CS-E4650 - Assignment 4

Arttu Häkkinen - 596077

November 26, 2021

Task 2

a)

The calculations for maximum common sub-graph sizes for all necessary combinations are done by hand and can be found as scanned PDF file (MCSs.pdf) from the zip file which was attached to this report in the assignment return box.

The Union-normalized distance between two graphs G1 and G2 is calculated with the following equation

$$UDist(G1, G2) = 1 - \frac{|MCS(G1, G2)|}{|G1| + |G2| - |MCS(G1, G2)|},$$

where $|G1|$ and $|G2|$ are the sizes of the graphs G1 and G2 respectively, and $|MCS(G1, G2)|$ is the number of nodes in maximum common sub-graph of the two graphs G1 and G2. The calculated Union-normalized distances (script in the zip file) to each graph from class M graphs are presented in the table below.

	G1	G2	G3	G4	G5	G6
G1	0.0	0.4	0.4	0.53	0.77	0.24
G3	0.4	0.53	0.0	0.56	0.81	0.44
G5	0.77	0.75	0.81	0.84	0.0	0.81

The Maximum-normalized distance between two graphs G1 and G2 is calculated with the following equation

$$MDist(G1, G2) = 1 - \frac{|MCS(G1, G2)|}{max\{|G1|, |G2|\}},$$

where $max\{|G1|, |G2|\}$ is greater of the graph sizes of the graphs G1 and G2 respectively, and $|MCS(G1, G2)|$ is the number of nodes in maximum common sub-graph of the two graphs G1 and G2. The calculated Maximum-normalized distances (script in the zip file) to each graph from class M graphs are presented in the table below.

	G1	G2	G3	G4	G5	G6
G1	0.0	0.31	0.31	0.4	0.64	0.24
G3	0.31	0.36	0.0	0.47	0.71	0.41
G5	0.64	0.64	0.71	0.73	0.0	0.71

The two nearest neighbors for class M graphs based on both distances are presented in the table below. The nearest neighbors don't include the graph itself.

	Udist	Mdist
G1	G2 and G3	G2 and G3
G3	G1 and G6	G1 and G2
G5	G1 and G2	G1 and G2

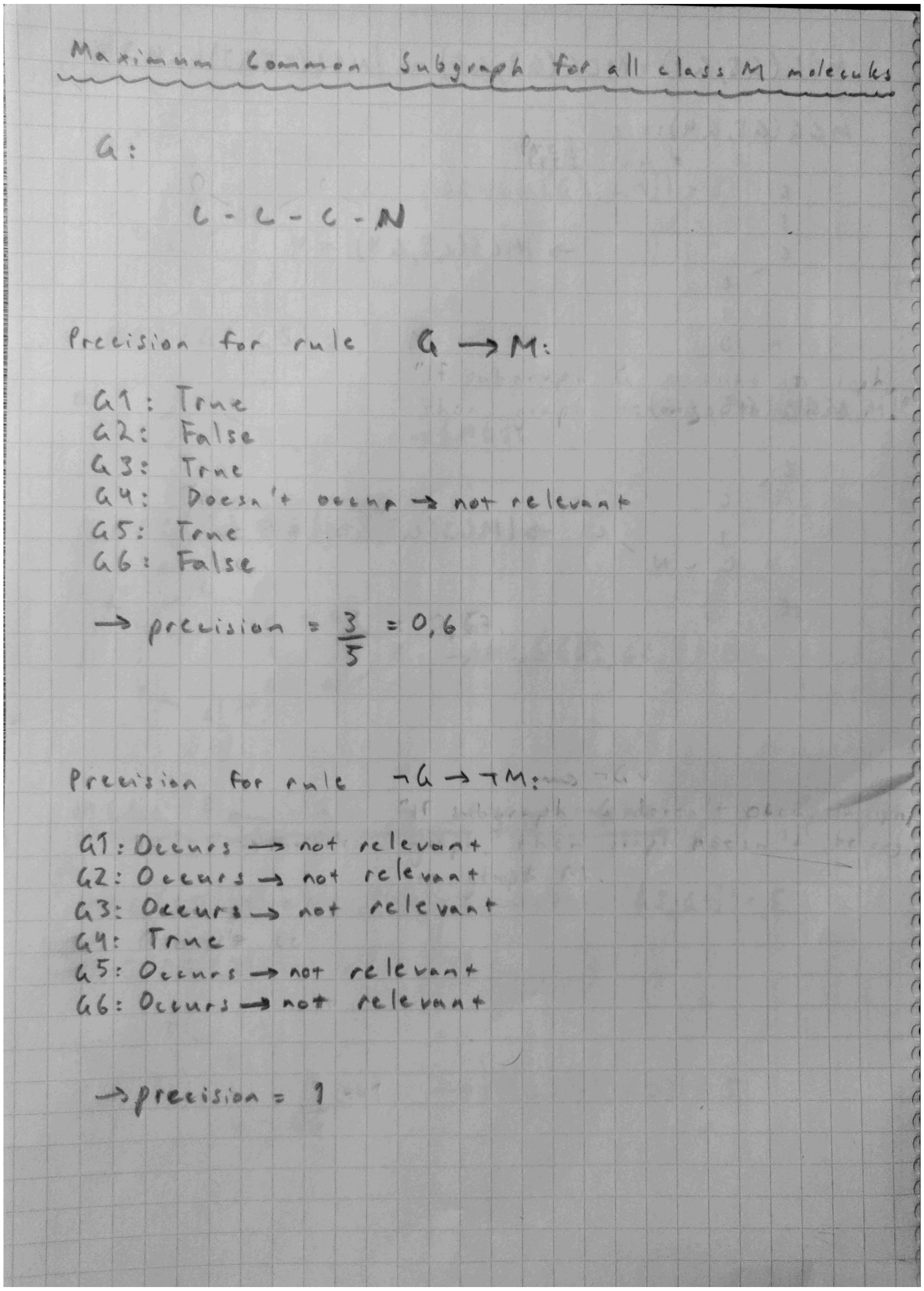
Union-normalized distance separates class M molecules slightly better from other molecules, since 4 out of 6 nearest neighbors are from class M to the class M molecules. The Maximum-normalized distance managed to only capture 3 out of 6 nearest neighbors correctly.

b)

The definition of presicion is the fraction of positive/true instances among the relevant instances. So in our case, when the rule is $G \rightarrow M$, oly the molecules that have the subgraph G are relevant, and on the other hand when the rule is $\neg G \rightarrow \neg M$, only the molecules where G doesn't occur are relevant. The rules can be interpreted as

- $G \rightarrow M$: "If G occurs in a graph X, then it follows that graph X belongs to class M"
- $\neg G \rightarrow \neg M$: "If G does not occur in graph X, then it follows that graph X does not belong to class M"

The calculations are done by hand and are attched below.



Maximum Common Subgraph for all class M molecules

G:

C - C - C - N

Precision for rule $G \rightarrow M$:

G1: True
G2: False
G3: True
G4: Doesn't occur → not relevant
G5: True
G6: False

→ precision = $\frac{3}{5}$ = 0,6

Precision for rule $\neg G \rightarrow \neg M$:

G1: Occurs → not relevant
G2: Occurs → not relevant
G3: Occurs → not relevant
G4: True
G5: Occurs → not relevant
G6: Occurs → not relevant

→ precision = 1

So, the precision for rule $G \rightarrow M$ is 0.6 and for rule $\neg G \rightarrow \neg M$ is 1.

c)

First thing that would need to be done is to search for the frequent sub-graphs based on the given minimum frequency threshold. We would somehow intelligently need to figure out such threshold, since we would not want to consider all possible sub-graphs, obviously.

When searching for the frequent sub-graphs, we should utilize the monotonicity of frequency. This means that if we find a graph X that is a sub-graph of graph Y, we know that the frequency of the graph X has to be at least as high as the frequency of the graph Y.

When we have found the frequent sub-graphs and decided some initial set of candidates, then we can apply the GraphAPriori algorithm. This is also where the minimum frequency is important. If we had too low minimum frequency, the algorithm would go through more unnecessary iterations and the computation time would be high. On the other hand, if our minimum frequency was too high, we would not probably find all meaningful statistical association between sub-graphs and attributes, since we would miss some small sub-graph to attribute implications. Adjusting the minimum frequency threshold is our best way to effect how meaningful patterns we find and what is the computational cost of the procedure. So, optimizing this hyperparameter as much as possible is really important.

In general this method is good. The procedure of selecting candidates (can be huge!) and selecting the minimum frequency are hard tasks, but these are executed properly and with care, the possibility of finding good associations is high.

The problem with this method is that the number of candidate patterns may be really large. Also all the isomorphism problems (identifying sub-graphs for joining, redundancy checking, monotonicity pruning, frequency counting) are exponentially complex problems and we have many of them so that can be a problem computationaly. This will, on the other hand become easier if we have many unique labels, we are only searching for small sub-graphs or edge-based join is used (this usually results in less candidates).

References