Exam

Course Data Mining for Networks

2020-2021

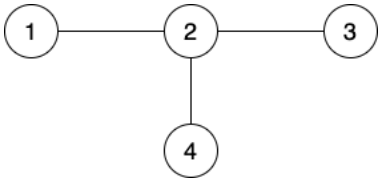
Part 1 : Graph Kernels (9 points)

I. General questions about the course (5,5 points)

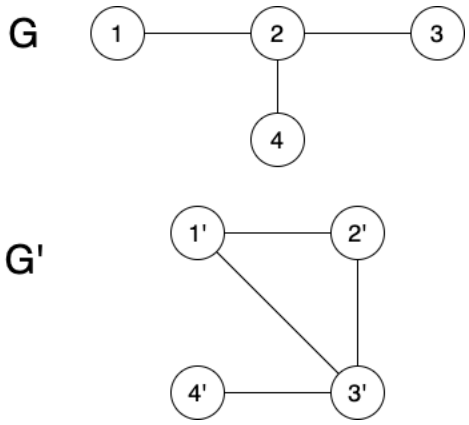
1. Explain what is the graph isomorphism problem? Write the pseudo-code of the most efficient algorithm you know to solve it. What is its complexity? What other methods to you know to compare graphs? What are their advantages and drawbacks? (pseudo-code+figures+max 15 lines)
2. Explain what is the ideal graph kernel. Provide an example with a graph of at least 4 vertices and compute its features. What are the advantages and drawback of the ideal graph kernel? (figures+max 20 lines of text).
3. Explain what is the gradient descent method and how it works. What is the use of the method for learning algorithms? (figures+max 15 lines of text).
4. Present an example of anomaly-based detection to mitigate network attacks. (figures+max 10 lines of text).

II. Random Walk Kernels (5,5 points)

1. Compute the number of walks of length 1, 2, 3 of the following graph using matrix multiplication.

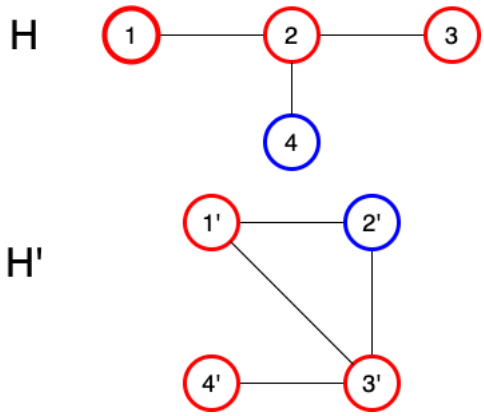


2. Build the direct product G x G' of the two following graphs G and G'.



3. How many walks are there in the direct product graph G x G' between the vertices (2,1') and (3,3') of lengths 1, 2, 3, 4, 5? Exhibit them. *If there are too many walks for some values, explain how to compute them.*

4. Build the direct product H x H' of the two following labeled graphs H and H'. Nodes have 2 labels: Red or blue. Edges do not have labels



5. How many matching walks are there in the direct product graph H x H' between the vertices (2,3') and (1,1') of lengths 1, 2, 3, 4, 5? Exhibit them. *If there are too many walks for some values, explain how to compute them.*

Part 2 : Reinforcement Learning

I. General questions about the course (5 points)

1) Explain what is the goal for the agent in Reinforcement Learning. Depict the learning process with a figure and explain what are the mathematical notations.
2) What is one potential problem with a Q-learning agent that always chooses the action which maximizes the Q-value?
3) Describe two ways to force a Q-learning agent to explore.
4) What is the learning rate in the Q-value function? What does it mean when it equals 0? 1?
5) In a MAB problem, explain what is an action-value method. Provide two action selection rules. What are their advantages and drawbacks?

II. Q-Learning Algorithm (4 points)

Assume we are an agent in a 3x2 grid-world, as shown in the figure below. We start at the bottom left node (1) and finish in the top right node (6).

When node 6 is reached, we receive a reward of 10 and return to the start for a new episode. On all other actions that not lead to state 6, the reward is 1.

4	5	**finish** 6
start 1	2	3

In each state we have four possible actions: up, down, left and right. For each action we move deterministically in the specific direction on the grid. Assume that we cannot take actions that bring us outside the grid.

We will apply some episode of the Q-Learning algorithm with $\alpha = 0.5$ and $\Upsilon = 0.7$.

1) 1st episode: Initializing the Q-table to 0 at the beginning, what are the estimates after the episode: 1 – up – 4 – right – 5 – right – 6?
2) The current estimates of *Q(s; a)* are given in the below table:
   (all blank boxes have value 0, so has the sixth row for final state 6).

Q(1,up)=4			Q(1,right)=3
Q(2,up)=6		Q(2,left)=3	Q(2,right)=8
Q(3,up)=9		Q(3,left)=7	
	Q(4,down)=2		Q(4,right)=5
	Q(5,down)=6	Q(5,left)=5	Q(5,right)=8

The agent applies here the ε-greedy strategy for action selection and samples the following episode: 1 – right – 2 – right – 3 – up – 6.

a) On which time steps can we claim that the agent has performed the random action choice? On which time steps could this possibly have occurred?
b) What is the best strategy so far to go from state 1 to 6?
c) Update the corresponding *Q(s; a)* following the given episode and explain the information you get now for the strategy. What solution do you propose to get a final good strategy?