Topic 4: Distances

1. We first discussed the notion of the distance between scalars (points on the number line). In this setting, consider the following three pairs of observations:

    Set 1: 10 and 20
    Set 2: 555 and 565
    Set 3: 10000 and 10010

    a. Calculate the distance between the two observations for each pair.
    b. Compare these distances.

    Suppose that the first number in each pair is the amount of money (in dollars) that you invest in the stock market and that the second number is the amount of money (also in dollars) that you have returned to you when you decide to withdraw your investment.

    c. Which set results in the greatest return relative to the amount of money you invested?
    d. The distances you calculated in part c are often called *relative distances*. Similarly, those calculated in part a are sometimes referred to as *absolute distances*. In the investment scenario, which type of distance do you think is more useful?

NOTE: No other problems will be asking about relative distance.

2. Compute **by hand** both the Euclidean and Manhattan distances between the following vectors:

    a. (1, 0) and (0, 1)
    b. (3, 4) and (6, 8)
    c. (3, 4) and (7, 4)
    d. (2.5, 3) and (2, 3)
    e. (1, 0, 1) and (2, 1, 2)
    f. (3, 5, 7) and (3, 8, 7)
    g. (1.5, 2.9, 4.3) and (0.8, 5.0, 6.7)
    h. (3, -1, 5) and (5, 3, -1)
    i. (1, 9, 4, 3) and (1, 9, 5, 4)
    j. (0, 3, 5, 7) and (1, 3, 5, 7)

Check your answers for this problem using Matlab.

3. Refer to your answers to problem 2. In what cases were the Euclidean and Manhattan distances identical? Can you see a pattern for the occurrence of this? If so, what is it and does it make sense with the formulas for the two distances?

4. Based on the code we saw this morning and that found in “distance_ex.m” on the SuMAc website, write a function to calculate the Euclidean distance between two vectors of the same arbitrary length *n*. To verify that it works, try using it to calculate some of the distances that you found in problem 2.
   HINT: The function should have two inputs and one output. The command “size” may also be of great use.

5. Repeat problem 4, but for Manhattan distance instead of Euclidean distance.

6. In the file “distance_ex.m”, find the code with heading “Calculate distance between grayscale images: Coins”.

   a. Try changing the number in “Ino=imnoise(I,'gaussian',0.1);” to different values between 0 and 1. What affect does changing this have on the two types of distances between the images?
   b. Uncomment the line containing “Ino=imnoise(I,'salt & pepper',0.1)” to change the type of noise that is added. Repeat what you did in part a for this type of noise.
   c. Which type of noise seems to have the greatest effect on the distances? Why do you think this is the case?

7. Repeat all parts of problem 6, but with the section marked “Calculate distance between RGB images: Space”.

8. Repeat problem 7, but first convert image I1 to grayscale using the command “rgb2gray”. Do the two types of noise have the same type of effects on the distances as you saw in problem 7?

9. Based on the code we saw this morning and that found in “distance_ex.m” on the SuMAc website, write a function to calculate the Euclidean distance between two **grayscale** images of the same arbitrary dimensions *m* by *k*. To verify that it works, try calculating some of the distances that you found in problem 6.

10. Repeat problem 9, but for the Manhattan distance instead of the Euclidean distance.

11. Based on the code we saw this morning and that found in “distance_ex.m” on the SuMAc website, write a function to calculate the Euclidean distance between two **RGB** images of the same arbitrary dimensions *m* by *k* by 3. To verify that it works, try calculating some of the distances that you found in problem 7.

12. Repeat problem 11, but for the Manhattan distance instead of the Euclidean distance.

13. Refer again to the code with heading "Calculate distance between grayscale images: Coins".

    a. Find the edges of image I using both the Canny and Sobel edge detectors. Compute the Euclidean and Manhattan distances between these two images.
    b. As in this original section of code, add Gaussian noise to I. Then find the edges using each edge detector. Finally, calculate the Euclidean distance between the two edge images.
    c. Repeat part b for the Manhattan distance between the two edge images.
    d. Which type of distance is most heavily affected by the noise?
    e. Repeat parts b, c, and d with Salt & Pepper noise instead of Gaussian noise.

14. Refer to the code with heading "Use image histograms to compute distances between images". Repeat problem 6, but, this time, calculate the distances between the image histograms rather than the images, themselves.

15. In order to calculate the distance between two images directly, both images must be of the same dimensions. For example, if one image is 100 by 200, then we can only compute distances to other images that are also 100 by 200. However, if we compute distances between image histograms, we can work with images of differing sizes.

    a. Read in the image "coins.png" just like we have often done before. Call it I1. Now, define a second image using the following command: "I2=imresize(I1,0.7)". This command will create a copy of I1 that is 70% the size of the original image.
    b. Save the counts for the image histogram for each image. Use these counts to calculate the Euclidean distance between the histograms.
    c. Now, plot the image histograms for both images. What is the same about the two histograms? What is different? In light of this, are you surprised by your answer to part b?
    d. To account for the differences in size of the images, we can divide the counts from the histograms by the number of pixels in the image. This will make the count vectors be **relative frequencies** rather than frequencies. Use the "size" command to calculate the number of pixels in each image and divide the histogram counts from part b by these numbers. Then calculate the Euclidean distance between the histograms using these new vectors.
    e. Which distance do you think is more meaningful: the one we got in part b or the one from part d?

16. Read in the following three images: “tire.tif”, “liftingbody.png”, and “moon.tif”.

    a. Calculate relative frequency image histograms for these three images as in part d of problem 15.
    b. Calculate the Euclidean distances between image 1 and image 2, image 1 and image 3, and image 2 and image 3. Which images are the closest?
    c. Repeat part b for the Manhattan distance.
    d. Do the two types of distances agree on which images are closest to each other? Plot the image histograms of each. Do you think your answer to this question makes sense with what you see in the histograms? In other words, which type of distance do you think is most meaningful to use here?