Stat 610 Homework 8

Due Wednesday, November 13, 11:59pm

Assignment

In this assignment, you will implement stochastic gradient descent and mini-batch gradient descent for logistic regression. Recall that in logistic regression, we are trying to find the coefficient vector β that maximizes the log likelihood of the data, or, equivalently, minimizes the negative log likelihood of the data. If $\mathbf{x}_i, i = 1, \ldots, n$ are p-vectors giving the values of p predictor variables for the ith sample and $y_i, i = 1, \ldots, n$ are 0/1-valued response variables, then the negative log likelihood for the logistic regression model is

$$\begin{aligned} -\ell(\beta) &= \sum_{i=1}^{n} f_i(\beta) \\ f_i(\beta) &= -[y_i \log(p_i(\beta)) + (1 - y_i) \log(1 - p_i(\beta))] \end{aligned}$$

where

$$p_i(\beta) = \frac{\exp(\mathbf{x}_i^T \beta)}{1 + \exp(\mathbf{x}_i^T \beta)}$$

The gradient is then

$$f'(\beta) = \sum_{i=1}^{n} f_i'(\beta) = -\sum_{i=1}^{n} (y_i - p_i(\beta)) \mathbf{x}_i$$

and in standard gradient descent with a fixed step size η we would start with some guess $\beta^{(0)}$ for the optimal value of β, and then iterate

$$\beta^{(t+1)} = \beta^{(t)} - \eta_t f'(\beta)$$

until $f'(\beta)$ is sufficiently small.

For stochastic gradient descent, we notice that $f'(\beta)$ is a sum, and it can be interpreted as an estimator of a "true" gradient. With that interpretation, any single element of the sum is also an estimator of the true gradient (albeit a noisier one).

The stochastic gradient descent algorithm is based on this idea, and it is as follows:

1. Choose some starting value for β and set $t = 1$.
2. Randomly permute the samples.
3. For $i = 1, \ldots, n$, (i being the index within the permuted samples) update $\beta \leftarrow \beta - \eta_t f_i'(\beta)$.
4. Update $t \leftarrow t + 1$.
5. If the algorithm has converged, stop, otherwise return to (2).

You should do the following:

1. Generate data from a logistic regression model like we did in class (https://jfukuyama.github.io/teaching/stat610/notes/lecture19.html#(25)), but with a larger number of points, say, $n = 10^4$.
2. Create a function that takes performs gradient descent for the logistic regression model. Your function should take the data, a starting value for the coefficients, and the number of iterations to perform (for the purposes of this assignment, don't bother with checking for convergence in the function, just perform a set number of iterations). In addition to returning the optimal value for the coefficient vector, your function should record the log likelihood at each iteration.
3. Create a function that performs stochastic gradient descent for the logistic regression model. The function should again take the data, a start value for the coefficients, and a number of iterations to perform. As for standard gradient descent, it should return the optimal value for the coefficient vector, and it should further record the log likelihood after every gradient update.
4. Apply your two functions to your simulated data. Plot the log likelihood as a function of iteration for the two methods, noting that every gradient step for standard gradient descent takes about n times as much computational effort as a gradient step for stochastic gradient descent.
5. Bonus: Do the same thing for mini batch gradient descent. In mini batch gradient descent, instead of having updates $\beta \leftarrow \beta - \eta_t f_i'(\beta)$, we partition the samples into K groups, $S_1, \ldots, S_K$. Then we repeat many times: for $k = 1, \ldots, K$, make the update $\beta \leftarrow \beta - \eta_t \sum_{i \in S_k} f_i'(\beta)$.

Submission parameters

You should submit an Rmd file and the corresponding pdf or html on canvas. The files should contain both the code you ran and explanations.