STA 414/2104, Spring 2011 — Assignment #3

Due at the start of class on April 7. 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 paper or electronic).

In this assignment, you will implement Gaussian mixture models trained using the EM algorithm, and use them to fill in missing covariate values for linear regression.

Regression models model only the distribution of the response given the values of the covariates, not the distribution of the covariates themselves. Consequently, regresssion models cannot be applied when one or more covariate values are missing - the function for computing the expected value of the response given the covariates just can't be evaluated. One way to try to handle this is to fill in the missing values (also called "imputing" them), after which the regression model can be applied. For example, one simple way is to replace every value for covariate j that is missing by the sample mean of covariate j in the training cases where it is not missing. One would hope that some more sophisticated method will do better, however.

One should note that all methods for filling in missing covariates assume that whether a covariate is missing or not in some case is unrelated to the value of the response variable for that case, given the values of the non-missing covariates. If this is not true, it will be very difficult to make any valid inferences from the data. We'll assume here that whether a covariate is missing or not is unrelated to *any* of the variables in the model (they are "missing completely at random"). For example, the covariates might be responses to a survey, for which the participants answered all questions, but for which some answers were lost when the wind blew away some of the papers the answers were on. (Unfortunately, this is usually not a good model of how data comes to be missing, but we'll assume it here.)

With this assumption, we can try to filling missing values for covariates by modelling the joint distribution of all covariates, and from this finding the conditional distribution for the missing covariate values in a case given the values that are not missing. For this assignment, we'll then fill in the missing values with the mean of this conditional distribution. (This throws away information about variation, however, which could be retained by filling in the values in more than one way — what's called "multiple imputation".)

We'll use a Gaussian mixture to model the joint distribution of the covariates, with the covariance matrix for a mixture component restricted to being diagonal. The parameters will be estimated by maximum likelihood (avoiding singular solutions), using the EM algorithm.

You can use the simple R program for the EM algorithm discussed in lectures (and found on the course web page) as a starting point for your program, but you will need to extend it significantly, to handle more than one variable, and to handle estimation when values for some variables are missing. You should also extend the program so that it prints the log likelihood after each iteration, so that you can see how the EM algorithm is progressing, and compare estimates found with different starting points. (Be sure to find the log likelihood by adding the log probability for each case, not by multiplying the probabilities for each case and then taking the log at the end, since the latter is likely to lead to numbers that overflow or underflow.) You do not need to automatically determine how many EM iterations to perform. Instead, as for the program demonstrated in class, the number

of iterations can be specified manually, though you may need to adjust this number yourself based on the results. Finally, you will need to write a function to fill in missing values using a mixture model that you have fit.

To get the probability for a case with values for some variables missing, you simply omit the factors corresponding to those variables. This is also done when finding the "responsibilities" of mixture components for training items in the E step of the algorithm. In the M step, when re-estimating the mean and standard deviation of variable j for each of the mixture components, you simply ignore training cases where variable j is missing.

Your program should be written to handle any number of variables, but we will apply it to a single dataset available from the course web page in which there are three covariates to be modelled with the mixture model. There is a file of 100 training cases, in which some covariate values are missing, and another file of 900 test cases in which no covariates are missing. You can read the data as follows:

```
data.trn <- as.matrix(read.table("a3-train-data.txt",head=FALSE))
x.trn <- data.trn[,1:3]
t.trn <- data.trn[,4]

data.tst <- as.matrix(read.table("a3-test-data.txt",head=FALSE))
x.tst <- data.tst[,1:3]
t.tst <- data.tst[,4]
```

Note that although the test cases I provided have no missing covariate values, test cases with missing values could be handled by filling in the missing values using the mixture model, in the same way as for the training cases.

To start, you should just use the R `lm` function to estimate a regression model of `t.trn` on `x.trn`. When some covariate values are missing, `lm` simply ignores all cases with any missing values, and produces estimates based on the remaining cases. You should use these estimates to predict the response in the test cases, and report the average squared error.

You should then try filling in the missing covariate values with the mean value for training cases where they are not missing. Once these values are filled in, you can use `lm` with all the training cases, and again you can see what the average squared error is when predicting for the test cases.

You should try filling in the missing covariate values using mixture models fit with 3 components and with 5 components. For each number of components, you should run the EM algorithm from several random starting points, and use the estimates found that have highest likelihood, except that any estimates corresponding to a singular solution (in which one more more standard deviations go to zero) should be ignored. Report the average squared error on test cases when using these 3-component and 5-component models.

You should hand in these results, and your R program, and a brief discussion (about one page). In your discussion you can talk about how well the EM algorithm worked, and about what explains the differences in average squared error that you saw.