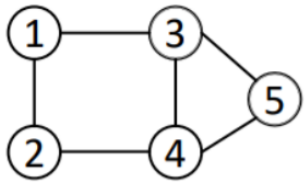
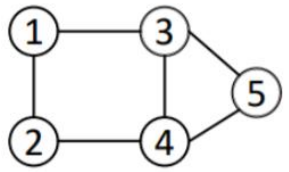
Summary Questions of the lecture

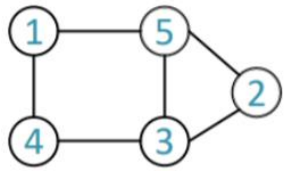
- Why is the Graph Generation Tasks hard?
- GGT is difficult because 1) the output graph space is large and variable, 2) a single graph has multiple representations, and 3) determining edge connections depend on long-range and extensive information.



0	1	1	0	0
1	0	0	1	0
1	0	0	1	1
0	1	1	0	1
0	0	1	1	0

5 nodes: 25 values
1K nodes: 1M values





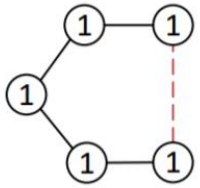
0	1	1	0	0
1	0	0	1	0
1	0	0	1	1
0	1	1	0	1
0	0	1	1	0

0	0	0	1	1
0	0	1	0	1
0	1	0	1	1
1	0	1	0	0
1	1	1	0	0

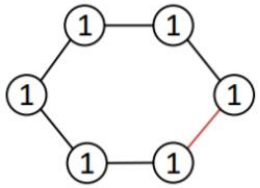
Same graph

Very different representations!

Example: Generate a ring graph on 6 nodes:



Shouldn't have edge!



Should have edge!

Existence of an edge may depend on the entire graph!

Summary Questions of the lecture

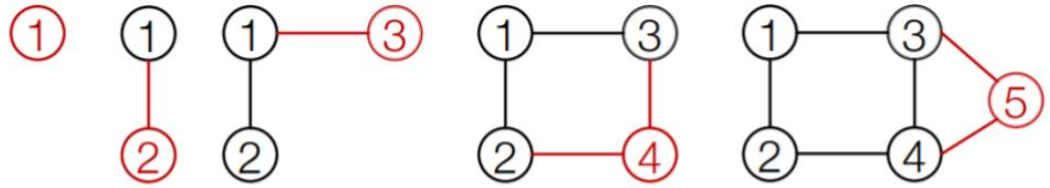
- Discuss the machine learning background of auto-regressive model to obtain the graph generative models.
- Auto-regressive models sequentially predict graph-generation actions (adding a node and one or more edges) on the current time step based on all past actions. These actions are represented by random variables, whose distribution is estimated from the given data distribution with a parametric distribution model in a maximum likelihood manner, and then actions are sampled from the distribution by the estimated model.

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}\sim P_{data}(\boldsymbol{x})} \log\, P_{model}(\boldsymbol{x}|\boldsymbol{\theta})$$

$$P_{model}(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_{\boldsymbol{t=1}}^{\boldsymbol{n}} P_{model}(x_t|x_1, \ldots, x_{t-1}, \boldsymbol{\theta}) \qquad x_t \sim P_{data}(\boldsymbol{x})$$

Summary Questions of the lecture

- Explain GraphRNN which is a sequence process to generate a graph with a node ordering.
- GraphRNN generates a graph through a sequenc of sequences. That is, on time-step t, node t is first added to the graph, which forms node level sequence. Then, at the node level action t, edge level sequence is conducted by determining whether the node is connected to each of the existing nodes $1.2.\cdots.t-1$.



$S^{\pi} = (S_1^{\pi} , S_2^{\pi} , S_3^{\pi} , S_4^{\pi} , S_5^{\pi})$

"Add node 1" ... "Add node 5"

$S_4^{\pi} = (S_{4,1}^{\pi} , S_{4,2}^{\pi} , S_{4,3}^{\pi})$

"Not connect 4, 1" "Connect 4, 2" "Connect 4, 3"

0 1 1