Math 345 - Notes
k-Nearest Neighbors (k-NN)
October 8, 2018

k-Nearest Neighbors

k-NN is a machine learning algorithm for supervised learning, used in classification. The goal is to classify unlabeled data using the k labeled points from the training set that are *closest* to the unlabeled point. As discussed in class, there are many ways in which we can measure *closeness*, depending on the context of the problem. We will use the *Euclidean distance* in the following examples:

$$\text{dist}\{(x_1, x_2, \ldots, x_n), (y_1, y_2, \ldots, y_n)\} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}.$$

Once we establish the k nearest neighbors, we pick the label that is **most common** among these neighbors and assign it to the unlabeled point.

Consider the following example, where customer data (age, salary) is used to label their credit (low/high) :

Age	Salary	Rating
69	3	low
66	57	low
49	79	low
49	17	low
58	26	high
44	71	high

We want to decide on the credit score for a 52 year old who makes $\$59K$ per year. We order the data points from the training set, from closest to farthest to the *unlabeled point* (we can use distance squared here):

Age	Salary	Rating	Distance^2
66	57	low	200
44	71	high	208
49	79	low	409
58	26	high	1125
49	17	low	1773
69	3	low	3425

(a) using 1-NN: the closest neighbor is $(66, 57, \text{low})$, so we label our unclassified point as **low**.

(b) using 2-NN: the closest 2 neighbors are $(66, 57, \text{low})$ and $(44, 71, \text{high})$, so we flip a coin to pick the label for our unclassified point.

(c) using 3-NN: the top 3 closest points are labeled low/high/low, so the most common label is **low**, which we assign to the unclassified point.

(d) using 4-NN: the closest 4 neighbors are labeled low/high/low/high, so we flip a coin to pick the label for our unclassified point.

(e) using 5-NN: the top 5 closest points are labeled low/high/low/high/low, so the most common label is **low**, which we assign to the unclassified point.

(f) using 6-NN: the top 6 closest points are labeled low/high/low/high/low/low, so the most common label is **low**, which we assign to the unclassified point.

Now suppose that we want to pick the best k for our data set. And suppose we have some data points we can use to test against. (For example, one can use 80% of a data set for training and 20% or testing):

Age	Salary	Rating
45	14	low
55	29	low
60	80	high

For $(45, 14, \text{low})$, the nearest neighbors (from closet to farthest):

k	Age	Salary	Rating	prediction	correct?
1	49	17	low	low	yes
2	58	26	high	high	no
3	69	3	low	low	yes
4	66	57	low	low	yes
5	44	71	high	low	yes
6	49	79	low	low	yes

For $(55, 29, \text{low})$, the nearest neighbors (from closet to farthest):

k	Age	Salary	Rating	prediction	correct?
1	58	26	high	high	no
2	49	17	low	high	no
3	69	3	low	low	yes
4	66	57	low	low	yes
5	44	71	high	low	yes
6	49	79	low	low	yes

For $(60, 80, \text{high})$, the nearest neighbors (from closet to farthest):

k	Age	Salary	Rating	prediction	correct?
1	49	79	low	low	no
2	44	71	high	high	yes
3	66	57	low	low	no
4	58	26	high	high	yes
5	49	17	low	low	no
6	69	3	low	low	no

For each of these points we consider the labeling from k-NN and measure the accuracy:

$$\text{accuracy} = \frac{\text{data points correctly predicted}}{\text{size of testing data set}}.$$

Then for various k, the accuracy is:

k	1	2	3	4	5	6
accuracy	1/3	1/3	2/3	3/3	2/3	2/3

Therefore, the best k for this (very small set) is $k = 4$. Thus, to predict an unlabeled point, we would look at the 4 closest neighbors and label according to the majority label (picking at random to break ties).