Problem Set 6
CMSC 422
Assigned October 23, 2019
Due Monday, November 4, 2019 at 2pm

E-M

The purpose of this problem is to implement the Expectation-Maximization algorithm for the problem of grouping points into lines. That is, we assume that we are given a set of points, and we want to find two lines that explain them. In the expectation step, we find the line that minimizes the weighted sum of squares distance from points to lines. The variance is estimated using the distance between each line and the points. In the maximization step we assign (probabilistically) each point to each line based on its distance to the lines. You can base your implementation on Yair Weiss' notes (Only Sections 1 and 2 are needed to implement this problem set. But the rest is interesting):

http://www.cs.huji.ac.il/~yweiss/emTutorial.pdf

1. **Line Fitting (20 points)** Write a function that fits a line to data. Note that Weiss describes a method for doing this using weighted least squares, which essentially only looks at error in the y direction. This fits the examples below, in which noise is added to the y coordinate. If you are interested, you can also implement, for a small amount of extra credit, a total least squares method, that takes account of the Euclidean distance between each point and the line, and see what difference this makes. You will have to do a little research to see how this works. Test your function with the following set of points (I'm using Matlab notation, in which 'a:b:c' means to select numbers from a to c, in increments of b. So 1:2:9 creates a vector of numbers, [1,3,5,7,9]. randn generates a random variable with Gaussian distribution, zero mean and variance of 1). randn(size(x)) creates a vector of such numbers that is the same size as x. Operations like abs(x), 2*x, -x, apply to every element in x. So –x creates a new vector that is the same size as x, and contains every element of x made negative.

   (i) x=0:0.05:1; y=2*x+1
   (ii) x=0:0.05:1; y=2*x+1+0.1*randn(size(x)).
   (iii) x=0:0.05:1; y=(abs(x-0.5) < 0.25).*(x+1)+(abs(x-0.5) >=0.25).*(-x);

   In all cases plot the data and the best fitting lines.

2. **E-M (80 points)** Write a function that estimates the parameters of two lines using E-M. It should get as input vectors x,y and return (a1,b1,c1), (a2,b2,c2) the parameters of the two lines as well as the weight vectors w_1 and w_2. (Set the free parameter in Eq. (2) and (3), sigma^2=0.1.) You must figure out how to initialize E-M appropriately, and how to set other parameters, if any. You should be able to manage things to that the algorithm converges to a reasonable answer.
    a. Test your function on the data in part (iii) of the previous question. Plot the data and the two fitted lines as estimated after each of the first five iterations. Also, show in separate plots the membership vectors after every iteration.
    b. Experiment with adding Gaussian noise to the y coordinates. How much noise can you add before the algorithm breaks? Describe your experiment and illustrate with appropriate plot(s).