STA 414/2104, Spring 2013 — Assignment #3

Due at the start of class on March 21. 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 write an R function for fitting a mixture of Gaussians to multivariate data by maximum penalized likelihood, with the penalty on the means of the Gaussians in the mixture being derived from a Gaussian process model that is designed to produce a set of components that capture one-dimensional structure in the data.

The data that will be modeled consists of n independent items, for each of which there are measurements on p variables. You should model this as a Gaussian mixture with K components, in which the probability density of item x_i is given by

$$P(x_i) \;=\; \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} N(x_{ij} \,|\, \mu_{kj}, \sigma^2)$$

where $N(x \,|\, \mu, \sigma^2)$ is the normal probability density function with mean μ and variance σ^2. The parameters of this model are the mixing proportions, $\pi_1, \ldots, \pi_K$, the means of the components, $\mu_1, \ldots, \mu_K$, each of which is a p-dimensional vector, and a single standard deviation parameter, σ, that is used for all components and all variables.

Your R function should try to find the parameter values that maximize the log likelihood plus a "penalty" that is equal to the log of the density of $(\mu_1, \ldots, \mu_K)$ under a Gaussian process model described below. (Note that with this "penalty", higher values are better.) You can do this using the EM algorithm, as also described below. You may write your R function starting with the example EM function on the course web page (although since it would need considerable modifications, you might instead decide to just start anew).

The "penalty" to use is the log of the following Gaussian density for $(\mu_1, \ldots, \mu_K)$:

$$P(\mu_1, \ldots, \mu_K) \;=\; \prod_{j=1}^{p} N((\mu_{1j}, \ldots, \mu_{Kj}) \,|\, 0, C)$$

where C is a covariance matrix for the means of a variable j in each of the K mixture components (the same for all variables), which you should define as follows:

$$C_{k,k'} \;=\; \mathrm{Cov}(\mu_{kj}, \mu_{k'j}) \;=\; 2^2 \;+\; 2^2 \exp\left(-\left(\frac{k - k'}{Ks} \right)^2 \right) \;+\; 0.001^2 \delta_{kk'}$$

Here, s is a scale factor, which you will need to manually tune for each dataset. The final term above (which adds 0.001^2 to the diagonal of C) has negligible statistical effect, but helps avoid numerical problems with matrices that are almost singular. The effect of adding the log of the density above to the log likelihood is to, for each variable, j, encourage the means $\mu_{1j}, \ldots, \mu_{Kj}$ to vary smoothly with k (more smoothly when s is larger).

The EM algorithm can easily be adapted to find maximum penalized likelihood estimates rather than maximum likelihood estimates — referring to the general version of the algorithm that is presented in the lecture slides, the E step remains the same, but the M step will now maximize $E_Q[\log P(x, z|\theta) + G(\theta)]$, where $G(\theta)$ is the "penalty".

The EM algorithm can also be generalized so that the M step does not need to find the actual maximum, as long as it improves the objective. In particular, if there are two sets of parameters, θ_1 and θ_2, the M step can first maximize with respect to θ_1 with θ_2 fixed, and then maximize with respect to θ_2 with θ_1 fixed.

For this assignment, the M step should consist of three sub-steps: estimation of π, estimation of μ (based on the previous estimate for σ), and estimation of σ (based on the μ just estimated). You should work out yourself how π and σ should be estimated in the M step. The μ parameters can be estimated, separately for each variable, as follows:

$$(\hat{\mu}_{1j}, \ldots, \hat{\mu}_{Kj}) \;=\; \Big[C^{-1} + D\Big]^{-1} \begin{bmatrix} \sum_i x_{ij} r_{i1}/\sigma^2 \\ \vdots \\ \sum_i x_{ij} r_{iK}/\sigma^2 \end{bmatrix} \tag{1}$$

where D is a diagonal matrix with diagonal entries $\sum_i r_{i1}/\sigma^2, \ldots, \sum_i r_{iK}/\sigma^2$.

Your R function for running EM on this problem should take the following arguments: an $n \times p$ data matrix, the number of mixture components, K, a value for the scale factor, s, in the covariance function, an initial value for σ, and the number of iterations of EM to do. This function should start by randomly setting the matrix of "responsibilities" (as in the example program on the course web page), then alternate M and E steps for as many iterations as requested. It should return a list with elements giving the final estimates for π, μ, and σ. You may add an extra argument to control whether any trace output that you find useful is enabled.

You should try out this function on three datasets provided on the course web page. Each dataset consists of 1000 items (one per line), of which you should use only the first 200 for training the model. The number of variables varies. You should read the data with read.table with HEAD=FALSE, and then convert it to a matrix with as.matrix (for speed of access).

For each dataset, you should try various values of K and s (values of $K = 10$ and $s = 1$ may be reasonable starting points), looking for a choice for which the model fits the data in a sensible way. One way to judge this is by plotting the data (two variables at a time) along with the locations of the component means, to see whether they have become ordered in a way that corresponds to the low-dimensional structure in the data. You should also see how successfully the model predicts the 800 data items not used for fitting, using the final parameter estimates, judging performance by the average log probability density over these 800 items. Note that we're just trying to see whether the model will do the sort of thing we're hoping it will, not to judge precisely how well it works, so we won't worry about selecting K and s on the basis of the "test data". (We'd need to use cross-validation in a serious assessment of how well the model works.)

You will need to do multple runs of EM with different random initializations of the responsibilities, since EM may sometimes converge to a bad local maximum. You will also need to judge how many iterations are necessary (100 or more may be needed).

Setting s to a very small value (eg, 0.001) will produce close to maximum likelihood estimates (with no penalty), against which you can compare results with larger values of s.

You should hand in a listing of the R function you wrote for fitting by EM, the R scripts you used to apply this function to the three datasets, the final parameter estimates you found, and a discussion of the results. For your discussion, you may want to output more information, or produce various plots, in order to gain as much insight as you can into how well this combination of Gaussian mixture modelling with a Gaussian process penalty works.

Bonus (10 marks): Derive the formula (1) above for estimation of μ.