Homework 2

Evan Ott
UT EID: eao466

September 14, 2016

Stochastic Gradient Descent

(A)

I proved this last time (see `/hw1/hw1.pdf` on GitHub), by showing

$$\begin{aligned}
\nabla l(\beta) &= -\sum_{i=1}^{N} \left[y_i - m_i w_i(\beta)\right] x_i \\
w_i(\beta) &= \frac{1}{1+\exp\left(-x_i^\top \beta\right)} \\
\hat{y}_i &= m_i w_i(\beta) \\
\nabla l(\beta) &= -\sum_{i=1}^{N} \left[y_i - \hat{y}_i\right] x_i \\
&= \sum_{i=1}^{N} \left[\hat{y}_i - y_i\right] x_i \\
g_i(\beta) &= \left[\hat{y}_i - y_i\right] x_i \\
\nabla l(\beta) &= \sum_{i=1}^{N} g_i(\beta) \quad \blacksquare
\end{aligned}$$

(B)

I think all this proof needs is:

$$E[n g_i(\beta)] = n E_{\{x_i, y_i\} \in \{X, Y\}}[g_i(\beta)] = n \frac{1}{n} \sum_{i=1}^{n} g_i(\beta) = \nabla l(\beta)$$

(C)

To start, I built my code on top of the functions from the first homework, looping through a fixed number of iterations, with the following inner loop:

```
...
pt = sample(1:N, 1)
xSample = matrix(X[pt,], nrow = 1)
ySample = matrix(Y[pt,], nrow = 1)
gL = gradL(xSample, beta, ySample, m)
beta = beta - step * gL
...
```

where `gradL` computes the gradient of the negative log-likelihood. This selects a point (with replacement), computes the gradient of the log-likelihood, and uses it to update β. This way, all the rest of the code from homework 1 can be re-used.

Notes from class

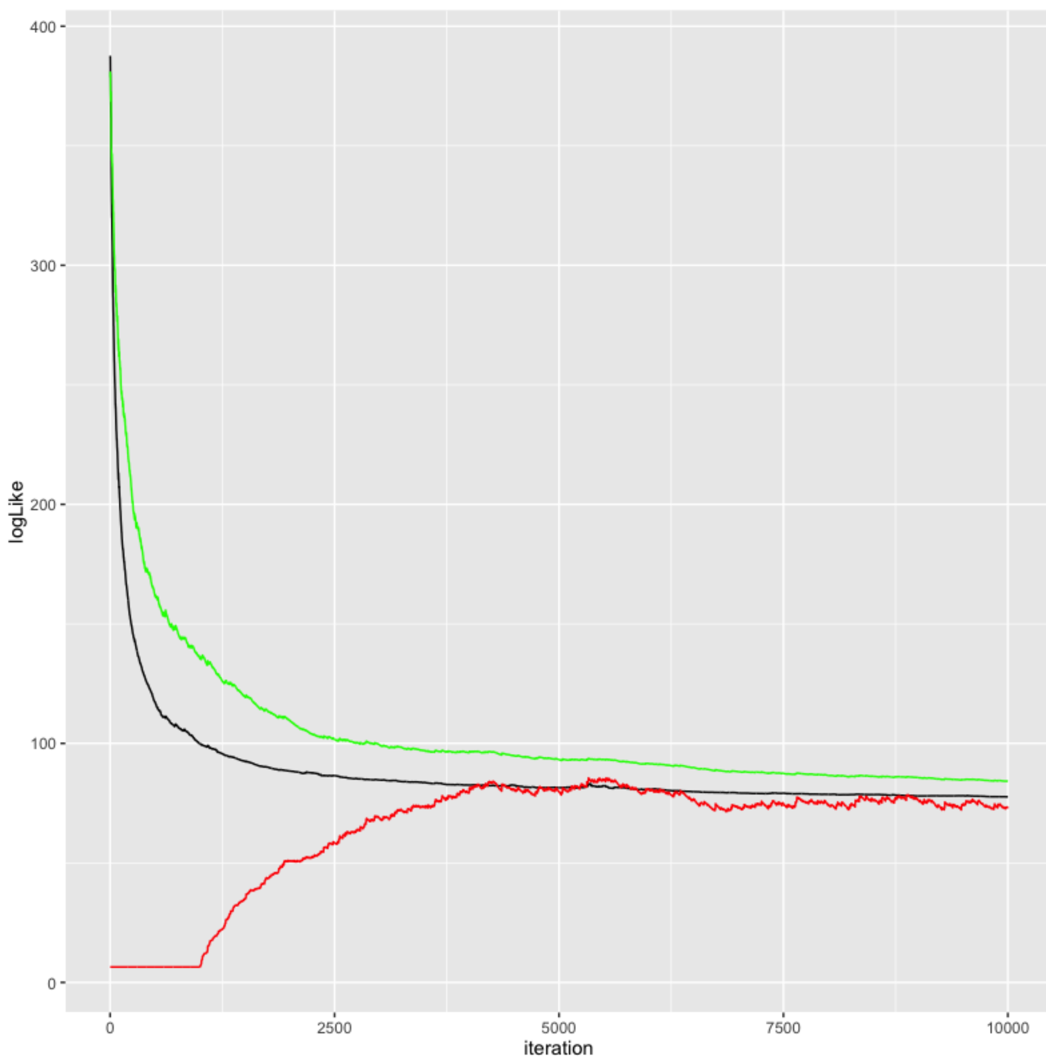

Figure 1: Results for code in part C with $\gamma^{(t)} = \gamma = 0.01$, 10000 iterations, and a constant of $\alpha = 0.5/N$ for the exponentially-weighted moving average. In black, the full negative log-likelihood, computed on all data for reference. In green, the simple moving average. In red, the exponentially-weighted moving average.