Professional Certificate in Machine Learning and Artificial Intelligence

Module 8: k-Nearest Neighbours

Learning outcomes:

- Calculate distance functions for *k*-nearest neighbours methods.
- Apply normalisation methods to scale data sets.
- Predict and select the value of *k* for regression and classification problems using validation and test sets.
- Recognise how the curse of dimensionality can impact the validity of distance-based methods that use high-dimensional training data.

Distance functions

We use distance functions to define the concept of proximity in *k*-Nearest Neighbour Algorithms. A standard way of measuring distances is by using a q-norm distance, where q can be:

- **Any natural number**

$$\boldsymbol{dist_1}\left((\boldsymbol{X_{il}}, \dots, \boldsymbol{X_{ip}}), (\boldsymbol{X_1}, \dots, \boldsymbol{X_p})\right) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{p}} \left|\boldsymbol{X_{ij}} - \boldsymbol{X_j}\right|$$

- **The Euclidean or bird's eye distance**

$$\boldsymbol{dist_1}\left((\boldsymbol{X_{il}}, \dots, \boldsymbol{X_{ip}}), (\boldsymbol{X_1}, \dots, \boldsymbol{X_p})\right) = \sqrt{\sum_{\boldsymbol{j=1}}^{\boldsymbol{p}} \left(\boldsymbol{X_{ij}} - \boldsymbol{X_j}\right)^2}$$

- **The maximum distance**

$$\boldsymbol{dist_1}\left((\boldsymbol{X_{il}}, \dots, \boldsymbol{X_{ip}}), (\boldsymbol{X_1}, \dots, \boldsymbol{X_p})\right) = \mathbf{max}\left\{\left|\boldsymbol{X_{ij}} - \boldsymbol{X_j}\right| : \boldsymbol{j} = \mathbf{1}, \dots, \boldsymbol{p}\right\}$$

Scaling

Whenever we use distance functions in machine learning, we should be aware of the issue of scaling. There are two methods commonly used:

1. Min-max normalisation
2. Z-score normalisation

Min-max normalisation: Steps

1. Calculate the minimum value across all our training samples of the Jaft input variable
2. Calculate the maximum value for the Jaft field for the Jaft input variable across all our training samples.
3. For all of your training samples, replace the Jaft input variable value with the formula below.
4. Take the previous input value, subtract the minimum value across all the training samples.
5. Divide the difference by the difference of the maximum and the minimum occurrence of that input variable.
6. This makes sure that subsequently all of our training samples are going to be scaled between zero and one.

$$X_{ij}^{new} = \frac{X_{ij}^{old} - min\{X_{ij}^{old}: i = 1,\dots,n\}}{max\{X_{ij}^{old}: i = 1,\dots,n\} - min\{X_{ij}^{old}: i = 1,\dots,n\}}$$

Min-max normalisation works well if the predictor or input variable in question is uniformly distributed.

Z-score normalisation: Steps

1. Calculate the mean value of the Jaft feature as well as the standard deviation of the Jaft feature across all the training samples using the formulas for Mu J and Sigma J given below.
2. Go through all the training samples, and replace the Jaft input variable value with the difference of its previous value and the mean value Mu J, divided by the standard deviation Sigma J.

$$X_{ij}^{new} = \frac{X_{ij} - \mu_j}{\sigma_j}$$

with

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}^{\text{old}}$$

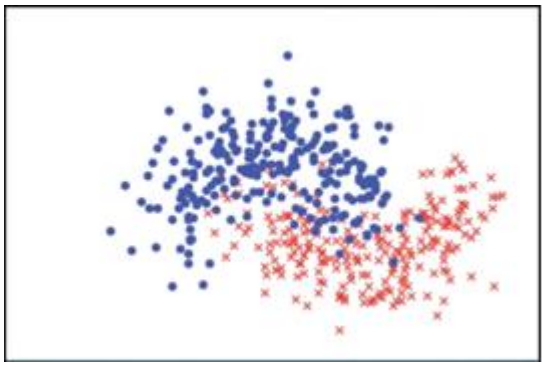
and

$$\sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (X_{ij}^{\text{old}} - \mu_j)^2}$$

Z-score normalisation, in turn, works better if we have the presence of outliers.

How to choose *k*

Consider a dataset of red and blue points with binary outcome variables, red or blue, and two numeric input variables, the positions on the *x* and *y*-axis.



For small values of *k*, the graph showing the boundary between what is predicted to be a blue sample and what is predicted to be a red sample is very wiggly.

On the other hand, for a large value of *k* such as k = 99, the boundary between what is predicted to be the red region and what is predicted to be the blue region becomes almost a line.

The optimal value of *k* lies somewhere in the middle.

Predicting the value of k using training set and validation set

In order to choose a proper value of *k*, you need to split up your data into a training data set and a validation data set. You could use 80 per cent of our data as training data and 20 per cent of your data as validation data. If you want to choose different values of k,

- Train each k-nearest neighbour method on the training data and then evaluate the misclassification rate or any other performance indicator that you might wish to use instead, on the validation data set.
- Then, choose the best model, the best value of k, based on the performance of the validation set.
- For a true, out-of-sample performance of the best model that we chose using the validation set, keep some data aside as a test set and, subsequently, measure the performance of the best choice of k on this test dataset.

The curse of dimensionality

In high dimensions even if the samples are uniformly distributed:

1. The closest data samples are far away from each other.
2. Data samples cluster at the corners of the space.

First viewpoint

Let us assume:

- We have 5000 training samples with p numerical features in [0, 1]
- The training samples are uniformly distributed in $[0, 1]^p$
- You will measure distances by the maximum norm

Question: If you have any particular point in those 5,000 training samples, how close are the five nearest neighbours to that point?

First, you need to find the smallest hypercube that covers 1/1000 of volume of the $[0, 1]^p$ hypercube.

- The $[0, 1]^p$ hypercube has volume 1
- A hypercube with side length *c* has volume c^p
- $c^p = \frac{1}{1000} \Leftrightarrow c = \left(\frac{1}{1000}\right)^{1/P}$

In one-dimensional space, to fill 50% of the 0-1 interval, we just need to go between 0 to 0.5. In two-dimensional space, to fill 50% of the 0-1 rectangle or square, we need side length of 71%.

In three-dimensional space, to fill 50% of the volume of the cube, we need side length of 80% of the overall 0-1 cube

Second viewpoint

The volume of an n dimensional ball of radius R is given by this formula:

$$\frac{\pi^{n/2}}{r\left(\frac{n}{2}+1\right)} R^n$$

Where, r is Euler's gamma function.

Question: How much of the given volume is concentrated at an epsilon slice at the surface of this *n*-dimensional ball, where epsilon is a small number?

- Calculate the volume of the entire ball

$$\frac{\pi^{n/2}}{r\left(\frac{n}{2}+1\right)} R^n$$

- Calculate the volume of the interior of that ball, where you just take an epsilon slice away around the surface.

$$\frac{\pi^{n/2}}{r\left(\frac{n}{2}+1\right)} (R-\in)^n$$

- Take the difference of those two volumes and observe what happens to the distance.

$$\Rightarrow \frac{\pi^{n/2}}{r\left(\frac{n}{2}+1\right)} [R^n - (R-\in)^n]$$

$$\rightarrow \frac{\pi^{n/2}}{r\left(\frac{n}{2}+1\right)} R^n$$

You will see that when *n* gets large, the distance converges to the volume of the ball itself.