Homework 5.1 (70 points)
CS 6375: Machine Learning

Fall 2013

Due date: Wednesday, December 4, 2013

1 K-means clustering on images [30 points]

In this problem, you will use K-means clustering for image compression. We have provided you with two images.

- Display the images after data compression using K-means clustering for different values of K (2, 5, 10, 15, 20).
- What are the compression ratios for different values of K? Note that you have to repeat the experiment multiple times with different initializations and report the average as well as the standard deviation for the compression ratio across various runs.
- Is there a tradeoff between image quality and degree of compression. What would be a good value of K for each of the two images?

We have provided you java template KMeans.java that implements various image input/output operations. You have to implement the function kmeans in the template. See the file for more details.

What to turn in for this question:

- Your source code for the kmeans algorithm.
- A report containing your write up and plots.

Note that your program must compile and we should be able to replicate your results. Otherwise no credit will be given.

2 EM algorithm [40 points]

- Download the data from the class website.
- Implement the EM algorithm for general Gaussian mixture models (assume that the data is an array of long doubles). Use the algorithm to cluster the given data (remember data is 1-D as we discussed in class). I recommend that you run the algorithm multiple times from a number of different initialization points (different θ^0 values) and pick the one that results in

the highest log-likelihood (since EM in general only finds local maxima). One heuristic is to select r different randomly-chosen initialization conditions. For example, for each start, select the initial K Gaussian means by randomly selecting K initial data points, and select the initial K covariances as all being some multiple of the overall data covariance—the selection of initial covariances is not as critical as the initial means). Another option for initialization is to randomly assign class labels to the training data points and then calculate θ^0 based on this initial random assignment (or begin the iterations by executing a single M-step, which is also fine).

Report the parameters you get for different initializations. What initialization strategy did you use? How sensitive was the performance to the initial settings of parameters.

- Now assume that variance equals 1.0 for all the three clusters and you only have to estimate the means of the three clusters using EM. Report the parameters you get for different initializations. Which approach worked better, this one or the previous one. Why? Explain your answer.

What to turn in for this part:

- Your code. EM for general GMMs and EM for GMMs with known variance.
- A report containing answers to the questions above.