CS4780/5780 Homework 1 Solution

Problem 1: Train/Test Splits

1. We split the training data by person - for example, we can allocate recordings from 80% of people (for all phonemes) as the training set, then 10% and 10% for the validation and test sets respectively. It is critical that one person's recordings do not appear in two different datasets, as the different datasets should be independent from one another.

2. As the system will only be used exclusively for Kilian, we can use all the original dataset as the training set along with, for example, 80% of Kilian's recordings as the training set - ideally we want roughly equal numbers of recordings for each of Kilian's phonemes. Then, the remaining 20% can be split equally between the validation and test sets.

Problem 2: K-nearest Neighbors

1. See figure 1. Blue = positive, Yellow = negative, Red = boundary line.

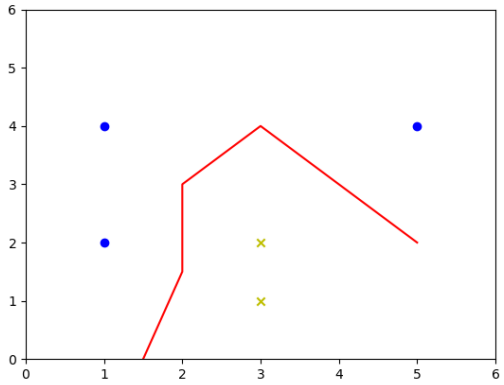

Figure 1: Decision boundary for 1-NN

2. You cannot classify (500, 1) as negative by a 1-NN classifier since it is closes to the positive point (500, 4).
   However, after we scale the data linearly to the range $[0, 1] \times [0, 1]$, namely, change the x-coordinate from x to $\frac{x-0}{500-0}$ and change the y-coordinate from y to $\frac{y-0}{5-0}$, five data points become to (0.2, 0.4), (0.2, 0.8), (1, 0.8), (0.6, 0.2) and (0.6, 0.4) and test points becomes to

(1, 0.2). (1, 0.2) is closes to a negative point (0.6, 0.2) and our 1-NN classifier can correctly classify the test point (500, 1).

3. Since we are performing 2-NN, the two closest points by Euclidean distance are (0, 0) and (1, 1), each with labels 1, 2 respectively. Therefore, since this is a regression problem, we take the average of these two labels and return 1.5 as our answer.

4. Yes, we can remove those features vectors that have missing values and use K-NN on the new dataset.

5. Unless there is serious preprocessing done on the training data that aids in distance computation, applying a k-NN classifier will take more time: training consists solely of storing the points, while applying must compute the distance between the test point and all stored training points.

6. K-NN still works on images because the underlying latent representation behind images is of low dimension - the curse of dimensionality only takes hold on data with high latent dimensionality.