STA 414/2104, Spring 2012 — Assignment #4

Due at the start of class on April 5. Please hand it in on 8 1/2 by 11 inch paper, stapled in the upper left, with no other packaging.

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. In particular, you should not leave any discussion with someone else with any written notes (either on paper or in electronic form).

In this assignment, you will implement a Gaussian mixture model with diagonal covariance matrix, estimate its parameters by maximum penalized likelihood using the EM algorithm, and choose the number of mixture components and the penalty magnitude using a validation set. You will also try out the model on datasets that I will provide on the course web page, and discuss the results.

The mixture model is for observed data items $x_1, \ldots, x_n$, each of which is a vector of dimension p. The data items are assumed to be independent, with each having the following density function:

$$P(x) \;=\; \sum_{k=1}^{K} \pi_k \, N(x|\mu_k, \Sigma_k)$$

where K is the number of mixture components, $\pi_1, \ldots, \pi_K$ are the probabilties of these components (non-negative, summing to one), μ_k and Σ_k are the mean vector and covariance matrix for component k, and $N(x|.,.)$ denotes a multivariate normal density function. We will assume that Σ_k is diagonal, with diagonal elements $\sigma_{k1}^2, \ldots, \sigma_{kp}^2$.

You will estimate the parameters (π, μ, and σ) by maximizing the log likelihood minus the following penalty:

$$\lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_{kj} - \bar{x}_j| \,/\, s_j$$

where $\bar{x}_j$ is the sample mean of variable j over the whole estimation set, s_j is the sample standard deviation of variable j over the whole estimation set, and λ is the magnitude of the penalty, which you will determine using a validation set separate from the estimation set.

For a given value of λ, you should find the maximum penalized likelihood estimates using the EM algorithm. The course web page has a simple demonstration function that does EM for a one-dimensional mixture model, with no penalty, which you may use as a starting point if you wish. Your function should take as arguments the matrix of data values, the number of mixture components to use, the number of iterations of EM to do, the value of λ to use, and a matrix of initial "responsibilities" of components for data items. This function should return a list with elements pi, mu, sigma, and r containing the parameter estimates and resposibilities from the last iteration.

The EM algorithm for maximum likelihood estimation can be adapted to maximize the log likelihood minus a penalty by simply including the penalty when doing the maximization in the M step. Maximization with respect to each μ_{kj} parameter can be done separately (see the week 10 lecture notes), with the relevant terms of the log likelihood minus penalty being

$$-\sum_{i=1}^{n} r_{ik}(x_{ij} - \mu_{kj})^2 \,/\, 2\sigma_{kj}^2 \;-\; \lambda |\mu_{kj} - \bar{x}_j| \,/\, s_j$$

The μ_{kj} maximizing this will either be a value of μ_{kj} greater than $\bar{x}_j$ that maximizes this

expression with $|\mu_{kj} - \bar{x}_j|$ replaced by $(\mu_{kj} - \bar{x}_j)$, or a value of μ_{kj} less than $\bar{x}_j$ that maximizes this expression with $|\mu_{kj} - \bar{x}_j|$ replaced by $-(\mu_{kj} - \bar{x}_j)$, or if neither of these maxima are on the right side of $\bar{x}_j$, then the value $\bar{x}_j$. You will need to derive the formulas for finding the maximum of the quadratic expressions for μ_{kj} that are found by replacing $|\mu_{kj} - \bar{x}_j|$ with $(\mu_{kj} - \bar{x}_j)$ or $-(\mu_{kj} - \bar{x}_j)$ in the expression above.

You may find that you need to run EM with no penalty (ie, $\lambda = 0$) for some number of iterations, and then use the responsibilities from the last iteration of that run to start a run of EM with a positive λ. If you start with random responsibilities and $\lambda > 0$, you may find that EM ends up in a local maximum in which the penalty is small but the fit to the data is poor. As discussed in class, you also have to be careful that EM didn't find a silly global maximum in which one component fits a single data item with infinite probability density.

You should explicitly set the random number seed, with the `set.seed` function, before a run that will set the initial responsibilities randomly. If you don't set the seed, it will be impossible to reproduce the same results when you're trying to debug your program.

After each M step of EM, you should print the log likelihood and the log likelihood minus the penalty (which should never go down from one iteration to the next). You should compute the log likelihood in a way that avoids overflow or underflow even if the component densities overflow or underflow. This means you can't just compute the density of a data item under each mixture component, add these densities times the corresponding π_k together, and then take the log. Instead, you need to find the log of the mixture density without ever explicitly representing the individual component densities, since they might overflow or underflow. This can be done as illustrated below:

$$\log\Big(\sum_h \exp(a_h)\Big) \;\; = \;\; m \;+\; \log\Big(\sum_h \exp(a_h - m)\Big), \quad \text{where } m = \max_h a_h$$

Because of the subtraction of m, the exponentials on the right will never overflow. One of these exponentials will be $\exp(0) = 1$, so although some other of them may underflow to zero, those will negligible anyway, so any such underflows won't produce a large error.

I will provide three data sets on the web page for you to try out your program on. For each data set, I will provide an estimation set for you to run the EM algorithm on, a validation set for you to use to choose K and λ, and a test set for you to use to evaluate the final performance of the method. (I've divided the full training set into estimation and validation sets myself in order to ensure consistency between students — in a real application, you would need to randomly divide the training set yourself.) You should use these data sets in their original form, **not** standardizing the variables to have mean zero and variance one.

You should evaluate values of K and λ by the average log probability of the validation items. You should take the precautions needed to avoid overflow and underflow described above when computing this. Similarly, you should use the average log probabiity of the test items to evaluate how well the whole method worked.

You should hand in your function implementing EM for this problem, the R scripts that you used to select K and λ for each data set, and the output of your runs (including the final estimates and the values of the log likelihood and log likelihood minus penalty from the EM run that produced the final estimates). Your scripts needn't be fully automatic — for example, you can manually select the number of iterations needed based on preliminary runs (you only need to hand in the output of the final run). Finally, you should discuss the results — for instance, how fast or slow EM was, how the penalty changed the estimates, and whether use of the penalty resulted in improved performance on test cases.