Stat243: Problem Set 8, Due Dec. 4

November 17, 2015

Comments:

- This covers Units 10 and 11.
- It's due at the start of class on Friday Dec. 4.
- As usual, your solution should mix textual description of your solution, code, and example output. Feel free to write out answers to the mathematical problems by hand if you like. If you do so, please staple them into any PDF pages in the correct order to avoid having Harold have to hunt around for what problem solution is where.
- Please note my comments in the syllabus about when to ask for help and about working together.
- Please give the names of any other students that you worked with on the problem set.

Problems

1. Describe the process of carrying out a simulation study for the following scenario. You are interested in how regression methods perform when there are outlying values, in particular in comparing a new method that is supposedly robust to outliers to standard linear regression. In particular you're interested in absolute prediction error and in the coverage of prediction intervals for new observations, where the prediction intervals are based on using the nonparametric bootstrap.

   (a) Describe (in text, formulas and/or with pseudo-code as helpful) the steps involved in the simulation study and how you would estimate the prediction error and coverage and compare the two methods. Be precise in your notation about the various sample sizes involved, where and how random values are generated, and where loops are involved and the indexing of those loops. You don't need to describe in detail how the bootstrap calculations are done (you can treat it as a black box), but be clear what values are the inputs to and outputs from the bootstrap calculations and any looping involved in the bootstrapping.

   (b) Provide a concrete example using specific numerical values and a specific strategy for generating the simulated datasets.

2. Let's consider importance sampling and explore the need to have the sampling density have heavier tails than the density of interest. Assume that we want to estimate EX and $E(X^2)$ with respect to a density, f. We'll make use of the Pareto distribution, which has the pdf $p(x) = \frac{\beta\alpha^\beta}{x^{\beta+1}}$ for $\alpha < x < \infty$, $\alpha > 0$, $\beta > 0$. The mean is $\frac{\beta\alpha}{\beta-1}$ for $\beta > 1$ and non-existent for $\beta \leq 1$ and the variance is $\frac{\beta\alpha^2}{(\beta-1)^2(\beta-2)}$ for $\beta > 2$ and non-existent otherwise.

(a) Does the tail of the Pareto decay more quickly or more slowly than that of an exponential distribution?

(b) Suppose f is an exponential density with parameter value equal to 1, shifted by two to the right so that $f(x) = 0$ for $x < 2$ and our sampling density, g, is a Pareto distribution with $\alpha = 2$ and $\beta = 3$. Use $m = 10000$ to estimate EX and $E(X^2)$. Recall that Var$(\hat{\mu}) \propto$ Var$(h(X)f(X)/g(X))$. Create histograms of $h(x)f(x)/g(x)$ and of the weights $f(x)/g(x)$ to get an idea for whether Var$(\hat{\mu})$ is large. Note if there are any extreme weights that would have a very strong influence on $\hat{\mu}$.

(c) Now suppose f is the Pareto distribution described above and our sampling density, g, is the exponential described above. Respond to the same questions as for part (b).

3. Consider probit regression, which is an alternative to logistic regression for binary outcomes. The probit model is $P(Y_i = 1) = \Phi(X_i^\top \beta)$ where Φ is the standard normal CDF. We can rewrite this model with latent variables, one latent variable 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 of $E(W|W > \tau)$ and Var$(W|W > \tau)$ where W is normally distributed. 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.

4. Consider the "helical valley" function (see the *helical.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()* from the graphics unit (Unit 13) will be helpful. 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.

5. (Extra credit) Write Spark code to implement IWLS for GLMs in parallel, in particular for logistic regression. You should be able to take the demo code from the end of Unit 7 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. Given our limited credits on AWS, please do your development for this problem using only two Spark worker nodes with a subset of the airline dataset.