Subgraph Augmentation with Application to Graph Mining

Jiajun Zhou, Jie Shen, Yalu Shan, Qi Xuan, and Guanrong Chen

September 3, 2021

Notation

- $G = (V, E)$: an undirected and unweighted graph
- $V = \{v_i | i = 1, \ldots, n\}$: a node set
- $E = \{e_i | i = 1, \ldots, m\}$: an edge set
- A: the adjacency matrix
- $D = \{(G_i, y_i) | i = 1, \ldots, t\}$: the dataset

Introduction

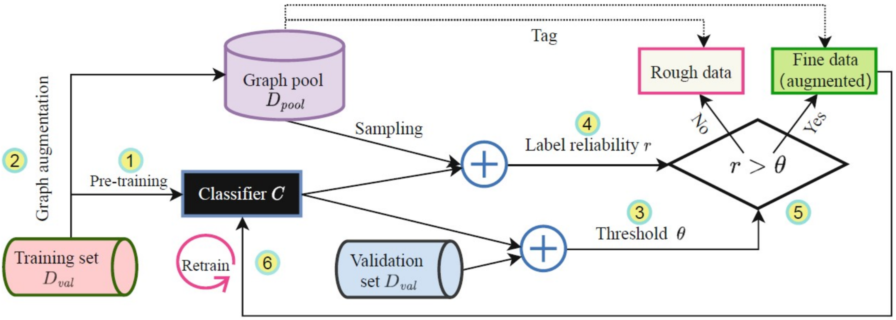

Figure 1: The pipeline.

- Introduction
- Subgraph Augmentation
- Data Filtration
- Model Evolution Framework
- Experiment

Motif-Similarity Mapping

Graph motifs are subgraphs that repeat themselves in a specific graph or even among various graphs.
Here, for simplicity, only consider open-triad motifs $\wedge_{ij}$ with chain structures.

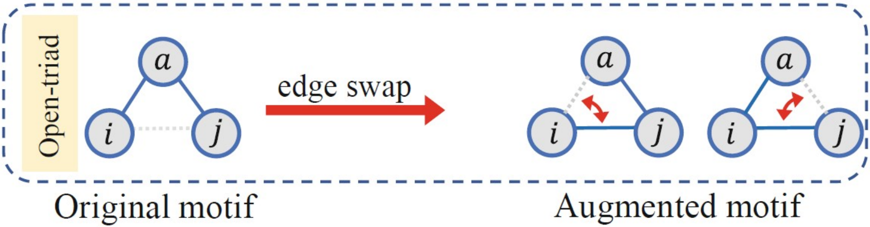

Figure 2: Open-triad motif and heuristic edge swapping.

Motif-Similarity Mapping

The candidate set of pairwise nodes is denotes as:

$$E_{add}^c = \{(v_i, v_j) | A_{ij} = 0, A_{ij}^2 \neq 0; i \neq j\}$$

Then, we get E_{add}, the set of edges added to G, via weighted random sampling from E_{add}^c. For each $\wedge_{ij}$ involving pairwise nodes (v_i, v_j) in E_{add}, we remove one edge from it via weighted random sampling, and all of these removed edges constitute E_{del}.

Resource Allocation(RA) index

The RA score s_{ij} and addition weight w_{ij}^{add} can be computed as follows:

$$
\begin{aligned}
s_{ij} &= \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{d_z} \\
S &= \{s_{ij} | \forall (v_i, v_j) \in E_{add}^c\} \\
w_{ij}^{add} &= \frac{s_{ij}}{\sum_{s \in S} s} \\
W_{add} &= \{w_{ij}^{add} | \forall (v_i, v_j) \in E_{add}^c\}
\end{aligned}
$$

where $\Gamma(i)$ denotes the neighbors of v_i and d_z denotes the degree of node z.

Resource Allocation(RA) index

Similarly, for edge deletion,

$$w_{ij}^{del} = 1 - \frac{s_{ij}}{\sum_{s \in S} s}$$
$$W_{del} = \{w_{ij}^{del} | \forall (v_i, v_j) \in \wedge_{ij}\}$$

Motif-Similarity Mapping

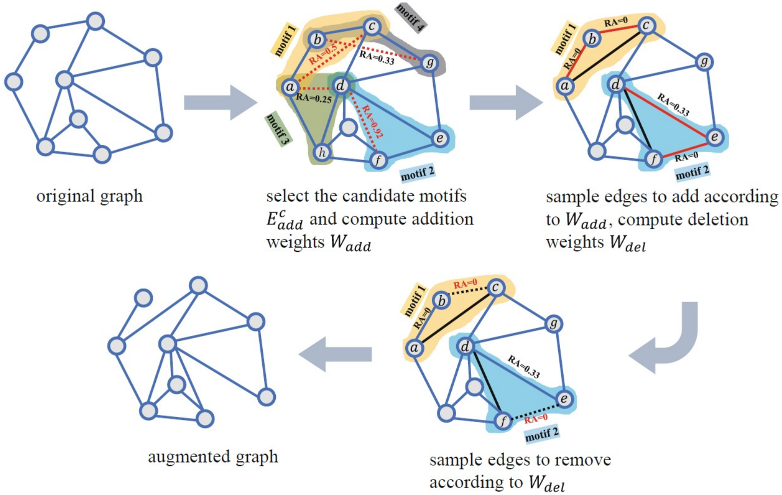

Figure 3: An example of subgraph augmentation via motif-similarity mapping; red lines is the candidates and black lines is the modified edges.

- Introduction
- Subgraph Augmentation
- Data Filtration
- Model Evolution Framework
- Experiment

Data Filtration

Each graph G_i in D_{val} will be fed into classifier C to obtain the prediction vector $\boldsymbol{p}_i \in \mathbb{R}^{|Y|}$, which represents the probability distribution as how likely an input example belongs to each possible class.

$$\boldsymbol{q}_k = \frac{1}{\Omega_k} \sum_{y_i = k} \boldsymbol{p}_i$$

$$\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_{|Y|}]$$

where $|Y|$ is the number of classes for labels. Ω_k is the number of graphs belonging to the k-th class in D_{val} and $\boldsymbol{q}_k$ is the average probability distribution of the k-th class.

Data Filtration

The label reliability of an example (G_i, y_i) is computed as:

$$r_i = \boldsymbol{p}_i^\top \boldsymbol{q}_{y_i}$$

The threshold θ is defined as:

$$\theta = \arg\min_{\theta} \sum_{(G_i, y_i) \in D_{val}} \Phi[(\theta - r_i) \cdot g(G_i, y_i)]$$

where $g(G_i, y_i) = 1$ if $C(G_i) = y_i$ and $g(G_i, y_i) = 1$ otherwise, and $\Phi(x) = 1$ if $x > 0$ and $\Phi(x) = 0$ otherwise.

Model Evolution Framework

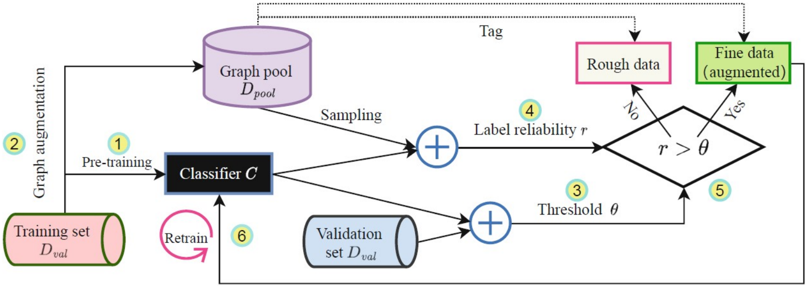

Figure 4: An example of subgraph augmentation via motif-similarity mapping; red lines is the candidates and black lines is the modified edges.

- Introduction
- Subgraph Augmentation
- Data Filtration
- Model Evolution Framework
- Experiment

Experiment

For link prediction, extract the local subgraph of target pairwise nodes, and the labels of subgraphs reflect the link existence. For node classification, extract the local subgraph of target nodes. The subgraph labels are equivalent to the corresponding node labels.

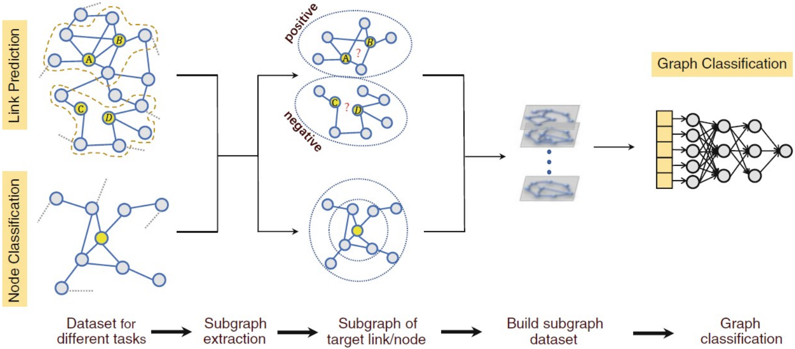

Figure 5: Unify multiple tasks into graph classification.

Results

Dataset	Mapping	Budget			Avg. RIMP
		0.10	**0.15**	**0.20**	
MUTAG	DGCNN	0.8447			–
	DGCNN + random	0.8447	0.8533	**0.8458**	+**0.38**%
	DGCNN + m-s	**0.8450**	**0.8547**	0.8436	+0.36%
PTC_MR	DGCNN	0.5775			–
	DGCNN + random	0.5739	0.5764	**0.5860**	+0.22%
	DGCNN + m-s	**0.5849**	**0.5962**	0.5733	+**1.26**%

Figure 6: Graph classification results of original and evolutive models.

$$\text{RIMP} = \frac{Acc_{en} - Acc_{ori}}{Acc_{ori}}$$

Results

Dataset	Mapping	Budget			Avg. RIMP
		0.10	**0.15**	**0.20**	
Router	GAE	0.5130			–
	VGAE	0.4999			–
	DGCNN	0.6721			–
	DGCNN + random	0.6430	0.6512	0.6694	−2.6%
	DGCNN + m-s	**0.6858**	**0.6852**	**0.6854**	**+1.7%**
Celegans	GAE	0.5256			–
	VGAE	0.5053			–
	DGCNN	0.6323			–
	DGCNN + random	0.6170	0.6125	0.6176	−2.6%
	DGCNN + m-s	**0.6353**	**0.6379**	**0.6379**	**+0.7%**

Figure 7: Link prediction results in baselines, DGCNN and evolutive models.

Results

Dataset	Mapping	Budget			Avg. RIMP
		0.10	**0.15**	**0.20**	
Blog	GCN	0.7200			–
	GAT	0.6630			–
	DGCNN	0.7453			–
	DGCNN + random	0.7502	0.7493	**0.7483**	+0.53%
	DGCNN + m-s	**0.7589**	**0.7560**	0.7457	**+1.10%**
Flickr	GCN	**0.5460**			–
	GAT	0.3590			–
	DGCNN	0.4192			–
	DGCNN + random	0.4471	0.4499	0.4505	+7.15%
	DGCNN + m-s	0.4888	0.4884	0.5014	**+17.57%**

Figure 8: Node classification results in baselines, DGCNN and evolutive models.