Trying to learn the subgraph isomorphism

G. Renton, S. Adam, P. Héroux, B. Gaüzère

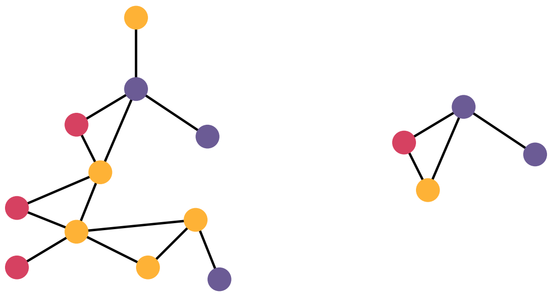
University of Rouen

May 3, 2019

Introduction

Subgraph Isomorphism

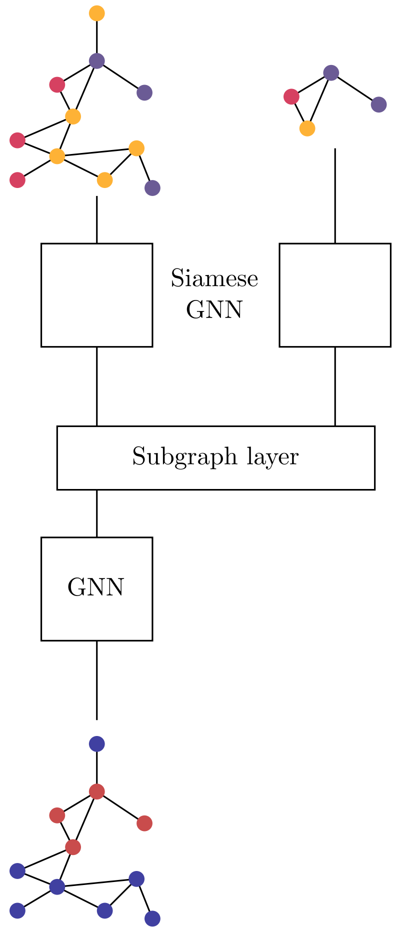
- Find a graph inside a bigger graph.
- NP-complete problem
- Yet existing algorithms with nominal values.



Question :

- Can we learn to solve the subgraph isomorphism to use it on real valued graph ?

Tested Architecture

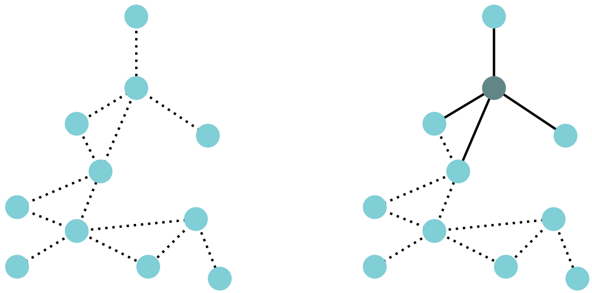


Graph Neural Network

GNN basic idea :

- Update nodes labels depending on :
  - The node actual label.
  - Its neighbourhood labels.
  - Possibly its edges labels.

$$h_v^k = f_k(h_v^{(k-1)}, h_{N_{(v)}}^{(k-1)}, e_{N_{(v)}})$$



GNN Models

Simplest way

Nodes update depending on their neighbourhood (message passing strategy).

$$\tilde{x_v} = x_v \cdot W + \sum_{w \in N(v)} x_w \cdot W$$

Working on tensors

$$\tilde{X} = A \cdot (X \cdot W)$$

with $A \in \mathbb{R}^{N \times N}$, $X \in \mathbb{R}^{N \times F}$, $W \in \mathbb{R}^{F \times H}$
2 major limitations :

- No difference for the node and its neighbourhood.
- Do not take into account edges attributes

GNN Models

Make a difference between the node and its neighbourhood

Easy way :

$$\tilde{x_v} = x_v \cdot W_0 + \sum_{w \in N(v)} x_w \cdot W_1$$

Another way : use attention network.

With tensors

$$\tilde{X} = X \cdot W_0 + A \cdot (X \cdot W_1)$$

with $W_0, W_1 \in \mathbb{R}^{F \times H}$
Problem : still don't take into account edges attributes

GNN Models

Edge Network

Idea : use a neural network over the edge attributes.

$$\tilde{x_v} = x_v \cdot W_0 + \sum_{w \in N(v)} x_w \cdot W_1 \cdot (e_{vw} \cdot W_a)$$

With tensors

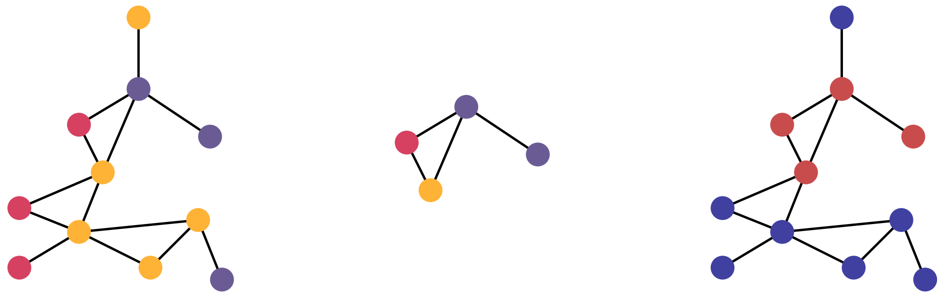
$$\tilde{X} = X \cdot W_0 + (X \cdot W_1) \cdot (W_A \cdot A)$$

with $W_A \in \mathbb{R}^{E \times H^2}$. Thus, $(W_A \cdot A)$ is reshaped to get a squared matrix.

Using GNN for subgraph isomorphism

Model objective

- 2 inputs : the graph and the subgraph.
- 1 output : the graph, with nodes of the subgraph activated.



Basis Idea

We want to update the graph nodes states, depending on the subgraph we are looking for.

Subgraph Layer

Model

$$\tilde{X} = X \cdot (\bar{X}_{sub} \cdot W)$$

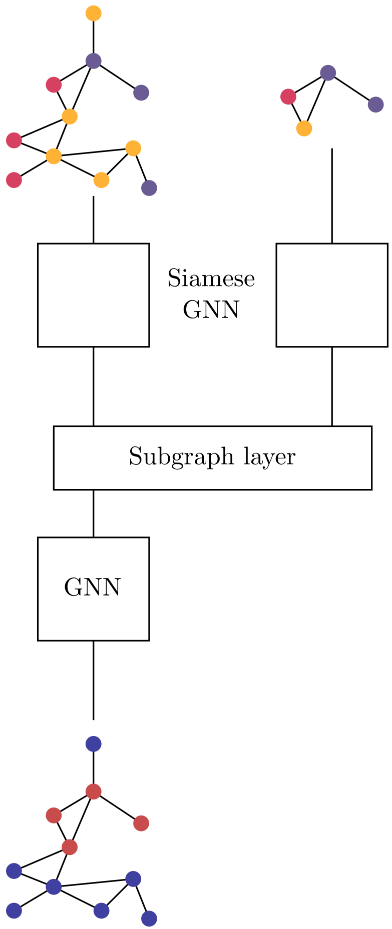
Where $\bar{X}_{sub}$ is a graph embedding of the subgraph

Embedding

Multiple ways of providing a graph embedding.

- Mean, max, sum of the nodes features.
- Learning a graph embedding
- Features covariance.

Tested Architecture



Experiments

Dataset presentation

- Randomly generated graphs.
- Each graph has between 11 and 26 nodes.
- Each node has a letter for attribute (12 letters possibles).
- No edge attribute

Random generation

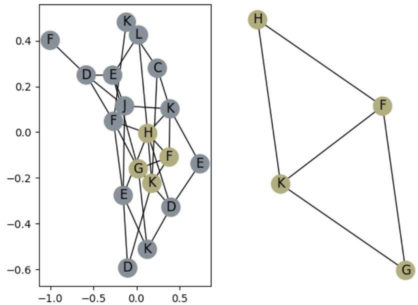
- We add a random node
- We add a random number of node between 10 and 25 iteratively.
- Each time a node is added, we connect it to each other node with a probability of 0.2
- If a node is not connected to anyone, we link it to the last one.

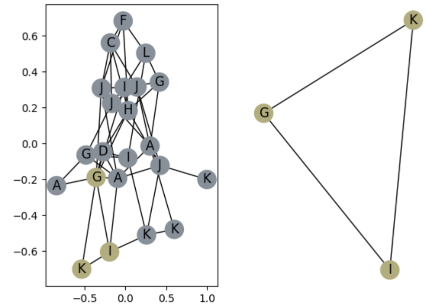
Experiments

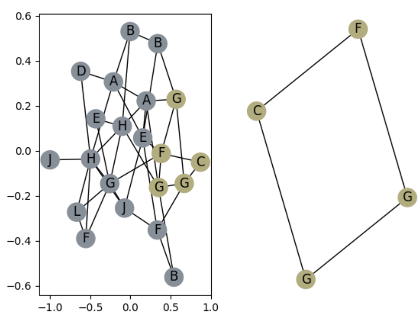
Graph and subgraph generations

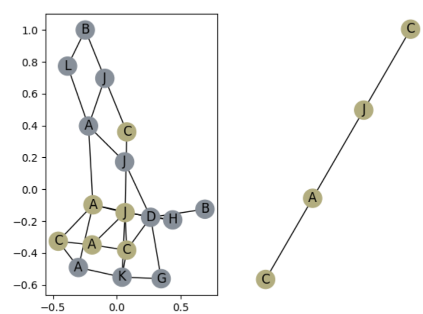
- We create 100 000 graphs with previous method.
- For each generated graph, we create 5 subgraphs.
- Subgraph generation :
  - Random number of nodes is chosen, between 3 and 4.
  - A starting node is randomly chosen
  - Random walk from the starting node to create the subgraph.
- Finally, we have 500 000 pairs of graphs and subgraphs.

Experiments
Some data

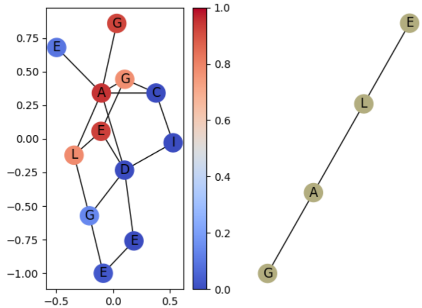


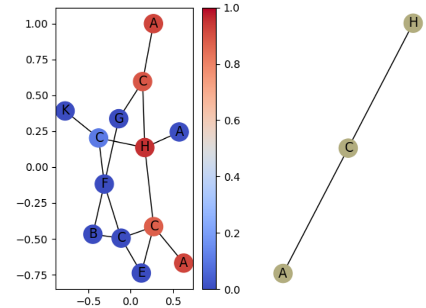


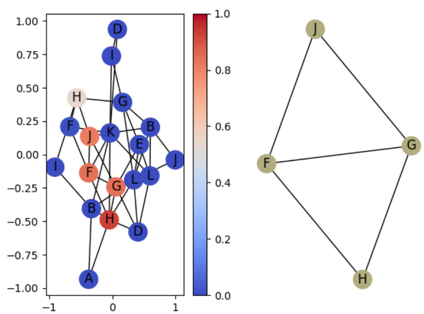


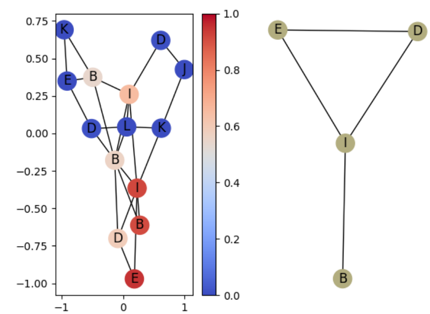


Experiments
Some results









Conclusion

Future Work

- Work on the embedding of the subgraph layer
- More complete architecture to take global graph into account.
- More challenging data (Bigger graphs, real values).
- Real data.