2 Task 2

2.1 Task2.a

According to formula in lecture 9, we can compute nearest neighbours of class M molecules using Udist and Mdist. M contains 3 molecules,so we average MCG distances on all molecules. The results are below:

Udist:

$$\begin{aligned} Udist(M, G_4) &= \frac{1}{3}(Udist(G_1, G_4) + Udist(G_2, G_4) + Udist(G_3, G_4)) \\ &= \frac{1}{3}[(1 - \frac{|MCG(G_1, G_4)|}{|G_1| + |G_4| - MCG(G_1, G_4)}) + Udist(G_2, G_4) + Udist(G_2, G_4)] \\ &= \frac{1}{3}[(1 - \frac{9}{13 + 15 - 9}) + 0.71 + 0.56] \\ &= 0.60 \end{aligned} \quad (1)$$

$$Udist(M, G_5) = 0.80 \quad (2)$$

$$Udist(M, G_6) = 0.37 \quad (3)$$

Mdist:

$$\begin{aligned} Mdist(M, G_4) &= \frac{1}{3}(Mdist(G_1, G_4) + Mdist(G_2, G_4) + Mdist(G_3, G_4)) \\ &= \frac{1}{3}[(1 - \frac{|MCG(G_1, G_4)|}{max\{|G_1|, |G_4|\}}) + Mdist(G_2, G_4) + Mdist(G_3, G_4)] \\ &= \frac{1}{3}[(1 - \frac{9}{15}) + 0.60 + 0.47] \\ &= 0.51 \end{aligned} \quad (4)$$

$$Udist(M, G_5) = 0.69 \quad (5)$$

$$Udist(M, G_6) = 0.35 \quad (6)$$

In both Udist and Mdist measures, The Ranking of nearest neighbours of M are G_6, G_4, G_5.

2.2 Task2.b

The maximum common subgraph G for all class M molecules shows in Fig 5. Then we can compute the confidence of graph class rules $G \rightarrow M$ and $\neg G \rightarrow \neg M$. Rule condition G means that subgraph G occurs in the graph and $\neg G$ means that it doesn't. we compute confidence as follows:

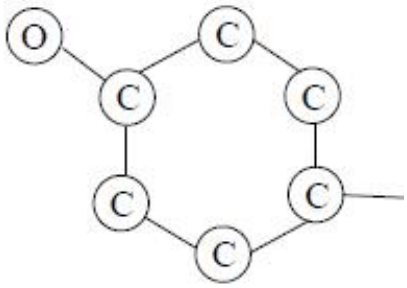

Figure 5: Maximum common subgraph G for all class M

$$Conf(G \rightarrow M) = \frac{Supp(G \cup M)}{Supp(G)} = \frac{3}{4} \tag{7}$$

$$Conf(\neg G \rightarrow \neg M) = \frac{Supp(\neg G \cup \neg M)}{Supp(\neg G)} = \frac{2}{2} = 1 \tag{8}$$

According to the current confidences, we can say if there are G in a molecule, this molecules most likely belongs to class M (monoamines). If there are no G in a molecule, this molecules won't belongs to class M (monoamines).

2.3 Task2.c

We would like to find most significant statistical associations between subgraphs and attributes in a database of compounds. Then we use the GraphApriori algorithm to find the frequent subgraphs in the database. Then we choose subgraphs that don't overlap too much, aiming to find more diverse subgraphs as features. We create new features based on remaining filtered subgraphs. So these compounds are represented by us in the form of different features as vectors. A compound may have more than one subgraph, i.e. more than one feature. Then We are able to use the previous text-based classification method to compute clustering on the data represented by frequent subgraphs features.

But it is a challenge to select frequent subgraphs with diversity. Only the richer and more different subgraphs are selected, the better results can be obtained in the subsequent feature expression. And because this is a graph-based method, it can only be implemented in small-sized graph databases, which consumes too much computing power and time when the number of compounds is too large or the structure is too complex.