

Homework 1

Nearest neighbor classification

1. *Casting an image into vector form.* A 10×10 greyscale image is mapped to a d -dimensional vector, with one pixel per coordinate. What is d ?
2. *The length of a vector.* The Euclidean (or L_2) length of a vector $x \in \mathbb{R}^d$ is

$$\|x\| = \sqrt{\sum_{i=1}^d x_i^2},$$

where x_i is the i th coordinate of x . This is the same as the Euclidean distance between x and the origin. What is the length of the vector which has a 1 in every coordinate? Your answer may be a function of d .

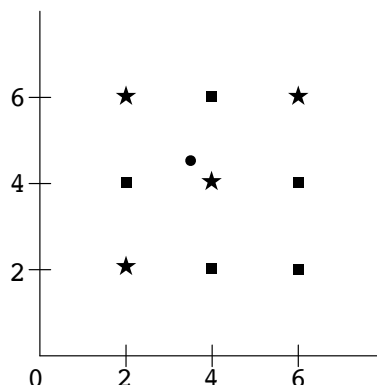
3. *Euclidean distance.* What is the Euclidean distance between the following two points in \mathbb{R}^3 ?

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

4. *Accuracy of a random classifier.* A particular data set has 4 possible labels, with the following frequencies:

Label	Frequency
A	50%
B	20%
C	20%
D	10%

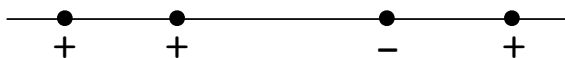
- (a) What is the error rate of a classifier that picks a label (A, B, C, D) at random, each with probability $1/4$?
 - (b) One very simple type of classifier just returns the same label, always.
 - What label should it return?
 - What will its error rate be?
5. In the picture below, there are nine training points, each with label either **square** or **star**. These will be used to guess the label of a query point at $(3.5, 4.5)$, indicated by a circle.



Suppose Euclidean distance is used.

- (a) How will the point be classified by 1-NN? The options are **square**, **star**, or **ambiguous**.
 - (b) By 3-NN?
 - (c) By 5-NN?
6. We decide to use 4-fold cross-validation to figure out the right value of k to choose when running k -nearest neighbor on a data set of size 10,000. When checking a particular value of k , we look at four different training sets. What is the size of each of these training sets?
7. An extremal type of cross-validation is *n-fold cross-validation* on a training set of size n . If we want to estimate the error of k -NN, this amounts to classifying each training point by running k -NN on the remaining $n - 1$ points, and then looking at the fraction of mistakes made. It is commonly called *leave-one-out cross-validation* (LOOCV).

Consider the following simple data set of just four points:



What is the LOOCV error for 1-NN? For 3-NN?

Programming exercises

Before attempting this problem, make sure that Python 3 and Jupyter are installed on your computer.

8. *Nearest neighbor on MNIST*. For this problem, download the archive `hw1.zip`, available from the course website, and open it. The Jupyter notebook `nn-mnist.ipynb` implements a basic 1-NN classifier for a subset of the MNIST data set. It uses a separate training and test set. Begin by going through this notebook, running each segment and taking care to understand exactly what each line is doing.

Now do the following.

- (a) For test point 100, print its image as well as the image of its nearest neighbor in the training set. Put these images in your writeup. Is this test point classified correctly?
- (b) The *confusion matrix* for the classifier is a 10×10 matrix N_{ij} with $0 \leq i, j \leq 9$, where N_{ij} is the number of test points whose true label is i but which are classified as j . Thus, if all test points are correctly classified, the off-diagonal entries of the matrix will be zero.

- Compute the matrix N for the 1-NN classifier and print it out.
 - Which digit is misclassified most often? Least often?
- (c) For each digit $0 \leq i \leq 9$: look at all training instances of image i , and compute their mean. This average is a 784-dimensional vector. Use the `show_digit` routine to print out these 10 average-digits.