Assignment 1: Density estimation and mixture models

CSC 2521, Fall 2003

Due date: October 15

This is a short written and programming assignment, designed to help make concrete the material in class. I highly recommend that you complete (or, at least, begin) everything except for HMMs before the lecture on HMMs. (That lecture will probably occur on September 29.) Contact me if you have questions. You may also wish to work out some of the problems in the text for which solutions are provided, such as Exercise 2.4 or 3.8. You may provide your solutions by handing them in or by email.

For the written problems, first write a probability model, then solve for the desired values.

1. Problem 2.37 (page 42) from MacKay. Does the result make sense intuitively?

2. Monty Hall with n doors. The basic Monty Hall problem is given in Exercise 3.8 (page 61); the solution is also provided. Now, consider the case with n doors: There are n doors, only of one of which hides a prize (with uniform prior probability). You select door 1, and the host opens all the other doors, except for door k. What is the probability that the prize is behind door k? What if $n = 1,000,000$?

3. Gaussian learning. Suppose we have a set of N scalar data values $\{x_i\}$, drawn from a Gaussian with unknown mean μ and variance σ^2. Assume that our prior over the mean is a Gaussian with known mean $\bar{\mu}$ and variance σ_μ^2. What is the MAP estimate of μ and σ^2? How does the prior influence the result as N gets larger? *Suggestion:* When estimating σ^2, it may help to substitute in $w = 1/\sigma^2$ and solve for w first.

4. Clustering. Implement k-means clustering, Mixtures-of-Gaussians (MoGs), and Hidden Markov Models. I'll provide two 2D datasets on the class webpage. Provide a visualization of the results (as shown in class). Use full covariance matrices. The choice of programming platform is up to you, although I encourage using MATLAB or Python for small prototypes such as this. I'll provide a little source code on the web as well.

For each data set and choice of k (the number of mixtures), which algorithm appears to give a better clustering of the data? Which appears to converge fastest? How do the results compare? How do the results change as you change k?

A few debugging tips:

- You should be able to reuse much of the MoG code for the HMM.
- Visualize the clustering/Gaussians/labeling after every iteration.

- MATLAB returns eigenvalues in *increasing* order.
- For debugging, try simpler cases, e.g. force $\pi_i = 1/N$ in the MoGs.
- For k-means, print out the objective function after each step, to make sure that it is going down with each iteration. If not there, is a bug with the step that increased the energy. You can assess convergence by detecting when the improvements to the energy function are tiny.
- For MoGs, print out the variational free energy after each step, to make sure that it is going down with each iteration. Note, also, that after the E-step, the variational free energy should be identical to the negative log-probability. A similar strategy can also be applied to debugging HMMs, although it requires working out the free energy and/or the log-probability for the HMM.
- One way to initialize a model is to start with a random occupancy matrix, run an M-step, and then add random perturbations to the means.
- Try clustering with two clusters on the following 2D data set: $(0,0), (1,0), (0,1), (1,1), (3,3), (3,4), (4,3), (4,4)$. With suitable initialization, you should get one cluster on the first set of points, and one cluster on the second set.