CSE 6242 Assignment 3

Vincent La (Georgia Tech ID - vla6)

October 21, 2017

Question 1: Data Preprocessing

Question 2: Theory

Part a: Write down the formula for the loss function used in Logistic Regression, the expression that you want to minimize: L(θ)

Taken from the lecture "MLE and Iterative Optimization":

$$\hat{\theta}_{MLE} = argmin_{\theta} \sum_{i=1}^{n} log(1 + e^{y^{(i)} * <\theta, x^{(i)}>})$$

Thus, the loss function is

$$L(\theta) = \sum_{i=1}^{n} log(1 + e^{y^{(i)} * <\theta, x^{(i)}>})$$

where $y^{(i)} = 1$ or $y^{(i)} = -1$.

Part b: Derive the gradient of the loss function with respect to model parameters: $\frac{dL(\theta)}{d\theta}$ or $\frac{\partial L(\theta)}{\partial \theta_j}$.

$$\frac{\partial L(\theta)}{\partial \theta_j} = \frac{\partial \sum_{i=1}^{n} log(1 + e^{y^{(i)} * <\theta, x^{(i)}>})}{\theta_j}$$

Furthermore, we know that $\frac{\partial}{\partial x} log(x) = \frac{1}{x}$. Also, we can use the chain rule here to complete the derivative.

$$\frac{\partial L(\theta)}{\partial \theta_j} = \sum_{i=1}^{n} \frac{1}{1 + e^{y^{(i)} * <\theta, x^{(i)}>}} * \frac{\partial}{\partial \theta_j} e^{y^{(i)} * <\theta, x^{(i)}>}$$

$$\frac{\partial L(\theta)}{\partial \theta_j} = \sum_{i=1}^{n} \frac{e^{y^{(i)} * <\theta, x^{(i)}>}}{1 + e^{y^{(i)} * <\theta, x^{(i)}>}} * \frac{\partial}{\partial \theta_j} (y^{(i)} * < \theta, x^{(i)} >)$$

$$\frac{\partial L(\theta)}{\partial \theta_j} = \sum_{i=1}^{n} \frac{(e^{y^{(i)} * <\theta, x^{(i)}>}) * y^{(i)} x_j^{(i)}}{1 + e^{y^{(i)} * <\theta, x^{(i)}>}}$$

Part c: Based on this gradient, express the Stochastic Gradient Descent (SGD) update rule that uses a single sample $< x^{(i)}, y^{(i)} >$ at a time.

Stochastic Gradient Descent (SGD) can be used when your data is very big. The steps are:

1. Initialize the dimensions of θ vector to random values.
2. Pick one labeled data vector $(x^{(i)}, y^{(i)})$ randomly, and update for each $j = 1, ..., d : \theta_i \leftarrow \theta_j - \alpha \frac{\partial log(1+exp(y^{(i)} < \theta, x^{(i)} >))}{\partial \theta_j}$
3. Repeat step (2) until the updates of the dimensions of θ become too small.

So basically substituting in the expression for $\frac{\partial}{\partial \theta_j} log(1 + exp(y^{(i)} < \theta, x^{(i)} >))$ that we found previously, we get:

$$\frac{\partial}{\partial \theta_j} log(1 + exp(y^{(i)} < \theta, x^{(i)} >)) = \frac{(e^{y^{(i)} * < \theta, x^{(i)} >}) * y^{(i)} x_j^{(i)}}{1 + e^{y^{(i)} * < \theta, x^{(i)} >}}$$

Thus, the update rule becomes,

for each $j = 1, ..., d : \theta_i \leftarrow \theta_j - \alpha * \frac{(e^{y^{(i)} * < \theta, x^{(i)} >}) * y^{(i)} x_j^{(i)}}{1 + e^{y^{(i)} * < \theta, x^{(i)} >}}$

Part d: Write pseudocode for training a model using Logistic Regression and SGD.

1. for (j from 1, . . . , d), initialize θ_j to random values. (Initialize the dimensions of θ vector to random values.)
2. Pick one labeled data vector randomly, call it $(x^{(i)}, y^{(i)})$
3. for (j from 1, . . . , d) set θ_i equal to $\theta_j - \alpha * \frac{(e^{y^{(i)} * < \theta, x^{(i)} >}) * y^{(i)} x_j^{(i)}}{1 + e^{y^{(i)} * < \theta, x^{(i)} >}}$, where α is some step size, decaying as the gradient descent iterations increase.
4. Repeat step (3) until the updates of the dimensions of θ become too small.

Part e: Estimate the number of operations per epoch of SGD, where an epoch is one complete iteration through all the training samples. Express this is Big-O notation, in terms of the number of samples (n) and the dimensionality of each sample (d).