COMS 6998-01 Advanced Machine Learning
Assignment 2
February 19, 2002
Prof. Tony Jebara

The assignment due date has been extended to give students who aren't familiar with Matlab more time to catch up. Extra details in the assignment are shown in italic.
The assignment is now due on **March 12th 2002, before 2pm** either in my office CEPSR 605 or via email to jebara@cs.columbia.edu. If you email me the assignment, please use standard formats, i.e. send me plain text, latex, postscript, pdf or word.

1. **The EM Algorithm for Gaussian Mixtures**

   Write (in Matlab or some other programming language if you prefer) an implementation of the Expectation-Maximization algorithm for Gaussian mixture models for clustering D-dimensional vector data using a mixture of M multivariate Gaussian models (with variable covariance). Have the algorithm initialize the Gaussian mixture model randomly and then iterate to a (local) maximum likelihood solution with the EM steps we described in class. Compute the log-likelihood at each iteration and plot it to verify that it increased monotonically.

   The details of the EM algorithm for mixtures of Gaussians was covered in the class notes and is also available in the text Chapter. 9, pages 2-4 and pages 8-9.

   The mixture model should have the following form:

$$p(x|\Theta) = \sum_{m=1}^{M} \alpha_m \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_m|}} \, e^{-\frac{1}{2}(x-\mu_m)^T \Sigma_m^{-1}(x-\mu_m)}$$

   The free parameters to estimate are the mixing proportions $\alpha_1, \ldots, \alpha_M$ (which are all positive scalars that sum to unity), the Gaussian mean vectors $\mu_1, \ldots, \mu_M$, and the Gaussian covariances $\Sigma_1, \ldots, \Sigma_M$ (which are positive semi-definite and symmetric).

   To test out your code, try clustering dataset1 and dataset2 on the web page. Use 3 Gaussians for the algorithm.

   Use the function **randInit.m** (or write your own) to initialize the parameters randomly. The function is available on the Matlab page at the bottom. Use the functions **plotClust.m** and **plotGauss.m** (or your own plotting function if not using Matlab) to display the Gaussians overlayed on a plot of the first two dimensions of the data sets after EM converges. Also, show the log-likelihood as a plot from the initial (random) configuration to the converged solution.

   To make sure that the Gaussians don't get numerically unstable covariance matrices, make sure to add a small constant to the diagonals of the covariance matrix that EM generates after the maximization step.

   You should hand in the plots of the log-likelihood from random initialization to convergence for each data set as well as a plot of the Gaussians as they are fit to the 2D data. In addition, hand in in text form, the parameters of the mu, covar and mix datastructures for dataset1 and dataset2 that you estimated with EM (from your best run). Also, please submit your EM code and supporting functions as plain text via email or printed in the hardcopy (don't include the functions that have been provided for you on the web page). Finally you should write a brief blurb about some of the peculiarities you noted with EM, numerical issues, convergence issues, etc. To make print out your plots and images, use the print command. Type 'help print' for details. For example, 'print -depsc filename.eps' prints the current figure as an encapuslated postscript file.

2. **Cross-Validation**

Cross-validation involves splitting a data set into two where one piece of it is used to train up a model (for example using maximum likelihood) and the other piece of the data (the control set) is used to test or evaluate that model. That way, you can try different models and see which one is best (for example different numbers of Gaussians in a mixture model).

Download dataset3 (training data) and dataset4 (testing data) from the web page. Run the EM code on data set 3 until it converges for multiple values of $M = 1 \ldots 5$ (**only do 1 to 5 Gaussians, no need to go up to 10 like the original assignment stated**) and repeat each run multiple times (remember each EM run will have a different random configuration and end up in a possibly different local maximum of likelihood). Record the log-likelihood on the training and testing data sets and show it in a table or plot. Using the multiple runs, guess or estimate what is the best value of M for this data set and discuss the issues of overfitting and underfitting and why you came to that conclusion.

You should hand in the plots of a single log-likelihood converged from random initialization to convergence for dataset3 as well as a plot of the converged log-likelihoods on dataset3 (training) and dataset4 (testing) for various numbers of Gaussians (1 to 5 Gaussians in the mixture) and various random trials. Then, for the optimal number of Gaussians and the best EM run (i.e. the one that gives the highest log-likelihood on dataset4 after training on dataset3), you should show the means of the Gaussians that are produced by printing them out with the imageData function. You should write a brief blurb commenting about your results, what seems to be happening with EM, numerical issues, convergence issues, etc. on this 64-dimensional data set.