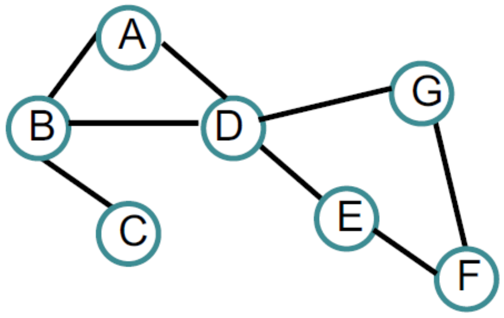
School of Computing and Information Systems (CIS)
The University of Melbourne
COMP90073
Security Analytics
Tutorial exercises: Week 9

1. How the following GCN propagates information across the graph (only state the first 3 layers/depth) to compute node features?

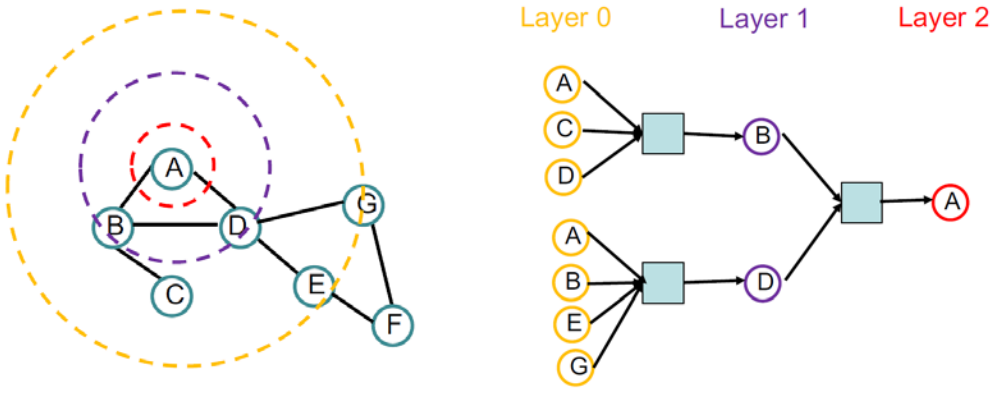


Solution:

Model can be of arbitrary depth:

- Nodes have embeddings at each layer
- Layer-0 embedding of node u is its input feature, x_u
- Layer-K embedding gets information from nodes that are K hops away

Every node defines a computation graph based on its neighbourhood
Note: The square boxes in the following diagrams represent neural networks.



2. (a) What are the applications of Graph Convolutional Networks (GCNs)?

   (b) What kind of dataset it may most suit regarding anomaly detection?

3. A database has five transactions. Let *min_sup* = 60% and min_conf = 80%.

   (a) Find all frequent *itemsets* using Apriori and FP-growth, respectively.

   (b) Compare the efficiency of the two mining processes.

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I ,E}

Solution:

- Apriori:

C1 =

m	3
o	3
n	2
k	5
e	4
y	3
d	1
a	1
u	1
c	2
i	1

L1 =

m	3
o	3
k	5
e	4
y	3

C2 =

mo	1
mk	3
me	2
my	2
ok	3
oe	3
oy	2
ke	4
ky	3
ey	2

L2 =

mk	3
ok	3
oe	3
ke	4
ky	3

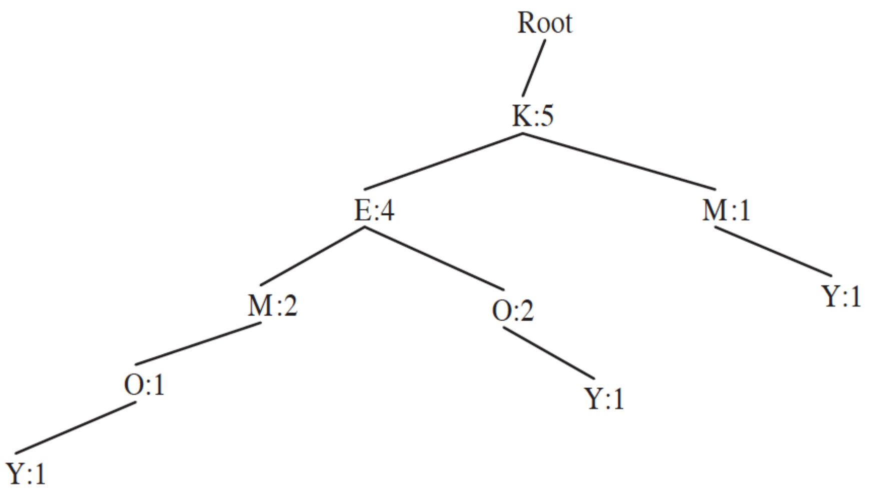
C3=

oke	3
key	2

L3=

oke	3

- FP-growth:



Item	Conditional Pattern Base	Conditional Tree	Frequent Pattern
y	{k,e,m,o: 1}, {k,e,o: 1}, {k,m: 1}	<k: 3>	{k,y: 3}
o	{k,e,m: 1}, {k,e: 2}	<k: 3>, <e: 3>	{k,o: 3}, {e,o: 3}, {k,e,o: 3}
m	{k,e: 2}, {k: 1}	<k: 3>	{k,m: 3}
e	{k: 4}	<k: 4>	{k,e: 4}

Efficiency comparison: Apriori has to do multiple scans of the database while FP-growth builds the FP-Tree with a single scan. Candidate generation in Apriori is expensive (owing to the self-join), while FP-growth does not generate any candidates.

4. The price of each item in a store is nonnegative. The store manager is only interested in patterns of the form: “one free item may trigger $200 total purchases in the same transaction”. State how to mine such patterns efficiently.

   Solution:

   To efficiently mine patterns of the form “one free item may trigger $200 total purchases in the same transaction”, we need to:

   - Find the set of frequent “free” 1-itemsets. Let us call this set S1.
   - Generate the set of frequent itemsets whose price is no less than $200, using FP-growth. Let us call this set S2.

     This is a monotonic constraint so we do not have to use the “generate-and-test” approach at each iteration, which will save us some unnecessary computation. If we have a frequent itemset whose price is at least $200, then we do not need to check the constraint for any superset of that itemset. This is because any additional items added to the itemset will only increase the price total, which is already at least $200. All that needs to be checked is if that superset is frequent.

     The reason that we use FP-growth and not Apriori is because Apriori discards frequent itemsets whose price is less than $200, and therefore is not be able to discover supersets of these itemsets that satisfy the constraint. FP-growth does not suffer from this problem because it keeps all of the information about the database in a tree structure.
   - Find the frequent itemsets from the set S1S2, where S1 is the set of frequent “free” 1-itemsets and S2 is the set of frequent itemsets whose price is no less than $200.
   - Generate patterns of the form S1⇒S2 from the frequent itemsets found above that meet the minimum confidence threshold.