Data Mining I: Homework 3
k-Nearest Neighbour and Naive Bayes

Toby Law

10 November 2019

Exercise 1.b

To find the best value of k, we can perform stratified cross-validation on the training dataset, by bootstrapping the training dataset. We can split the training dataset into multiple subsets, using one as the training set and compute the accuracy/precision/recall on the rest. We repeat this for each choice of k, and take the average of the resulting evaluation metrics obtained from all runs, to see which is the best k.

Exercise 1.c

Suppose the size of our training set is N.
k-NN is an instance-based/memory-based learning algorithm that compares new problem instances with the training instances, which have been stored in memory. The training step hence consists of the list of N training items, which makes the time complexity of the training step $O(N)$.
For the training step, we need memory blocks to remember all N points, including the values of their features/class in d dimensions, so the space complexity of the training step would be $O(Nd)$.

Exercise 1.d

The time complexity of the prediction step would not change with the number of c classes. We still are still simply looking the majority class out of the k-nearest neighbours, regardless of the number of classes the datasets are composed of.

Exercise 1.e

k-NN should work with other metrics that quantifies distances between data points on a vector space. Other metrics might predict slightly different classification results depending on the spread of the data points and how much noise

exists, but they can achieve the same purpose. However, it does not work well with semimetrics. The k-NN algorithm performs classification depending on shortest distances between test and training data points. If the metric does not satisfy the triangular inequality, it means that the straight line distance between two points is not necessarily the shortest distance, instead it could be the sum of the distances of going through intermediate training data points. Therefore, the distances obtained using semimetrics is not able to accurately reflect the distance contrast between the test data point and several training data points, and should not be used for k-NN.

Exercise 1.f

Yes, we can use k-NN for regression. Instead of our training set consisting of data points X := $\{x_1, x_2,...\}$ and their corresponding classification labels, we have a set of measurements Y := $\{y_1, y_2,...\}$. Similarly we can find for our new measurement x' its k-nearest neighbours in the training dataset, and instead of finding the majority class, the predicted measurement y' can be given by the mean of y for all those nearest neighbours. This can be expressed as the equation below:

$$y' = f(x) = \frac{1}{k}\sum_{j=1}^{k} y_{ij}$$

Exercise 2.a

We first find the probability of the data point (denoted as X) being in either class 2 and 4.

$$P(class = 2|X) = \frac{440}{664} \times 0.182 \times 0.081 \times 0.067 \times 0.97 = 6.34 \times 10^{-4}$$

$$P(class = 4|X) = \frac{224}{664} \times 0.188 \times 0.037 \times 0.105 \times 0.574 = 1.41 \times 10^{-4}$$

As $P(class = 2|X) > P(class = 4|X)$, the data point would be in class 2.

Exercise 2.b

We cannot account for the missing values when we are counting the frequency and calculating the probability that each value occurs for each feature. Therefore the resultant probabilities in Figure 4 will be smaller than when the missing values are known.

Exercise 2.c

The additive smoothing technique can be used. For features/values that have probability of 0 because there are no instances of it in the training data, we assign a small psuedocount $\alpha > 0$ to the number of instances for those features, and calculate the probability by:

$$\frac{x_i + \alpha}{N + \alpha d}$$

where x_i is the frequency of value i, N is the class frequency, and d is the number of possible values i (in our case is 10).

Exercise 3.a

$bowl \in \{1, 2\} \qquad brownie \in \{C, V\} \qquad P(bowl = 1) = \frac{1}{2}$

$$P(bowl = 1|brownie = V) = \frac{(P(brownie = V|bowl = 1)P(bowl = 1)}{P(brownie = V)}$$

$$P(brownie = V) = \frac{\text{number of vanilla brownies}}{\text{total number of brownies}} = \frac{30 + 20}{30 + 20 + 10 + 20} = \frac{5}{8}$$

$$P(brownie = V|bowl = 1) = \frac{\text{number of vanilla brownies in bowl 1}}{\text{number of brownies in bowl 1}} = \frac{30}{30 + 10} = \frac{3}{4}$$

$$P(bowl = 1|brownie = V) = \frac{3}{4} \times \frac{1}{2} \times \frac{8}{5} = \frac{3}{5}$$

Exercise 3.b

Task 1 and Task2

Please run the nbayes_uber.py script, which gives the following output lines:

```
Task 1:  Posterior distribution attains maximum at N = 60
Task 2:  The expected value of the posterior distribution is 333
```