Problem Set 2: E-M

The purpose of this problem set 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:

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

1. Line Fitting

   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.

   Test your function with the following sets of points (I'm using Matlab notation, in which "a:b:c" means to select numbers from a to c, in increments of b, and randn generates a random variable with Gaussian distribution, zero mean and variance of 1):

   (a) x=0:0.05:1; y=2*x+1.

   (b) x=0:0.05:1; y=2*x+1+0.1*randn(size(x)).

   (c) `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. Write a function that estimates the parameters of two lines using EM.

   (a) test your function on the data in part (c) 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?

3. Look into methods for choosing the number of clusters in EM (see, for example, Forsyth and Ponce). Describe and implement a method for choosing the number of lines E-M uses to fit to the data. Test this by generating random points selected from lines, varying the number of lines. How many different lines can you successfully separate using E-M?