Q3) Graph learning under Big data setting

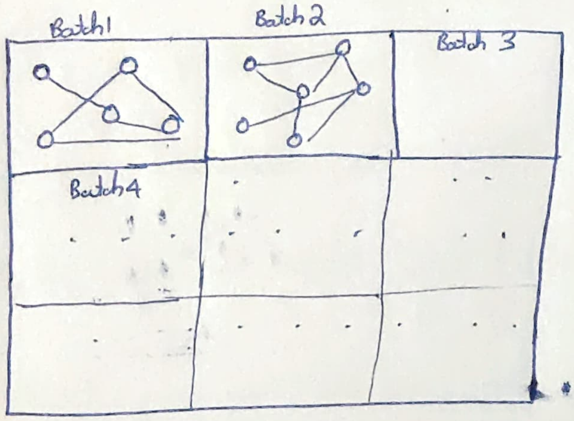
→ In the Big data setting, it is not really possible to process entire data every iteration.

→ We can tackle this problem by implementing something like Stochastic Gradient kind of approach, for which

→ we split data into smaller batches.

→ <u>Problem</u>:

If we try to learn all batches (say b) individually, that will result in 'b' disconnected graphs.



So we need to find some ways to find connections between batches to make the learned graph meaningfull, which requires some stronger assumptions from our side

Solution :

⇒ The idea is that we will be retaining some nodes (say batch 1) and integrate that node also in the 2nd batch that will be coming.

eg) Let no of nods in a batch = n_b

Let No of retained nodes from batch 1 = γ_1

∴ No of nodes in batch 2 = $n_b = \gamma_1 + n_{b2}$

where n_{b2} → Number of new nodes which were not from batch 1

→ By doing this we can ensure that the learned graph will be having sufficient number of inter connections.

→ So what will be the best 'γ' nodes which can be retained?

→ We can retain ~~for~~ those 'γ' nodes which have the lowest degree

→ The intuition (or assumption) is that node which have the lowest degree might not be similar to the nodes in that batch and might have stronger connections with other batches nodes.

→ We have implicitly assumed that if a node in a batch has sufficiently large degree D, it will not be having any more edge with other nodes in other batches.

→ This is my thoughts for the given problem statement 3.