

Task1 – K-NN classification

The Algorithm

This algorithm uses vectorization to make run time faster. This is done when calculating the distance matrix. This can be done as shown below:

Let X, Y be the test data and training data respectively where X_i is the i -th test data and Y_j is the j -th training data. The dimensions of X is $n \times d$ and the dimensions of Y are $m \times d$. Let DI be the distance matrix where DI_{ij} is the distance between the X_i and Y_j .

$$\begin{aligned} D_{ij} &= (x_i - y_j) * (x_i - y_j)^T \\ &= x_i * x_i^T - 2x_i * y_j^T + y_j * y_j^T \end{aligned}$$

Then let :

$$XX = x_i * x_i^T \quad \text{and} \quad YY = y_j * y_j^T$$

From this we know that DI is calculate by :

$$DI = (XX, \dots, XX) - 2 * X * Y^T + (YY, \dots, YY)^T$$

Where XX is repeated m times and YY is repeated n times. By calculating the distance this way, nested for loops are not necessary as vectorization speeds up the process.

The algorithm then sorts the distances and through a nested for loop takes k smallest classes and finds the mode which gives the predicted class.

Standard Output

The time taken for my_knn_system.m to run was 28.320365 seconds

k	1	3	5	10	20
N	7800	7800	7800	7800	7800
Nerrs	1184	1148	1132	1191	1306
acc	0.8482	0.8528	0.8549	0.8473	0.8326