Exercise 2

Bowen Hua

September 26, 2017

1 Stochastic Gradient Descent

1.1 (A)

Already shown in Exercise 1.

1.2 (B)

The random variable in SGD is i, the index of the datapoint chosen. It has the following distribution:

$$P(i=j)=\begin{cases}\frac{1}{n} & j\in\{1,2,\dots,n\}\\ 0 & \text{otherwise.}\end{cases} \tag{1}$$

Therefore,

$$\begin{aligned}\mathbb{E}(ng_i(\beta)) &= n\sum_{j=1}^{n} g_j(\beta)P(i=j) &(2)\\ &= \sum_{j=1}^{n} g_j(\beta) &(3)\\ &= \nabla \ell(\beta). &(4)\end{aligned}$$

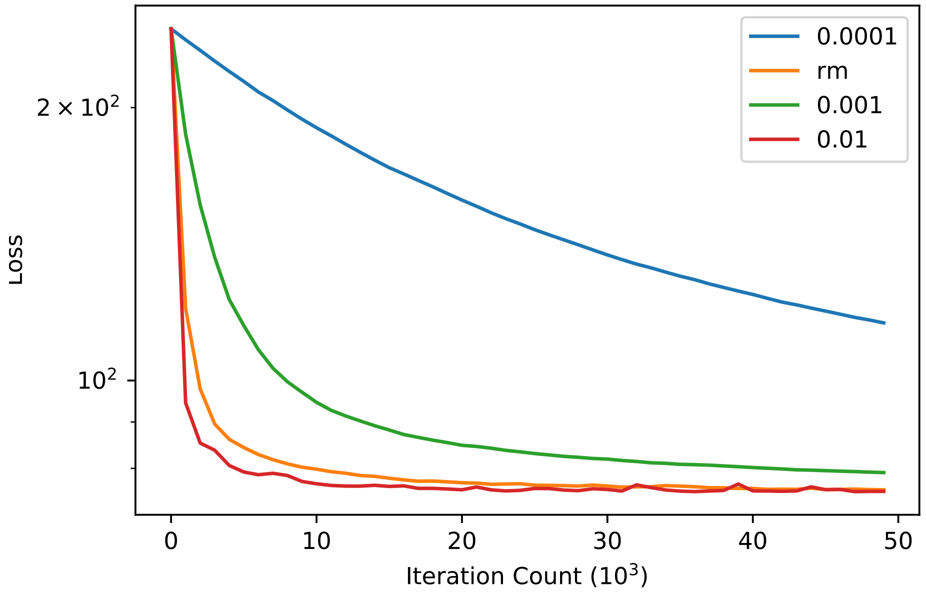
1.3 (C)

My codes are shown in the `logit_sgd.py` file. An important feature of my codes is that I re-use the functions used to compute the gradient and negative log likelihood. The only difference in SGD is that, instead of using arguments X (feature matrix) and y (labels), I use $X[i,:]$ and $y[i]$ as arguments, where i is our sample of a data point. The sampling is done *with replacement*.

We perform SGD with $50,000$ iterations of samples with replacement and constant step sizes of 0.0001, 0.001, 0.01. Our initial guess for β is randomly generated. We compute the loss function for every $1,000$ iterations.

Fig. 1 shows the negative log likelihood as a function of iteration count. We can see that for a constant step size of 0.0001, the convergence is slow; minimum is not reached after $50,000$ steps. For a constant step size of 0.01, the step size might be too large when we are close to the optimum, as can be seen from the zigzag curve in Fig. 1.

Figure 1: Loss function value as a function of iteration count



1.4 (D)

I use the following parameters for the RobbinsMonro rule: $C = 1$, $t_0 = 2$, $\alpha = 0.5$. The negative log likelihood as a function of iteration count is also shown in Fig. 1. We can see that the RobbinsMonro rule for step size performs well on this dataset; there is no zigzag behavior when we are near the minimum.