Mushroom clustering

The Mushroom Dataset contains descriptions of mushroom samples 8,124 samples taken from 23 species in the Agaricus and Lepiota Family[1]. For each sample, there is a label indicating whether the sample is edible or poisonous, and a feature vector detailing a number of its characteristics. In this exercise, we will build an algorithm that separates the poisonous samples from the edible ones using only their feature vectors.

1. Reading and cleaning up the data

Download the Mushroom Dataset from the UCI website and load the '.data' file into MATLAB after changing the extension to '.txt'.

Matlab tip: to create a table from the comma delimited text file that does not contain variable names as column headings type:

```
data = readtable('agaricus-lepiota.txt','ReadVariableNames',false);
```

Examine the data and look for any entries containing a '?', which indicates missing data. Keep only the columns with no missing data, and create a new table for the labels and another one for the feature vectors. Then, convert the contents of each table to categorical variables.

Matlab tip: to convert the table to categorical data type:

```
cats = categorical(data:,:)
```

2. Feature vector encoding

In each vector, the features are encoded as categorical variables, i.e., they can only take a finite set of values with no particular ordering. Moreover, the variables take alphabetical values, which limits the range of operations that can be performed. Convert the entries of the feature vectors to numerical values using ordinal encoding.

3. Forming a graph

Build a weighted graph where each feature vector is a node, and the edge weights denote the similarity between vectors. To calculate the weights, use a function such as the Hamming or Euclidean distance.
Matlab tip: lookfor `pdist2.m` function

4. Spectral clustering

Spectral clustering embeds the graph nodes into a low-dimensional vector space where data points are organized according to the graph weights. Hence, if two nodes have a large weight on their connecting edge, they will be close in the space embedding. Then, k-means is performed on the embedded vectors in order to divide them into clusters. Use the normalized graph Laplacian to obtain the embedding, and classify the samples into two clusters with k-means (use `kmeans.m` function)

5. Performance evaluation

Compare the result of the spectral clustering to the ground truth. How many samples are labelled correctly and incorrectly for each class? Build the confusion matrix and calculate the failure probabilities.

6. Visual inspection

Plot a scatter plot of the feature vectors in the embedded space, and color them according to the cluster each point belongs to. Is two the correct number of clusters? Repeat the spectral clustering with a different number.

7. Alternative embeddings

The various definitions of the Laplacian provide diverse embeddings that may result in different cluster assignments. Repeat the spectral clustering with the unnormalized and random walk Laplacian matrices.

[1] https://archive.ics.uci.edu/ml/datasets/mushroom