STA4241 Interactive Lab Week 2: The curse of dimensionality

Today we are going to learn more about the curse of dimensionality. This is the general idea that statistical approaches begin to fail as the dimensionality of the problem grows, and that the amount of data required for approaches in higher-dimensional spaces can grow exponentially.

(1) Suppose we want to run KNN for prediction or classification. We will assume here that our features $X_1, \ldots, X_p$ all come from independent uniform distributions on the interval [0,1]

(a) Let $p = 1$, so we only have one feature, and we want to predict an outcome for a test subject. Suppose we only are willing to consider someone a neighbor for this test subject if they are within 10% of the range of X. If the test set data point has $X = 0.6$ then something can only be a neighbor if it is within [0.55, 0.65]. What fraction of the available data do we expect to be able to use as neighbors for prediction? [For simplicity, ignore cases where $X < 0.05$ or $X > 0.95$]

(b) Now assume $p = 2$. I will only use a subject as a neighbor if both predictors are within 10% of the predictors for my test data point. What percentage of the data do I expect to have available for using in prediction

(c) Do the same thing for $p > 2$.

(d) Now suppose that we want to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test point that contains, on average, 10% of the training data observations. What is the length of the sides of the hypercube for $p = 1, 2, \ldots, 100$. Note that a hypercube is a generalization of a cube to any dimension. If $p = 1$, a hypercube is a line. If $p = 2$, it is a square, etc.

(2) One problem that KNN faces as p grows is that it uses euclidean distance between the covariates to determine neighbors. Euclidean distance may not work terribly well in practice as p grows. I want you to generate covariates $\boldsymbol{X}$ from a standard normal distribution. Vary $n \in \{100, 1000\}$ and $p \in \{5, 50\}$, so you should have four data sets in total. For each pair of subjects in your data set, I want you to calculate

$$d(\boldsymbol{X}_i, \boldsymbol{X}_{i'}) = \sqrt{\sum_{j=1}^{p} (X_{ij} - X_{i'j})^2}$$

In R, you can use the following two lines of code to generate a vector of these pairwise differences

```
## get distances between all pairs of points
distMat = as.matrix(dist(x))

## extract just upper triangle entries
distances = as.vector(distMat[which(upper.tri(distMat) == TRUE)])
```

Now, calculate the ratio of the maximum and minimum distances for each data set. What do you find? Does this tell you anything about euclidean distance as the dimension grows.

(3) So what do we do in higher dimensions? Spend 5 minutes thinking about how you might improve KNN in high dimensional scenarios.