Subgraph Neural Networks

Emily Alsentzer, Samuel G. Finlayson, Michelle M. Li, Marinka Zitnik

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

Subgraph Property Prediction

- Given Graph $G = (V, E)$ and subgraph $G' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$.
- Each subgraph S has a label y^S and many S^C which is a set of nodes in S that are connected to each other by a path.
- The task : if a subgraph has a specific property or not Graph G: subgraphs defined by node membership

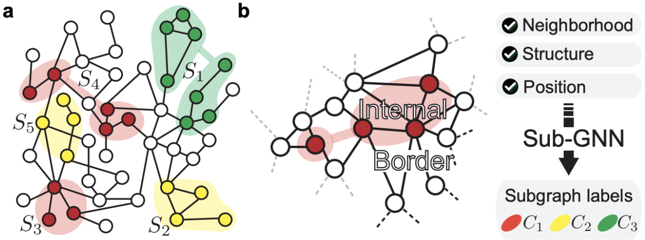


Figure: colors indicate labels

Why are subgraph property prediction challenging?

- make joint predictions over larger structures of varying sizes: do not correspond to simple k-hop, possibly disconnected and far off
- higher-order connectivity patterns: how nodes within the subgraph interact and how they interact with nodes outside the subgraph (border and extenral nodes)
- subgraphs can be localized within a region of the graph or spread out: learn about the subgraph positions within the graph
- subgraphs share edges and non edges

Formulating Subgraph Prediction

subgraph problem

Given subgraphs $(S_1, \ldots, S_n)$ the task is to get embeddings $z_S \in R^{d_S}$ for every subgraph S. SUB-GNN uses a GNN to learn a classifier $f : S \rightarrow \{1, \ldots, C\}$ $f(S) = \hat{y}_S$.

Difference from other gnns: operates directly on components

Subgraph Properties to encode

network properties that are not necessarily defined for either nodes or graphs.

SUB-GNN Channel	SUB-GNN Subchannel	
	Internal (I)	Border (B)
Position (P)	Distance between S_i's components	Distance between S_i and rest of G
Neighborhood (N)	Identity of S_i's internal nodes	Identity of S_i's border nodes
Structure (S)	Internal connectivity of S_i	Border connectivity of S_i

subgraph properties: position

- border position: distance to the rest of G
- internal: distance between the components of G

subgraph properties: neighborhood

- border neighborhood: nodes within k-hops of any node in S, each component has its own border neighborhood
- internal neighborhood

subgraph properties: structure

- internal : internal connectivity of each subgraph
- border: edges connecting internal nodes to border neighborhood

sub graph level message passing: Anchor Patches

- $\mathbb{A} = (A^1, \ldots, A^Q)$
- anchor patches are subgraphs sampled from G specific to each channel : P, N and S

sub graph level message passing: Anchor Patches to subgraph components

- $\mathbb{A} = (A^1, \ldots, A^Q)$
- anchor patches are subgraphs sampled from G specific to each channel : P, N and S
- $MSG_{X,C} = \gamma_x(A_X, S^C)p_X$
- γ is a similarity function for channel X
- $\boldsymbol{a}_{X,c} = AGG_M(MSG_X(S^C, A_X, p_X)\forall A_X in \mathbb{A}_X)$
- $\boldsymbol{h}^l_{x,c} = \sigma(\boldsymbol{W}_h[\boldsymbol{a}_{X,c}; \boldsymbol{h}^{l-1}_{x,c}])$

Property-aware output representations

- a matrix $\boldsymbol{M}_X$ where each row is an anchor set message computed by MSG_X
- pass through a non linear activation function to get $\boldsymbol{z}_{x,c}$
- for neighborhood: use $\boldsymbol{z}_{N,c} = h_{N,c}$

Agregating Property-aware output representations

- $\boldsymbol{z}_{x,c}$ for a channel x and a subgraph component c
- First aggregate using channel aggregator AGG_C
- Then aggregate using layer aggregator AGG_L
- now we have $\boldsymbol{z}_c$
- READOUT from $\boldsymbol{z}_c$ to $\boldsymbol{z}_S$
- Finally, SUB-GNN routes messages for internal and border properties (i.e., $\{P_I, P_B\}, \{N_I, N_B\}, \{S_I, S_B\}$) within subchannels for each channel P, N and S, and concatenates the final outputs

A, p_X, γ

- Sampling anchor patches
- Neural encoding of anchor patches
- Estimating similarity of anchor patches.

Sampling anchor patches

- $\phi_X : (G, S^C) \rightarrow A_X$
- Internal position sampler node $u_P \in S$, shared across all components in S
- Border position sampler node $v_P \in G$ shared across all S
- Neighborhood internal sampler node $u_N \in S^C$
- Neighborhood border sampler node $v_N \in k - hop\ of\ S^C$
- structure anchor sampler : connected component sampled from G via triangular random walks

Neural encoding of anchor patches

For position and neighborhood anchor patches, same as initial node embeddings :

$$\psi_N = \psi_p = N_i \tag{1}$$

For structure nodes:

$$\psi_S : A_S \to \boldsymbol{p} \in \mathbb{R}^d \tag{2}$$

- w fixed length triangular random walks $(u_{\pi_w(1)}, \ldots, u_{\pi_w(n)})$
- The triangular random walk samples triangular successors with probability β and non-triangular successors with probability $1 - \beta$.
- input to LSTM, use sum of hidden states $= \boldsymbol{p}$

Neural encoding of anchor patches

For structure nodes, random walk strategy:

- internal: random walks over set I $\{u|u \in A_s\}$
- neighborhood: random walks over set N $\{v|v \notin A_s\}$ limited to neighborhood k hops
- border: random walks over set $\{u|u \in I, v \in N, uv \in E\}$
- multiple random walks but single $\boldsymbol{p}$

Estimating similarity of anchor patches and subgraph components

- similarity between subcomponent of subgraph and anchor patch
- $\gamma_X : (S^C, A_S) \to [0, 1]$
- for the position channel, $\gamma_P = \dfrac{1}{d_{SP}(A_S, S_C) + 1}$
- d_{SP} is the shortest path between connected components S^C and anchor path A_S
- for structure channel, use the normalized Dynamic Time Warping (DTW) [1]
- $\gamma_S(S^C, A_S) = \dfrac{1}{DTW(d_{A_S}, d_{S_C}) + 1}$
- d_{A_S}, d_{S_C}: ordered degree sequences for the subgraph component and anchor patch

[1] used to compare time signals of varying speed

Algorithm summary

Algorithm 1: SUBGRAPH NEURAL NETWORK.

Input: Graph $G = (V, E)$; Node representations $\{\mathbf{x}_u | u \in V\}$; Subgraph S consisting of connected components $S^{(C)}$ for $c = 1, \ldots, R$; Channels N, S, and P corresponding to neighborhood, structure, and position; Subchannels I and B corresponding to internal and border subgraph topology; Anchor patch sampling function $\phi_{\text{X}} : (G, S) \to A_{\text{X}}$; Anchor patch encoder $\psi_{\text{X}} : A_{\text{X}} \to \mathbb{R}^d$; Trainable weight matrices $\mathbf{W}^{(l)}_{\text{X},z}$ and $\mathbf{W}^{(l)}_{\text{X},h}$ for each layer $l \in [1, L]$ and each channel X; Nonlinear activation function σ.

Output: Subgraph representation $\mathbf{h}_S$ for subgraph S

$\mathbf{z}^0_c = \sum_{u \in S^{(C)}} \mathbf{x}_u$

$\mathbf{h}^0_{\text{X},c} = \mathbf{z}^0_c$ for channel $\text{X} \in \{\text{N}, \text{S}, \text{P}\}$ // Channel-independent initialization

for *layer* $l = 1, \ldots, L$ **do**

  $A^{(q)}_{\text{X}} = \phi_{\text{X}}(G)$ for $q \in \{1, \ldots, Q\}$ and channel $\text{X} \in \{\text{S}_\text{I}, \text{S}_\text{B}, \text{P}_\text{B}\}$ // See Section 4.2

  $A^{(q)}_{\text{X}} = \phi_{\text{X}}(G, S)$ for $q \in \{1, \ldots, Q\}$ and channel $\text{X} \in \{\text{P}_\text{I}\}$

  for *connected component* $c = 1, \ldots, R$ **do**

    $A^{(q)}_{\text{X}} = \phi_{\text{X}}(G, S^{(C)})$ for $q \in \{1, \ldots, Q\}$ and channel $\text{X} \in \{\text{N}_\text{I}, \text{N}_\text{B}\}$

    for *channel* $\text{X} \in \{\{\text{P}_\text{I}, \text{P}_\text{B}\}, \{\text{N}_\text{I}, \text{N}_\text{B}\}, \{\text{S}_\text{I}, \text{S}_\text{B}\}\}$ **do**

      for *anchor patch* $q = 1, \ldots, Q$ **do**

        $\mathbf{p}^{(q)}_{\text{X}} = \psi_{\text{X}}(A^{(q)}_{\text{X}})$ // E.g., Algorithm 2

        $\mathbf{m}^{(l)}_{\text{X},c,q} = \text{MSG}_{\text{X}}(S^{(C)}, A^{(q)}_{\text{X}}, \mathbf{p}^{(q)}_{\text{X}})$ // (Eq.1)

        $\mathbf{M}^{(l)}_{\text{X},c}[q] = \mathbf{m}^{(l)}_{\text{X},c,q}$ if $\text{X} \in \{\text{S}_*, \text{P}_*\}$

      end

      $\mathbf{z}^{(l)}_{\text{X},c} = \sigma(\mathbf{W}^{(l)}_{\text{X},z} \cdot \mathbf{M}^{(l)}_{\text{X},c})$ if $\text{X} \in \{\text{S}_*, \text{P}_*\}$ // Property-aware output rep.

      $\mathbf{a}^{(l)}_{\text{X},c} = \text{AGG}_M(\{\mathbf{m}^{(l)}_{X,c,1}, \ldots, \mathbf{m}^{(l)}_{\text{X},c,Q}\})$ // Aggregate messages (Eq.2)

      $\mathbf{h}^{(l)}_{\text{X},c} = \sigma(\mathbf{W}^{(l)}_{\text{X},h} \cdot [\mathbf{a}^{(l)}_{\text{X},c}; \mathbf{h}^{(l-1)}_{\text{X},c}])$ // Order-invariant hidden rep. (Eq.2)

    end

    $\mathbf{z}^{(l)}_c = \text{AGG}_C(\mathbf{h}^{(l)}_{\text{N},c}, \mathbf{z}^{(l)}_{\text{S},c}, \mathbf{z}^{(l)}_{\text{P},c})$ // Aggregate channels

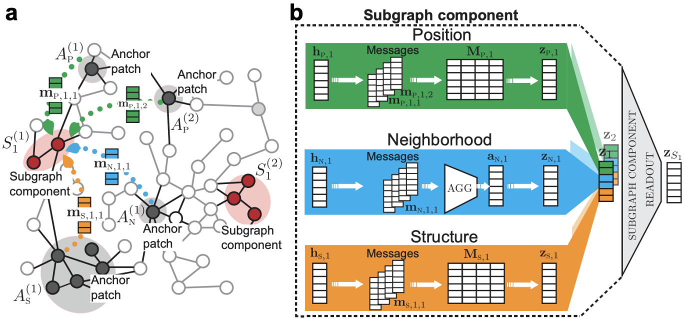
  end

end

$\mathbf{z}_c \leftarrow \text{AGG}_L(\{\mathbf{z}^{(0)}_c, \ldots, \mathbf{z}^{(L)}_c\})$ // Aggregate layers

$\mathbf{h}_S = \text{READOUT}(\{\mathbf{z}_1, \ldots, \mathbf{z}_R\})$ // Aggregate components

Sub-GNN Figure



Computational Complexity and model extensions

- function of (number of subgraphs, size of the subgraphs)
- also depends on number of anchor patches: prespecified and fixed
- possible to use other types of similarity or joint learning of node embeddings

Synthetic Experiments

- subgraph properties: density, cut ratio, coreness, component
- density: internal structure(250 subgraphs of size 20)
- cut ratio: border structure(250 subgraphs of size 20)
- coreness: average core number of the subgraph, tests border structure and position (221 subgraphs of size 20)
- component: the number of subgraph components,(250 subgraphs with 15 nodes per component) tests internal and external position.

Real World Datasets

- PPI-BP : 1591 subgraphs with 6 labels, labeled using Biological Process Ontology from MSigDB, Subgraphs are collections of proteins in the PPI network that are involved in the same biological process
- HPO-METAB : graphs of causal genes and symptoms, with subgraphs defined by symptoms. 2400 subgraphs with 6 labels from metabolic disorders: lysosomal, energy, amino acid, carbohydrate, lipid, and glycosylation.
- HPO-NEURO : about neurological disorders
- EM-USER : subgraphs about work out routines with 1343 sugraphs. Label is gender

HPO dataset challenges

- distinguishing subcategories of similar diseases (a challenge for averaging-based methods),
- exhibit class distributional shift between train and test, and have been designed to require inductive inference to nearby phenotypes using edges in the graph.
- require distinguishing subcategories of similar diseases (a challenge for averaging-based methods),

Baselines

- AVG: average of the node embeddings of the subgraph
- MN-GIN and MN-GAT: use virtual node to represent a subgraph
- s2v-N, s2v-S, s2v-NS: suhgraph 2 Vec
- GC: treat each subgraph as standalone graph using average of node embeddings
- pretrained: GIN on link prediction

Simulation microF1

Method	DENSITY	CUT RATIO	CORENESS	COMPONENT
SUB-GNN (Ours)	**0.919**±**0.016**	**0.629**±**0.039**	**0.659**±**0.092**	**0.958**±**0.098**
Node Averaging	0.429±0.041	0.358±0.055	0.530±0.050	0.516±<0.001
Meta Node (GIN)	0.442±0.052	0.423±0.057	0.611±0.050	0.784±0.046
Meta Node (GAT)	0.690±0.021	0.284±0.052	0.519±0.076	0.935±<0.001
Sub2Vec Neighborhood	0.345±0.066	0.339±0.058	0.381±0.047	0.568±0.039
Sub2Vec Structure	0.339±0.036	0.345±0.121	0.404±0.097	0.510±0.013
Sub2Vec N & S Concat	0.352±0.071	0.303±0.062	0.356±0.050	0.568±0.021
Graph-level GNN	0.816±0.068	0.377±0.058	0.419±0.070	0.526±0.081

Real World microF1

Method	PPI-BP	HPO-NEURO	HPO-METAB	EM-USER
SUB-GNN (Ours)	**0.324**$\pm$**0.013**	**0.632**$\pm$**0.010**	**0.537**$\pm$**0.023**	**0.751**$\pm$**0.021**
Node Averaging	0.289$\pm$0.043	0.490$\pm$0.059	0.443$\pm$0.063	0.744$\pm$0.086
Meta Node (GIN)	0.277$\pm$0.040	0.233$\pm$0.086	0.151$\pm$0.073	0.550$\pm$0.025
Meta Node (GAT)	0.308$\pm$0.032	0.259$\pm$0.063	0.138$\pm$0.034	0.536$\pm$0.047
Sub2Vec Neighborhood	0.309$\pm$0.023	0.211$\pm$0.068	0.132$\pm$0.047	0.503$\pm$0.035
Sub2Vec Structure	0.307$\pm$0.013	0.223$\pm$0.065	0.124$\pm$0.025	0.742$\pm$0.023
Sub2Vec N & S Concat	0.295$\pm$0.011	0.206$\pm$0.073	0.114$\pm$0.021	0.536$\pm$0.047
Graph-level GNN	0.291$\pm$0.026	0.577$\pm$0.015	0.480$\pm$0.026	0.505$\pm$0.041

Channel Ablation Analysis

aligns with their inductive biases

SUB-GNN Channel	DENSITY	CUT RATIO	CORENESS	COMPONENT
Position (P)	0.758±0.046	0.516±0.083	0.581±0.044 ✓	**0.958±0.098** ✓
Neighborhood (N)	0.777±0.057	0.313±0.087	0.485±0.075	0.823±0.089
Structure (S)	**0.919±0.016** ✓	**0.629±0.039** ✓	**0.663±0.058** ✓	0.600±0.170
All (P+N+S)	0.894±0.025	0.458±0.101	0.659±0.092	0.726±0.120