Methods of data mining

Assignment 4

Task 2

Miro-Markus Nikula

712686

a) To define the two nearest neighbors for each of the class M molecules, we need to compute the union-normalized distance (Udist) and maximum-normalized distance (Mdist) for all the pairs containing class M molecules. The graph sizes are in this case the number of atoms in the molecules and the MCS-value is the size of the maximum common subgraph between two molecules. The formulae are the same as presented in the lecture 9 and in the course textbook. The two simple methods for Udist and Mdist can also be found as python code in the Task2.zip.

G1 (serotonin):

MCS	Udist	Mdist
G1, G2	0.59	0.46
G1, G3	0.40	0.31
G1, G4	0.60	0.47
G1, G5	0.83	0.71
G1, G6	0.24	0.24

G1's nearest neighbors are the ones with the smallest distances, in this case G6 and G3.

G3 (dopamine):

MCS	Udist	Mdist
G3, G1	0.40	0.31
G3, G2	0.53	0.36
G3, G4	0.56	0.47
G3, G5	0.81	0.71
G3, G6	0.53	0.47

G3's two nearest neighbors are G1 and G2. This is quite interesting since G2 doesn't belong to the same class, unlike G6.

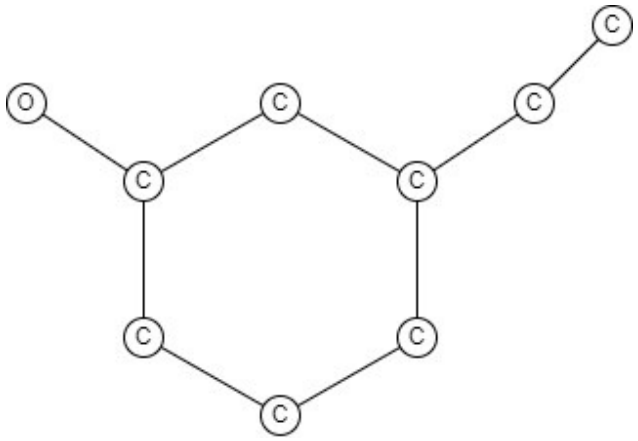
G6 (melatonin):

MCS	Udist	Mdist
G6, G1	0.24	0.24
G6, G2	0.53	0.47
G6, G3	0.53	0.47
G6, G4	0.61	0.47
G6, G5	0.81	0.71

G6's nearest neighbor is G1. G3 and G2 would be the second nearest with an equal distance to G6.

With no exception I received larger distances with Udist than Mdist, which means that Udist is the method that separates class M molecules better from other molecules. Although, it also separates G3 more from the other molecules of class M.

b) The maximum common subgraph of class M molecules is presented below:



We have 6 molecules in total: 3 of which belong to class M, 3 don't. Only the class M molecules have this substructure. This would mean that the confidence for both G -> M and not G -> not M, where G means includes the subgraph, is 100%. From the lecture 7 slides:

$\phi (X \rightarrow C) = P(XC) / P(X) \Rightarrow P(GM) / P(G) = 3 / 3 = 100\%$.

In this context the subgraph predicts class M extremely well. Of course, the sample size is very small, but in some cases a molecular structure is known to only belong to a certain group of chemical compounds (ie. phenols).

c) To find the most common significant statistical associations between subgraphs and attributes it is useful to find the most frequent subgraphs. The GraphApriori-algorithm is designed to this. Carbon is the basis of organic chemistry and therefore it will be present in most if not all the compounds of the database. For this reason, I wouldn't be too interested in very small subgraphs containing just a few carbon atoms. The most characteristic substructures of toxics, drugs and amino-acids I believe to include at least four or five atoms, and this is why any subgraphs smaller than that should not be considered as associations. This way some time could be saved.

In GraphApriori all the subgraphs that have a frequency less than the minimum frequency are pruned out. Defining a sensible minimum frequency is therefore crucial. I'm no chemist so I can't say for sure what is considered as significant, but I'd set the minimum frequency to maybe around 10% of the size of the database.

After recognizing at least these two factors I could start to run the GraphApriori algorithm and from the most frequent subgraphs I could then find the ones that are not too small, but still have a high

frequency and hopefully find some significant associations. Chemical compounds are challenging because they many times consist of the same atoms. There can be many carbon-, hydrogen-, oxygen- and nitrogen-atoms and therefore there are no unique labels, and the candidate list would be very large since there are so many matchings between graphs. Most likely the same subgraph will be created many times and there's a need to perform graph isomorphism for redundancy checking.

In my opinion this approach would be computationally very heavy, but at least the molecular structure graphs are not that large, at least compared to some other networks like the internet. I don't know if there would be a better algorithm than GraphApriori and as the size of the dataset is unknown, I would say that this would be a decent approach to find the most significant associations between subgraphs and attributes.