally, we conduct extensive experiments on both synthetic and real-world datasets, which demonstrate that our proposed method consistently outperforms the baselines in terms of edge identification and edge weight recovery.

***Index Terms*—** Graph Signal Processing, Graph Machine Learning, Heterogeneous Graphs, Graph Structure Learning

1. INTRODUCTION

Graphs are a powerful and ubiquitous representation of complex relational data, which is capable of capturing pairwise relationships between entities. However, a meaningful graph is not always readily available from the data [1], and sometimes the graph observed is not always clean [2]. Learning the underlying graph structure from observed data is therefore an important problem in the field of both graph signal processing [3, 4, 5] and graph machine learning [6, 7].

Model-based graph structure learning (GSL) approaches [1, 8, 9] often assume that the signals on the graphs should admit certain regularity or smoothness. However, current methods focus only on homogeneous graphs, where nodes and edges belong to a single type. Real-world graphs often exhibit heterogeneous patterns [10], with multiple types of nodes and edges representing different kinds of entities and relationships. For instance, in a network representing a recommender system [11], nodes can have distinct types (e.g. users and items), and edges can represent different types of relations (e.g. like/dislike for user-item edges or following/being followed for user-user edges). Other examples include social networks, academic networks [12, 13], and knowledge graphs [14, 15, 16].

Existing GSL methods are not well-suited for heterogeneous graphs, as the smoothness assumption does not generalize well when diverse node and edge types are present due to the following *limitations*: 1) All nodes and edges are treated equally without considering the different node and/or edge types that may affect the way the edges are formed; 2) In classical GSL frameworks all signal dimensions are considered contributing to the smoothness prior of the signal while in practice this may not be the case.

In this study, we address these limitations by introducing a novel heterogeneous graph structure learning (HGSL) framework. We first define a generalized smoothness prior on heterogeneous graphs by specifying a relation-wise embedding for each edge type, which helps the model to distinguish various edge types. The embeddings also serve as a re-weighting mechanism to emphasize the signal dimensions that contribute most to the HGSL task. Based on this approach, we propose an objective function for learning the adjacency tensor that captures the structure of the heterogeneous graph with different node/edge types. Our method generalizes GSL approaches to heterogeneous graphs and is applicable to a wide range of graphs, including bipartite, multi-relational, and knowledge graphs.

2. PRELIMINARIES

Model-based Graph Structure Learning [1, 8, 17]: Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ denote a graph on edges $\mathcal{E}$ and nodes $\mathcal{V}$ that are associated with the node signals $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times K}$. Current GSL algorithms aim to learn the structure of homogeneous graphs, represented by the weight matrix $\mathbf{W} = \{w_{ij}\}_{i,j\in\{1:N\}}$, behind observed signals by solving an optimization problem. The optimization objective often consists of a graph-signal fidelity term $S(\mathbf{X}, \mathbf{W})$ and a structural regularizer $\Omega(\cdot)$. The graph-signal fidelity term reflects prior knowledge of the data distribution, and a widely used assumption is the smoothness of the signals [2] - the signal should vary slowly along edges of the underlying graph. The smoothness is measured by the variation of signals on a graph [18], $S(\mathbf{X}, \mathbf{W}) = \frac{1}{2}\sum_{\{i,j\}\in\mathcal{E}} w_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. Thus, finding the optimal weight matrix that minimizes smoothness S facilitates the learning of graph topology. With a proper regularizer $\Omega(\cdot)$, the optimization problem is formulated as follows [1]: $\mathbf{W}^* = \arg\min_{\mathbf{W}} S(\mathbf{X}, \mathbf{W}) + \Omega(\mathbf{W})$.

An (Undirected) Heterogeneous Graph [19] $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is associated with a node type set $\mathcal{A}$ and an edge type set $\mathcal{R}$. Each node $v \in \mathcal{V}$ has a node type $\phi(v) \in \mathcal{A}$ and each edge $e \in \mathcal{E}$ is assigned with a relation type $r \in \mathcal{R}$. Since heterogeneous graphs are multi-relational, an edge in the graph is an undirected triplet $e = \{v, u, r\}$, which means node v and u are connected by relation type r. Thus, we represent the connections on the graph as a weighted 3-D tensor $\mathbb{W} = \{w_{vur}\} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|\times|\mathcal{R}|}$, with $|\cdot|$ the set cardinality. For each node pair, $\mathbb{W}_{vu:}$ is a $|\mathcal{R}|$-long vector with only one nonzero entry indicating the edge weight and type. Most of the heterogeneous graph datasets [12, 11, 20, 21, 22] condition edge types on the node types, i.e., if node u is type 'user' and v is 'item', the relation type r can only be 'like' or 'dislike'. We follow this convention to reduce the degree of freedom in determining relation types. Last, the node signals on a heterogeneous graph can be represented by a function $f : \mathcal{V} \to \mathbb{R}^K$, which assigns a vector $\mathbf{x}_v \in \mathbb{R}^K$ to node $v \in \mathcal{V}$. Some datasets [12, 23, 20] have a type-specific dimension for each node type. The study will focus on the cases with unified dimension sizes but the methods can be generalized to any heterogeneous graph by projecting the signals into a universal $\mathbb{R}^K$ with extra linear modules. Note that we do not consider edge signals encoding edge attributes and leave it for future work.

Heterogeneous Graph Structure Learning: Given nodes $\{v\}_{v\in\mathcal{V}}$ with corresponding type $\{\phi(v)\}_{v\in\mathcal{V}}$ and associated signals $\{\mathbf{x}_v\}_{v\in\mathcal{V}}$, together with a potential relation type set $\mathcal{R}$, we aim to learn a weighted and undirected heterogeneous graph $\mathcal{G}$ represented by tensor $\mathbb{W}$ that encodes the identification and types of edges.

3. MOTIVATING EXAMPLE

In this section, we give a motivating example to better understand how we could tackle the two limitations mentioned earlier. We start with an intuition behind academic networks, where nodes have types 'paper' and 'author' and signals are the attributes like 'topics', 'affiliation', and 'publication-date', etc. In such a context, papers on specific topics are inclined to cite papers in the same fields, indicating 'topics' as a pivotal attribute to identify the edges with the type 'Paper-Cite-Paper' ('PP'). Meanwhile, attributes like 'Affiliations' cannot help us determine the existence of 'PP'-typed edges, though they exist in the signal space and are beneficial for identifying other types of edges, e.g., 'Author-Write-Paper' ('PA'). Hence, when deciding whether to form an edge of type 'PP', the model should focus on the dimension of the node signals that represent highly relevant attributes like 'topics', but ignore the irrelevant ones like 'affiliation' or 'date'. To do so, a key concept, *dimension-wise smoothness* for the relation type r and the dimension k, is defined as $\mathcal{S}(\mathbf{X}_{:k}, \mathbb{W}_{::r}) = \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{x}_{v,k} - \mathbf{x}_{u,k}\|_2^2$, where $\mathbb{W}_{::r}$ is the r-th slice of the adjacency tensor and $\mathbf{x}_{v,k}$ is the k-th element of the signal at node v.

We then consider two questions when designing HGSL within the framework of minimizing smoothness: Q1) How does dimension-wise smoothness help form edges? Q2) Can we distinguish various relation types based on dimension-wise smoothness? We will answer these questions by studying two heterogeneous graph datasets. **IMDB** [11]: a movie review dataset with node types including directors (D), actors (A), and movies (M) and with signals as 3066-D bag-of-words representation of keywords in the movie plot. **ACM** [12]: an academic dataset contains papers (P), authors (A), and subjects (S). Signals correspond to the 1902-D bag-of-words representation of the keywords in diverse research areas.

To understand Q1, by exhaustive search, we identified the 200 most and least smooth dimensions $\mathcal{K}^{\mathrm{M}}(r)$ and $\mathcal{K}^{\mathrm{L}}(r)$, respectively, by ranking $\mathcal{S}(\mathbf{X}_{:k}, \mathbb{W}_{::r})$. For each of these two sets, we evaluate the pair-wise similarity (which is consistent with smoothness but normalized to [0,1]) of the signals, i.e., $\sum_{k\in\mathcal{K}^{\mathrm{L}}(r)} \mathbf{x}_{v,k}\cdot\mathbf{x}_{u,k}$, and $\sum_{k\in\mathcal{K}^{\mathrm{M}}(r)} \mathbf{x}_{v,k}\cdot\mathbf{x}_{u,k}$. For similarity higher than a given threshold, we establish an edge w_{vur} and compared the constructed graph with the ground truth one in an edge identification task. We compared the performance between the two generated graphs by the area under the curve (AUC) in table 1. We found that dimensions with higher smoothness are more helpful for edge identification task in HGSL.

To answer Q2, we assume that edges with distinct relation types are associated with smoothness evaluated in different signal dimensions, and the smoothest dimensions are the most representative ones. This motivates us to define *Smoothest-Dimension Overlapping Ratio (SDOR)*: for each pair of relations (r, r'), the SDOR is calculated by counting the overlapping dimensions in $\mathcal{K}^{\mathrm{M}}(r)$ and $\mathcal{K}^{\mathrm{M}}(r')$: $\mathrm{SDOR}(r, r') = \frac{|\mathcal{K}^{\mathrm{M}}(r)\cap\mathcal{K}^{\mathrm{M}}(r')|}{|\mathcal{K}^{\mathrm{M}}(r)\cup\mathcal{K}^{\mathrm{M}}(r')|}$. The SDOR reflects how different two relation types are in terms of exhibiting smoothness in signal dimensions and we report the pair-wise SDOR in table 1. The result suggests that for pairs of relation types with lower SDOR the signals will exhibit smoothness in different dimensions, which reveals the possibility of distinguishing them by comparing the smoothest dimensions found.

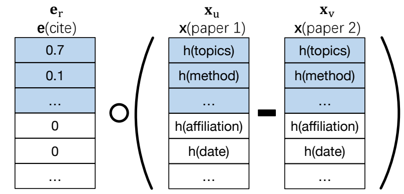


Fig. 1: Visualization of the generalized smoothness.

Table 1: Edge identification with different dimensions. AUC-M are the results with the smoothest dimensions and AUC-L the least ones.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\mathcal{R}$	AUC-M	AUC-L	SDOR
IMDB	11k	550k	MA	0.894	0.547	MA-MD: 0.65
			MD	0.900	0.546	
ACM	21k	87k	PP	0.771	0.493	PP-PA: 0.66
			PA	0.925	0.648	PP-PS: 0.67
			PS	0.804	0.846	PA-PS: 0.94

4. PROPOSED METHODOLOGY

In the following, we first propose a generalized smoothness that captures the signal behavior in heterogeneous graph structures. Based on this, we then derive the framework of HGSL and design novel algorithms to solve it.

4.1. Generalized Smoothness for Heterogeneous Graph Signals

According to section 3, we realize that solving HGSL requires: 1) focusing on specific dimensions of signals to measure smoothness and 2) weighing these dimensions depending on the relation type. To this end, we propose a *generalized smoothness* on the heterogeneous graph by specifying a *learnable* relation embedding $\mathbf{e}_r \in \mathbb{R}^K, \forall r\in\mathcal{R}$ that has the same dimension size as node signals. The idea is to introduce a *reweighted smoothness* scheme by first measuring dimension-wise smoothness, and then integrating it by emphasizing specific signal dimensions according to r. This process is illustrated in fig. 1. Using academic networks again as an example, while determining whether a 'Cite'-typed edge should be formed between two paper nodes, the model should focus on the signal dimensions that represent the 'topics', but weigh less the irrelevant ones like 'affiliation' or 'date'. In other words, larger weights should be assigned to the 'topics' dimension and smaller ones to others. According to this, our new generalized smoothness is,

$$\begin{aligned} S &= \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r\circ(\mathbf{x}_v - \mathbf{x}_u)\|_2^2 \\ &= \langle \mathbb{W}, \|(\mathbf{X}\otimes\mathbf{1} - \mathbf{1}\otimes\mathbf{X})\otimes\mathbf{E}\|_F^2\rangle \end{aligned} \tag{1}$$

where $\circ$ is the element-wise product and $\|\cdot\|_F$ is the Frobenius norm along the last dimension which collapses the tensor from dimension 4 to 3. $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_{|\mathcal{V}|}]^{\mathbf{T}} \in \mathbb{R}^{|\mathcal{V}|\times K}$, $\mathbf{E} = [\mathbf{e}_1, ..., \mathbf{e}_{|\mathcal{R}|}]^{\mathbf{T}} \in \mathbb{R}^{|\mathcal{R}|\times K}$, and $\mathbf{1}\in\mathbb{R}^{|\mathcal{V}|\times K}$ is an all-one matrix. $\langle\cdot,\cdot\rangle$ and $\otimes$ are the

Algorithm 1: HGSL with Alternative Weight Update

Input: Node signals $\{\mathbf{x}_v\}_{v=1}^{|\mathcal{V}|}$ and types $\{\phi(v)\}_{v=1}^{|\mathcal{V}|}$; Relation type set $\mathcal{R}$; Maximum steps T.
Output: The graph weight $\mathbf{w}$; Embeddings $\mathbf{e}_r, \forall r \in \mathcal{R}$;

```
Init: Initialize e_r^0 = 1^T/K, ∀r ∈ R; Initialize w^0 randomly ;
while t < T do
    /* Graph structure learning step */
    Calculate the smoothness vector z based on eq. (1);
    Optimize w^{t+1} based on eq. (5);
    /* Relation embedding update Step */
    if Update Method == Gradient Descent then
        e_r^{t+1} = GD(e_r^t);
    else if Update Method == Iterative Reweighting then
        Update e_r^{t+1} based on eq. (6);
    end if;
    t = t+1;
end
```

tensor inner product and outer product, respectively. Different from the original smoothness, if node v and u are connected w.r.t. r, the contribution of each signal dimension to the overall smoothness is reweighted by $\mathbf{e}_r$. Intuitively, a heterogeneous graph is thought to be smooth if strongly connected nodes (with larger w_{vur}) have similar signal values in the dimensions emphasized by $\mathbf{e}_r$ for the relation r.

4.2. Heterogeneous Graph Structure Learning Framework

Since the problem requires learning both the *graph structures* $\mathbb{W}$ and *relation embeddings* $\mathbf{E}$ (for reweighted smoothness), we adopt an alternating optimization scheme to solve it: $\mathbb{W}$ is optimized with $\mathbf{E}$ fixed, and $\mathbf{E}$ is optimized with $\mathbb{W}$ fixed. For notation consistency, we will use $\mathcal{E}'$ to denote the possible connections in a heterogeneous graph. Note that this is not the combination of all possible nodes and relation types that shapes a $|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{R}|$ space as node pairs with certain types can only be connected by specific edge types.

Graph structure learning step. We formalize the optimization of $\mathbb{W}$ as a maximum a posterior estimation problem with a fixed $\mathbf{E}$:

$$\begin{aligned}\mathbb{W}^* &= \arg\max_{\mathbb{W}} p(\mathbb{W} \mid \mathbf{X}, \mathbf{E}) = \arg\max_{\mathbb{W}} p(\mathbf{X}, \mathbf{E} \mid \mathbb{W}) \cdot p(\mathbb{W}) \\ &= \arg\min_{\mathbb{W}} (-\log p(\mathbf{X}, \mathbf{E} \mid \mathbb{W}) - \log p(\mathbb{W}))\end{aligned} \tag{2}$$

The objective function contains two parts, where $p(\mathbf{X}, \mathbf{E} \mid \mathbb{W})$ is the data fidelity (likelihood) term and $p(\mathbb{W})$ is the regularizer (prior) imposed on the heterogeneous graph structure. Extended from [8, 24], the likelihood is modeled based on the generalized smoothness:

$$p(\mathbf{X}, \mathbf{E} \mid \mathbb{W}) \propto \exp\left(-\frac{1}{\sigma} \sum_{\{v,u,r\} \in \mathcal{E}'} w_{vur} \|\mathbf{e}_r \circ (\mathbf{x}_v - \mathbf{x}_u)\|_2^2\right) \tag{3}$$

Following the GSL formulation, we use $\Omega(\mathbb{W})$ to replace $p(\mathbb{W})$. Substituting eq. (3) into eq. (2), the optimization problem of HGSL can be formulated as,

$$\begin{aligned}\min_{\mathbb{W}} \quad & \langle \mathbb{W}, \|(\mathbf{X} \otimes \mathbf{1} - \mathbf{1} \otimes \mathbf{X}) \otimes \mathbf{E}\|_F^2 \rangle + \Omega(\mathbb{W}) \\ \text{s.t. } & [\mathbb{W}]_{vur} = [\mathbb{W}]_{uvr} \geq 0, \text{ for } v \neq u, \forall r\end{aligned} \tag{4}$$

The first term is the generalized smoothness as in eq. (1). The constraints guarantee that the learned $\mathbb{W}$ is valid. $\Omega(\mathbb{W})$ is a regularizer that imposes certain structures on $\mathbb{W}$.

Since we focus on undirected graphs (i.e., $\mathbb{W}$ symmetric), we only need to focus on the upper triangle part of the tensor, i.e., the vectorized weights $\mathbf{w} \in \mathbb{R}_+^{|\mathcal{V}|\cdot(|\mathcal{V}|-1)\cdot|\mathcal{R}|/2}$. We reparameterize the training task as in [8] and obtain $\sum_{\{v,u,r\}\in\mathcal{E}'} w_{vur}\|\mathbf{e}_r \circ (\mathbf{x}_v - \mathbf{x}_u)\|_2^2 = \|\mathbf{w} \odot \mathbf{z}\|$, where $\mathbf{z} \in \mathbb{R}_+^{|\mathcal{V}|\cdot(|\mathcal{V}|-1)\cdot|\mathcal{R}|/2}$ is the half-vectorization of the tensor $\|(\mathbf{X} \otimes \mathbf{1} - \mathbf{1} \otimes \mathbf{X}) \otimes \mathbf{E}\|_F^2$. In addition, inspired by classical GSL methods [9, 1], we choose $\Omega(\mathbb{W})$ to consist of a log barrier regularizer on node degrees and a ℓ_1 norm regularizer on the weights to promote the connectivity and the sparsity of the graph respectively. With this, the objective is reformulated as,

$$\min_{\mathbf{w}} \|\mathbf{w} \odot \mathbf{z}\| - \alpha \mathbf{1}^\top \log(\mathcal{T}\mathbf{w}) + \beta\|\mathbf{w}\|_1 + \mathcal{I}_{\mathbf{w}>0} \tag{5}$$

and $\mathcal{T}$ is a linear operator that transforms $\mathbf{w}$ into the vector of node degrees such that $\mathcal{T}\mathbf{w} = (\sum_{r=1}^{|\mathcal{R}|} \mathbb{W}_{::r}) \cdot \mathbf{1}$. We follow the literature and solve eq. (5) by alternating direction method of multipliers (ADMM) [8] or primal-dual splitting algorithms (PDS) [9].

Relation embedding update step. We develop two methods to update relation embeddings $\mathbf{e}_r$. The first is to introduce $\mathbf{e}_r$ as an extra parameter in eq. (5) and optimize it by gradient descent (HGSL-GD). However, in our experiments, we found this unstable and easily stuck into a sub-optimum. Thus, a more efficient and stable solution is developed. We call this algorithm iterative reweighting (HGSL-IR). The key intuition is to assign higher weights for relation r to dimensions that express higher dimension-wise similarity on the graph learned at iteration t (i.e., w_{vur}^t), which is formulated as:

$$\mathbf{e}_{r,k}^{t+1} = \frac{\sum_{\{v,u,r\}\in\mathcal{E}'_r} w_{vur}^t \, \mathbf{x}_{v,k} \cdot \mathbf{x}_{u,k}}{\sqrt{(\sum_{\{v,u,r\}\in\mathcal{E}'_r} \mathbf{x}_{v,k}^2)(\sum_{\{v,u,r\}\in\mathcal{E}'_r} \mathbf{x}_{u,k}^2)}} \tag{6}$$

where $\mathcal{E}'_r$ is the set of possible edges of type r. In each iteration, we normalize relation embeddings to be unit vectors in terms of the ℓ_2 norm. When reweighted by w_{vur}^t, the node pairs that are more likely to be connected would contribute more to the similarity measurement. This gives a more accurate estimation of the relation embeddings. The whole process is in algorithm 1.

Factors impacting model performance: We consider two important factors that may influence the model performance. *[Homophily ratio]* (HR) [20] measures how similar the connected nodes are in a heterogeneous graph, which is related to the smoothness of the signal on a graph. It is defined based on the meta-paths set induced by the graph [10]. A meta-path Φ is a path following a specific sequence of node and relation types like $\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_L} \mathcal{A}_L$. When $\mathcal{A}_1 = \mathcal{A}_L$, the graph induced by Φ is a homogeneous graph $\mathcal{G}_\Phi$ with edges being the meta-path defined by Φ. This helps us to define the HR: HR $(\mathcal{G}_\Phi) = \frac{\sum_{\{i,j\}\in\mathcal{E}_\Phi} \mathbb{I}(y_i = y_j)}{|\mathcal{E}_\Phi|}$, where y_i and y_j are the node labels for downstream classification tasks. *[SDOR]* as defined in section 3 would be another factor impacting the model performance. If two relation types r and r' have a high SDOR, the relation embeddings learned in eq. (6) would be similar, thus solving eq. (4) lead to a result with equal $\mathbb{W}_{::r}$ and $\mathbb{W}_{::r'}$. This makes two relation types indistinguishable in the learned graph. Thus, the model's robustness against high SDOR should be tested.

5. EXPERIMENTS

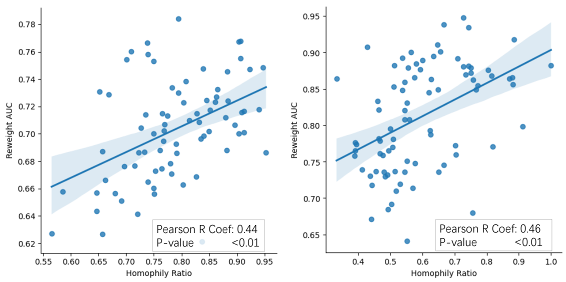
5.1. Dataset Setups

Synthetic Datasets: The synthetic dataset construction consists of 3 phases: 1) *Graph backbone generation*: We followed the process in [8] and used stochastic block model and Watts–Strogatz model to generate the graph backbone with 20-100 nodes that encodes the

Table 2: Experiment results on Heterogeneous Graph Structure Learning

Dataset	HR	K	Vanilla GSL		HGSL-GD		HGSL-IR	
			AUC	NMSE	AUC	NMSE	AUC	NMSE
Synthetic	0.95+	60	0.66 ± 0.05	0.03 ± 0.00	0.70 ± 0.04	0.27 ± 0.05	$\mathbf{0.75 \pm 0.02}$	$\mathbf{0.02 \pm 0.00}$
		300	0.65 ± 0.01	0.03 ± 0.00	0.75 ± 0.05	0.30 ± 0.04	$\mathbf{0.83 \pm 0.04}$	$\mathbf{0.02 \pm 0.01}$
		600	0.78 ± 0.03	$\mathbf{0.04 \pm 0.02}$	0.77 ± 0.04	0.26 ± 0.04	$\mathbf{0.89 \pm 0.04}$	0.06 ± 0.01
IMDB	0.51	3066	0.74 ± 0.04	0.29 ± 0.08	0.75 ± 0.06	0.21 ± 0.05	$\mathbf{0.81 \pm 0.07}$	$\mathbf{0.07 \pm 0.05}$
ACM	0.64	1902	0.65 ± 0.06	$\mathbf{0.07 \pm 0.10}$	0.62 ± 0.06	0.09 ± 0.15	$\mathbf{0.73 \pm 0.02}$	0.12 ± 0.07

* Experiment results are evaluated over 30 trials and the mean/standard deviation is calculated. 0.00 means the value < 0.01.



(a) ACM (Pearson Coef: 0.44) (b) IMDB (Pearson Coef: 0.46)

Fig. 2: The correlation test between homophily ratio and AUC

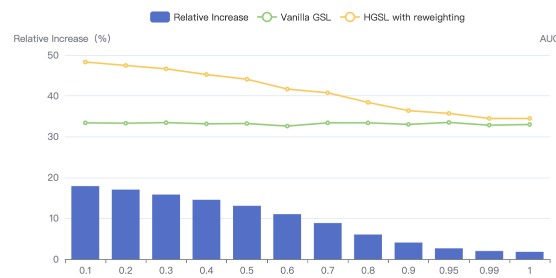


Fig. 3: The SDOR and model performance

graph structure. 2) *Graph labeling*: We label the graph based on the network schema that encodes the node and edge type information [25]. Network schema is a meta template of a heterogeneous graph that represents how nodes/edges with different types are connected. We traverse the graph by breadth-first search to follow the rule defined by the network schema and the nodes are labeled by the preset label distribution. 3) *Signal generation*: We then generate signals for nodes that satisfy the probability distribution in eq. (3).

Real-world Datasets: Experiments are conducted in datasets including IMDB [20], and ACM [12]. The statistics are in table 1. In order to augment more graph instances for a comprehensive evaluation, we subsample the dataset to generate smaller graphs with number of nodes ranging from 50-200 following the strategy in [26].

5.2. Experiment results

There are two research questions to ask: RQ1) How do our algorithms perform compared to the vanilla GSL algorithms? RQ2) How do the factors remarked earlier impact the HGSL performance?

Model Comparison: To answer RQ1, we benchmark existing algorithms with respect to structure recovery ability and stability. The edge identification error (measured by AUC) and edge weight recovery error (measured by the normalized mean squared error, NMSE) are reported in table 2. With our HGSL algorithm with iterative reweighting (IR), a consistent improvement is found in all the datasets in terms of both AUC and NMSE. This suggests the efficacy of our algorithm in both tasks. When testing the HGSL with gradient descent (GD), though a better AUC is found in most datasets, the improvement is rather marginal and the NMSE is much larger. The improvement in synthetic datasets (avg +18.48%) is relatively larger than the real-world datasets (avg +10.88%), which suggests the more challenging nature of the real-world experiments.

Robustness Analysis: We first look into how **the Homophily Ratio** (HR) of a graph could impact the model robustness. The overall HR of 3 datasets are reported in table 2. From the result, we are interested in how the algorithm performance is related to the homophily ratio. To verify our hypothesis that a lower HR impairs the performance, we record the homophily ratio of all the subgraphs when sampling and regress their corresponding AUC in both IMDB and ACM datasets, as shown in fig. 2. The Pearson Correlation Coefficients are reported. According to the results, we can conclude that the homophily ratio is positively correlated with AUC, and one future work could be understanding if this phenomenon is general among all the GSL algorithms. To understand the behavior of our algorithm against high **SDOR**, we fix the number of relation types $|\mathcal{R}| = 2$ in our synthetic experiment, and manually adjust the SDOR from 0 to 1. We illustrate the AUC of the vanilla GSL and HGSL algorithms in fig. 3, and calculate its relative increase. It is clear that the performance of HGSL is significantly affected when the SDOR increases, which suggests that a higher SDOR would impair the distinguishability of relation types. However, the HGSL algorithm is robust until the SDOR is increased to approximately 0.7, which is almost the same level in real-world datasets as shown in table 1.

6. DISCUSSION

In this study, we propose an HGSL framework based on minimizing generalized smoothness. The method is grounded in the assumption that "the smoothest dimensions are crucial", which has proven beneficial in our tests. However, certain real-world heterogeneous graphs, especially those with low homophily ratios, may not conform to this paradigm. In such instances, the smoothness assumption should be revisited and the model should also be modified to capture such signal behaviors. Additionally, the use cases of the current best method, HGSL-IR, are limited to a few assumptions made, e.g. the fixed signal dimension size and positive weights on different signal dimensions. However, HGSL with gradient descent is not constrained by these. Thus, another future work would be developing more reliable GD-based algorithms for general HGSL tasks.

7. REFERENCES

[1] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst, "Laplacian matrix learning for smooth graph signal representation," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 3736–3740.

[2] Siheng Chen, Aliaksei Sandryhaila, José M. F. Moura, and Jelena Kovacevic, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, 2015.

[3] Xiaowen Dong, Dorina Thanou, Michael G. Rabbat, and Pascal Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

[4] Xiaowen Dong, Dorina Thanou, Laura Toni, Michael M. Bronstein, and Pascal Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 117–127, 2020.

[5] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[6] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy, "Machine learning on graphs: A model and comprehensive taxonomy," *J. Mach. Learn. Res.*, vol. 23, pp. 89:1–89:64, 2022.

[7] William L. Hamilton, Rex Ying, and Jure Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017.

[8] Xingyue Pu, Tianyue Cao, Xiaoyun Zhang, Xiaowen Dong, and Siheng Chen, "Learning to learn graph topologies," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, Eds., 2021.

[9] Vassilis Kalofolias, "How to learn a graph from smooth signals," in *AISTATS*. 2016, vol. 51 of *JMLR Workshop and Conference Proceedings*, pp. 920–929, JMLR.org.

[10] Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and Philip S. Yu, "A survey on heterogeneous graph embedding: Methods, techniques, applications and sources," *CoRR*, vol. abs/2011.14867, 2020.

[11] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King, "MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding," in *WWW*. 2020, pp. 2331–2341, ACM / IW3C2.

[12] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang, "Are we really making much progress?: Revisiting, benchmarking and refining heterogeneous graph neural networks," in *KDD*. 2021, pp. 1150–1160, ACM.

[13] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han, "Graph-based consensus maximization among multiple supervised and unsupervised models," in *NIPS*. 2009, pp. 585–593, Curran Associates, Inc.

[14] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[15] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel, "Convolutional 2d knowledge graph embeddings," in *AAAI*. 2018, pp. 1811–1818, AAAI Press.

[16] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek, "YAGO3: A knowledge base from multilingual wikipedias," in *CIDR*. 2015, www.cidrdb.org.

[17] Sandeep Kumar, Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel P. Palomar, "Structured graph learning via laplacian spectral constraints," in *NeurIPS*, 2019, pp. 11647–11658.

[18] Dengyong Zhou and Bernhard Schölkopf, "A regularization framework for learning from graph data," in *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields (SRL 2004)*, 2004, pp. 132–137.

[19] Wang Xiao, Ji Houye, Shi Chuan, Wang Bai, Cui Peng, Yu P., and Ye Yanfang, "Heterogeneous graph attention network," *WWW*, 2019.

[20] Jiayan Guo, Lun Du, Wendong Bi, Qiang Fu, Xiaojun Ma, Xu Chen, Shi Han, Dongmei Zhang, and Yan Zhang, "Homophily-oriented heterogeneous graph rewiring," in *WWW*. 2023, pp. 511–522, ACM.

[21] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *NeurIPS*, 2020.

[22] Muhan Zhang and Yixin Chen, "Inductive matrix completion based on graph neural networks," in *ICLR*. 2020, OpenReview.net.

[23] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter, "Social structure of facebook networks," *CoRR*, vol. abs/1102.2166, 2011.

[24] Brenden M Lake and Joshua B Tenenbaum, "Discovering structure by learning sparse graphs," in *Proceedings of the 32nd Annual Conference of the Cognitive*, 2010.

[25] Chuan Shi, *Heterogeneous Graph Neural Networks*, pp. 351–369, Springer Nature Singapore, Singapore, 2022.

[26] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun, "Heterogeneous graph transformer," in *WWW*. 2020, pp. 2704–2710, ACM / IW3C2.