CSE 151A - Homework 01

Due: Wednesday, April 08, 2020

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope on Wednesday at 11:59 p.m.

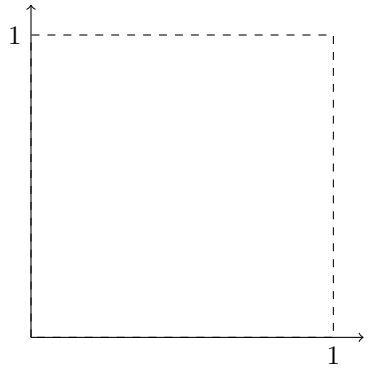
Note: The homework is substantially shorter than those that will follow, since it covers only one lecture.

Another Note: Make sure to read about the differences between essential and plus problems on cse151a.com/syllabus.html. Submit your answers to the essential problems to the Gradescope assignment named "Homework 01 - Essential Problems", and your solutions to plus problems to the Gradescope assignment named "Homework 01 - Plus Problems".

Essential Problem 1.

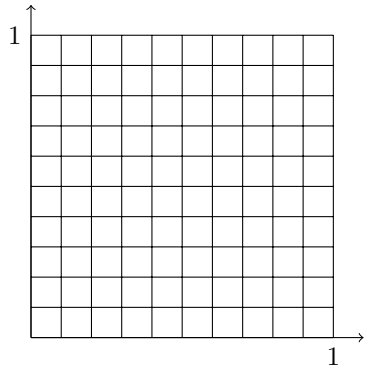
In lecture, it was mentioned that nearest neighbor classifiers struggle when the data is high dimensional due to the "curse of dimensionality". In this problem, we'll illustrate this curse in a simple way.

a) First, suppose that the data is two dimensional. Assume for simplicity that the x and y coordinate of each data point lies within the interval $[0, 1]$. That is, any data point must be in the dashed square:



In order for a nearest neighbor classifier to work well, whatever the input point is, there should be a data point nearby. In other words, we should have enough data points available so that each region of the input space has at least one data point in it.

Suppose we divide the input space into a grid by splitting each coordinate axis into ten equal bins, like this:



What is the minimum number of data points necessary so that each of the grid squares has at least one data point in it?

Edit (4/3/2020): Go ahead and assume that a point must be strictly inside a grid square. It should not lie on the border. You can assume this for the following parts, too.

> **Solution:** There are 10 bins along the first dimension and 10 bins along the second, so we need a minimum of $10 \times 10 = 100$ points.

b) Now suppose that the data is *3* dimensional, instead of two dimensional. That is, instead of being a point (x, y) in 2 dimensions, each data point is a point (x, y, z) in 3 dimensions. Assume still that each coordinate x, y, z of the data falls within $[0, 1]$.

Suppose we divide the 3-dimensional input space into a grid of cubes by splitting each coordinate axis into ten equal bins. What is the minimum number of data points necessary so that each of the grid cubes has at least one data point in it?

> **Solution:** There are 10 bins along each of the three dimensions, so we need a minimum of $10^3 = 1000$ points.

c) Now suppose that the data is *10* dimensional. That is, each data point is a point in ten dimensions., Assume still that each coordinate of the data falls within $[0, 1]$.

Suppose we divide the 10-dimensional input space into a grid of "hypercubes" by splitting each coordinate axis into ten equal bins. What is the minimum number of data points necessary so that each of the grid hypercubes has at least one data point in it?

> **Solution:** There are 10 bins along each of the 10 dimensions, so we need at least $10^{10} =$ 10,000,000,000 (ten billion) points.

Essential Problem 2.

Suppose that a particular data set has 3 possible labels with the following frequencies:

Label	Frequency
A	10%
B	75%
C	15%

a) What is the error rate of a classifier which always returns a prediction of "B"?

> **Solution:** Let A, B, C be the event that the true label is A, B, C respectively. Let $\hat{A}, \hat{B}, \hat{C}$ be the event that the classifier predicts class A, B, C respectively.
>
> We want the probability that our classifier does *not* produce the correct prediction.
>
> $$1 - P(B \cap \hat{B}) = 1 - \big(P(B) \cdot P(\hat{B})\big) = 1 - (0.75 \cdot 1) = 0.25 \text{ or } 25\%$$

b) What is the (expected) error rate of a classifier which picks a label uniformly at random from A, B, C?

Hint: https://www.loom.com/share/49e7ba62119940b0ba93aca517249d7e

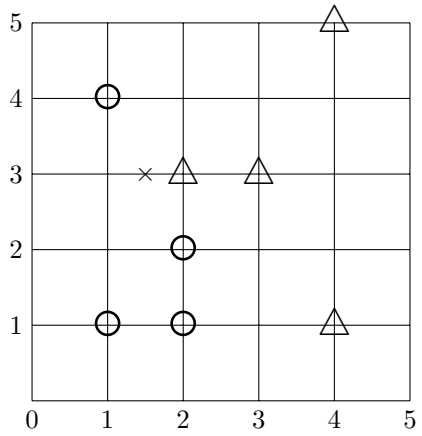
Solution: This time our classifier has an equal probability of any of our classes.

$$\begin{aligned} & 1 - P\Big((A \cap \hat{A}) \cup (B \cap \hat{B}) \cup (B \cap \hat{B})\Big) \\ &= 1 - (0.1 * \frac{1}{3} + 0.75 * \frac{1}{3} + 0.15 * \frac{1}{3}) \\ &= 1 - \frac{1}{3}(1) = \frac{2}{3} \approx 0.67 \approx 67\% \end{aligned}$$

Notice anything interesting about our result?

Essential Problem 3.

Suppose a training set of eight points is collected and shown below. Each point belongs to either one of two classes: circle or triangle.



Suppose we wish to classify a new point $(1.5, 3)$, show as the $\times$ above. We will **not** standardize the data before making our classification.

a) What will the predicted class be if we use the 1-nearest neighbor rule?

Solution: $(\triangle, 0.5)$ is the closest point, so we predict $\triangle$.

b) What will the predicted class be if we use the 3-nearest neighbor rule?

Solution: $(\triangle, 0.5), (\bigcirc, \sqrt{1.25}), (\bigcirc, \sqrt{1.25})$ are the closest three points, so we predict $\bigcirc$.

c) What will the predicted class be if we use the 5-nearest neighbor rule?

Solution: $(\triangle, 0.5), (\bigcirc, \sqrt{1.25}), (\bigcirc, \sqrt{1.25}), (\triangle, \sqrt{2.25}), (\bigcirc, \sqrt{4.25})$ are the closest five points, so we predict $\bigcirc$.

Plus Problem 1. (8 plus points)

For this problem, you'll be asked to implement the nearest neighbor rule in code. Please don't use library functions that implement the classifier itself, but it's OK to use library functions for calculating distance, sorting, etc. You may write your code in any language; please include it with your solution, (we'll look at it, but we won't run it).

The files http://cse151a.com/data/nba/train.csv and http://cse151a.com/data/nba/test.csv contain the NBA data analyzed in lecture split into a train and test set.

a) Using the 1-nearest neighbor rule, and **without** standardizing the data, what is the test error?

> **Solution:** The solution is posted as a Jupyter notebook at https://go.ucsd.edu/2UWwUVR. That link will take you to DataHub, where you'll be able to interact with the notebook in your browser.

b) Using the 1-nearest neighbor rule, and **with** standardizing the data, what is the test error?

> **Solution:** The solution is posted as a Jupyter notebook at https://go.ucsd.edu/2UWwUVR. That link will take you to DataHub, where you'll be able to interact with the notebook in your browser.