Stat243: Problem Set 8, Due Wednesday Dec. 3

November 17, 2014

Comments:

- This covers Units 12 and 13.
- It's due at the start of class on Dec. 3.
- As usual, simply providing the raw code is not enough; make sure to describe how you approached the problem, the steps you took, and output illustrating what your code produces.
- Please note my comments in the syllabus about when to ask for help and about working together.
- As discussed in the syllabus, please turn in (1) a copy on paper, as this makes it easier for us to handle AND (2) an electronic copy through Git following Jarrod's instructions. For problem 3a, you can write this by hand provided it is inserted in order with the remaining material that you turn in.

Note that while the Spark question on GLM fitting is extra credit, for those of you who will soon be on the job market and think that you will be applying for jobs that involve working with big datasets, I strongly recommend you do this problem. It may be a nice thing to be able to talk about in an interview. For those of you who got a 0 or a 1 on a problem set, I'll also consider successful completion of this problem to offset a portion of that problem set.

Questions

1. Experimenting with importance sampling.

   (a) Use importance sampling to estimate the mean of a t distribution with 3 degrees of freedom, truncated such that $X < (-4)$. Have your sampling density be a normal distribution centered at -6 and then truncated so you only sample values less than -4 (this is called a half-normal distribution). You should be able to do this without discarding any samples (how?). Use $m = 10000$ samples. Create histograms of $h(x)f(x)/g(x)$ and of the weights $f(x)/g(x)$ to get a sense for whether Var($\hat{\mu}$) is large. Note if there are any extreme weights that would have a very strong influence on $\hat{\mu}$. Hint: remember that your $f(x)$ needs to be appropriately normalized or you need to adjust the weights per the class notes.

   (b) Now use importance sampling to estimate the mean of a t distribution with 3 degrees of freedom, truncated such that $X < (-6)$, but have your sampling density be a t distribution centered at -4 and truncated so you only sample values less than -4. Again you shouldn't have to discard any samples. Respond to the same questions as above in part (a).

2. Consider the "helical valley" function (see the helical.R file in the repository). Try out *optim()* and *nlm()* for finding the minimum of this function (or use *optimx()*). Explore the possibility of multiple local minima by using different starting points. Plot slices of the function to get a sense for how it behaves (i.e., for a constant value of one of the inputs, plot as a 2-d function of the other two). Syntax for *image()*, *contour()* or *persp()* from the graphics unit (Unit 15) will be helpful.

3. Consider a censored regression problem. We assume a simple linear regression model, $Y_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2)$. Suppose we have an iid sample, but that for any observation with $Y > \tau$, all we are told is that Y exceeded the threshold and not its actual value. In a given sample, c of the n observations will (in a stochastic fashion) be censored, depending on how many exceed the fixed τ.

   (a) Design an EM algorithm to estimate the 3 parameters, $\theta = (\beta_0, \beta_1, \sigma^2)$, taking the complete data to be the available data plus the actual values of the truncated observations. You'll need to make use of $E(Y|Y > \tau)$ and Var$(Y|Y > \tau)$ where Y is normally distributed. Be careful that you carefully distinguish θ from the current value at iteration t, θ_t, in writing out the expected log-likelihood and computing the expectation and that your maximization be with respect to θ. You should be able to analytically maximize the expected log likelihood. A couple hints:

      i. From the Johnson and Kotz bibles on distributions, the mean and variance of the truncated normal distribution, $f(Y) \propto \mathcal{N}(\mu, \sigma^2)I(Y > \tau)$, are:

$$\begin{aligned} E(Y|Y > \tau) &= \mu + \sigma\rho(\tau^*) \\ V(Y|Y > \tau) &= \sigma^2 \left(1 + \tau^*\rho(\tau^*) - \rho(\tau^*)^2\right) \\ \rho(\tau^*) &= \frac{\phi(\tau^*)}{1 - \Phi(\tau^*)} \\ \tau^* &= (\tau - \mu)/\sigma, \end{aligned}$$

      where $\phi(\cdot)$ is the standard normal density and $\Phi(\cdot)$ is the standard normal CDF.

      ii. You should recognize that your expected log-likelihood can be expressed as a regression of $\{Y_{obs}, m_t\}$ on $\{x\}$ where Y_{obs} are the non-censored data and $\{m_{i,t}\}$, $i = 1, \ldots, c$ are used in place of the censored observations. Note that $\{m_{i,t}\}$ will be functions of θ_t and thus constant in terms of the maximization step. Your estimator for σ^2 should involve a ratio where the numerator involves the usual sum of squares for the non-censored data plus two additional terms that you should interpret statistically.

   (b) Propose reasonable starting values for the 3 parameters as functions of the observations.

   (c) Write an R function, with auxiliary functions as needed, to estimate the parameters. Make use of the initialization from part (b). You may use *lm()* for updating β. You'll need to include criteria for deciding when to stop the optimization. Test your function using data simulated from the model with (a) a modest proportion of exceedances expected, say 20%, and (b) a high proportion, say 80%. Take $n = 100$ and the parameters such that with complete data, $\hat{\beta}_1/se(\hat{\beta}_1) \approx 3$. (In other words, you'll need to figure out values of β_1 and σ^2 such that the signal to noise ratio is 3.) You'll also need to generate the xs in some reasonable fashion.

   (d) A different approach to this problem just directly maximizes the log-likelihood of the observed data, which for the censored observations just involves the likelihood terms, $P(Y_i > \tau)$. Estimate the parameters (and standard errors) for your test cases using *optim()* with the BFGS option in R. You will want to consider reparameterization, and possibly use of the *parscale* argument. Compare how many iterations EM and BFGS take. Note that parts (c) and (d) together provide a nice test of your code.

4. (Extra credit) Write Spark code to implement IWLS for GLMs in parallel. You should be able to take the demo code from the end of Unit 9 and modify it for this purpose. Use your code to fit a logistic regression model for the binary outcome of whether a plane arrives 15 or more minutes late. You can take the covariates to be those I used in the in class demo (distance and day of week) or use other covariates you might be interested in. Your code for fitting the GLM should be general and apply to any set of covariates, but your code to create the X matrix can just deal with the specific covariates you use in the airline model you fit.