Stat243: Problem Set 8, Due Fri. November 30

November 16, 2018

This covers Units 10 and 11.
It's due **as PDF submitted to bCourses** and submitted via Github at 2 pm on Nov. 30.
Some general guidelines on how to present your problem set solutions:

1. Please use Rmd/Rtex as in previous problem sets.
2. Your solution should not just be code - you should have text describing how you approached the problem and what the various steps were.
3. Your PDF submission should be the PDF produced from your Rmd/Rtex. Your Github submission should include the Rtex/Rmd file, any R code files containing chunks that you read into your Rtex/Rmd file, and the final PDF, all named according to the guidelines in *howtos/submitting-electronically.txt*.
4. Your code should have comments indicating what each function or block of code does, and for any lines of code or code constructs that may be hard to understand, a comment indicating what that code does. You do not need to show exhaustive output but in general you should show short examples of what your code does to demonstrate its functionality.
5. Please note my comments in the syllabus about when to ask for help and about working together. In particular, **please give the names of any other students that you worked with on the problem set and indicate in comments any ideas or code you borrowed from another student.**

Problems

1. Experimenting with importance sampling.
   (a) Use importance sampling to estimate the mean (i.e., $\phi = E_f X$) of a truncated t distribution with 3 degrees of freedom, truncated such that $X < (-4)$. Have your sampling density be a normal distribution centered at -4 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 the weights $f(x)/g(x)$ to get a sense for whether Var($\hat{\phi}$) is large. Note if there are any extreme weights that would have a very strong influence on $\hat{\phi}$. Estimate Var($\hat{\phi}$). 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 the same truncated t distribution with 3 degrees of freedom, truncated such that $X < (-4)$, but have your sampling density be a t distribution, with 1 degree of freedom (not 3), 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 *ps8.R* file in the repository). 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()* (or the ggplot2 equivalents) from the R bootcamp materials will be helpful. Now 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.

3. Consider probit regression, which is an alternative to logistic regression for binary outcomes. The probit model is $Y_i \sim \text{Ber}(p_i)$ for $p_i = P(Y_i = 1) = \Phi(X_i^\top \beta)$ where Φ is the standard normal CDF. We can rewrite this model with latent variables, one latent variable, z_i, for each observation:

$$\begin{aligned} y_i &= I(z_i > 0) \\ z_i &\sim \mathcal{N}(X_i^\top \beta, 1) \end{aligned}$$

   (a) Design an EM algorithm to estimate β, taking the complete data to be $\{Y, Z\}$. You'll need to make use the the mean and variance of truncated normal distributions (see hint below). Be careful that you carefully distinguish β from the current value at iteration t, β^t, in writing out the expected log-likelihood and computing the expectation and that your maximization be with respect to β (not β^t). Also be careful that your calculations respect the fact that for each z_i you know that it is either bigger or smaller than 0 based on its y_i. 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, $f(w) \propto \mathcal{N}(w; \mu, \sigma^2) I(w > \tau)$, are:

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

      where $\phi(\cdot)$ is the standard normal density and $\Phi(\cdot)$ is the standard normal CDF. Or see the Wikipedia page on the truncated normal distribution for more general formulae.

      ii. You should recognize that your expected log-likelihood can be expressed as a regression of some new quantities (which you might denote as m_i, $i = 1, \ldots, n$, where the m_i are functions of β^t and y_i) on X.

   (b) Propose reasonable starting values for β.

   (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 the update steps. You'll need to include criteria for deciding when to stop the optimization. Test your function using data simulated from the model, with say $\beta_0, \beta_1, \beta_2, \beta_3$. Take $n = 100$ and the parameters such that $\hat{\beta}_1/se(\hat{\beta}_1) \approx 2$ and $\beta_2 = \beta_3 = 0$. (In other words, I want you to choose β_1 such that the signal to noise ratio in the relationship between x_1 and y is moderately large.)

   (d) A different approach to this problem just directly maximizes the log-likelihood of the observed data. Estimate the parameters (and standard errors) for your test cases using *optim()* with the BFGS option in R. Compare how many iterations EM and BFGS take.