

Problem Set 3

Chris Conlon

Due: April 4, 2022

This problem set is designed to make sure that your `numpy` or `R` skills are up to speed. For each question, the expectation is that you complete the task, provide the appropriate code, and fully read the documentation, and work the `numpy` provided examples on your own. The tasks should be fairly easy. The goal is to accomplish the tasks in the most straightforward manner with the least amount of code.

If you are new to `numpy` I suggest the following tutorials:

- <https://docs.scipy.org/doc/numpy/user/quickstart.html>
- <https://docs.scipy.org/doc/numpy/user/numpy-for-matlab-users.html>

Part I: Logit Inclusive Value

The logit inclusive value or $IV = \log \sum_{i=0}^N \exp[x_i]$.

1. Show that the this function is everywhere convex if $x_0 = 0$.
2. A common problem in practice is that if one of the $x_i > 600$ that we have an “overflow” error on a computer. In this case $\exp[600] \approx 10^{260}$ which is too large to store with any real precision, especially if another x has a different scale (say $x_2 = 10$). A common “trick” is to subtract off $m_i = \max_i x_i$ from all x_i . Show how to implement the trick and get the correct value of IV . If you get stuck take a look at Wikipedia.
3. Compare your function to `scipy.special.logsumexp` in Python or `matrixStats.LogSumExp` in R. Does it appear to suffer from underflow/overflow? Does it use the max trick?

Part II: Numerical Integration

In this part we will look to calculation the logit choice probability $p(X, \theta)$ by numerical integration:

$$p(X, \theta) = \int_{-\infty}^{\infty} \frac{\exp(\beta_i X)}{1 + \exp(\beta_i X)} f(\beta_i | \theta) d\beta_i.$$

Assume $f \sim N(0.5, 2)$ and that $X = 0.5$.

A helpful relationship is the following:

Let $Y \sim N(\mu, \sigma^2)$ and apply COV $x = (y - \mu)/\sqrt{2}\sigma$

$$\begin{aligned} E[f(Y)] &= (2\pi\sigma^2)^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(y) \exp\left[-\frac{(y - \mu)^2}{2\sigma^2}\right] dy \\ \int_{-\infty}^{\infty} f(y) \exp\left[-\frac{(y - \mu)^2}{2\sigma^2}\right] dy &= \int_{-\infty}^{\infty} f(\sqrt{2}\sigma x + \mu) e^{-x^2} \sqrt{2}\sigma dx \end{aligned}$$

Gives the quadrature formula using Gauss Hermite (x_s, w_s) .

$$E[f(Y)] = \frac{1}{\sqrt{\pi}} \sum_{s=1}^S w_s f(\sqrt{2}\sigma x_s + \mu)$$

notice that we don't have the e^{-x^2} anymore, so we can use the (w_s, x_s) weights and nodes for Gauss-Hermite quadrature without a problem.

1. Create the function in an Python/R called `binomiallogit`. (It should take β the item you integrate over as its argument, it should take the PDF `scipy.stats.norm.pdf` in Python, or `pnorm` in R or as an optional argument).
2. Integrate the function using Python's `scipy.integrate.quad` or `gauss.quad` in R and setting the tolerance to 1×10^{-14} . Treat this as the "true" value.
3. Integrate the function by taking 20 and 400 Monte Carlo draws from f and computing the sample mean.
4. Integrate the function using Gauss-Hermite quadrature for $k = 4, 12$ (Try some odd ones too). Obtain the quadrature points and nodes from the internet. Gauss-Hermite quadrature assumes a weighting function of $\exp[-x^2]$, you will need a change of variables to integrate over a normal density. [See my notes <https://github.com/chrisconlon/Grad-IO/blob/master/Extra%20Notes/Numerical%20Integration/integration.pdf>] You also need to pay attention to the constant of integration.
5. Compare results to the Monte Carlo results. *Make sure your quadrature weights sum to 1!*
6. Repeat the exercise in two dimensions where $\mu = (0.5, 1)$, $\sigma = (2, 1)$, and $X = (0.5, 1)$.
7. Put everything into two tables (one for the 1-D integral, one for the 2-D integral). Showing the error from the "true" value and the number of points used in the evaluation.

Part III: Maximum Likelihood

I have generated data from the following utility model where student i is choosing among a set of apartments j that all have the same price.

$$u_{ij} = \beta_0 + \beta_1 \cdot \log \text{square feet}_j + \beta_2 \cdot \text{bathrooms}_j + \beta_3 \cdot \text{bathrooms}_j \cdot \text{family size}_i + \gamma_i \cdot \text{outdoor space}_j + \varepsilon_{ij}$$

The true data-generating process allows $\gamma_i \sim N(\bar{\gamma}, \sigma^2)$ so that students differ in their tastes for outdoor space.

1. Estimate a plain logit where $\gamma_i = \gamma$ for all individuals via maximum likelihood.
2. Estimate a mixed logit using Gauss-Hermite integration via maximum simulated likelihood.
3. Bonus: Compute the standard errors using the information matrix / Hessian relationship.