Sam Mikes
Math189R SU20.1
Homework 2
Tuesday June 16, 2020

1 (**Murphy 8.3**) Gradient and Hessian of the log-likelihood for logistic regression.
(a) Let $\sigma(x) = \frac{1}{1+e^{-x}}$ be the sigmoid function. Show that

$$\sigma'(x) = \sigma(x)\left[1 - \sigma(x)\right].$$

(b) Using the previous result and the chain rule of calculus, derive an expression for the gradient of the log likelihood for logistic regression.
(c) The Hessian can be written as $\mathbf{H} = \mathbf{X}^\top \mathbf{S}\mathbf{X}$ where $\mathbf{S} = \text{diag}(\mu_1(1-\mu_1), \ldots, \mu_n(1-\mu_n))$. Derive this and show that $\mathbf{H} \succeq 0$ ($A \succeq 0$ means that A is positive semidefinite).

Hint: Use the **negative** log-likelihood of logistic regression for this problem.

1. Show that $\sigma'(x) = \sigma(x)\left[1 - \sigma(x)\right].$

$$\begin{aligned}\frac{\partial \sigma}{\partial x} &= \frac{\partial}{\partial x}\frac{1}{1+e^{-x}} = -(\frac{1}{1+e^{-x}})^2 \cdot \frac{\partial}{\partial x}(1+e^{-x}) \\ &= (\frac{1}{1+e^{-x}}) \cdot (\frac{1}{1+e^{-x}}) \cdot e^{-x} \\ &= \sigma \cdot (\frac{e^{-x}}{1+e^{-x}})\end{aligned} \tag{1}$$

Since

$$\begin{aligned}1 - \sigma = 1 - (\frac{1}{1+e^{-x}}) &= \frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}} \\ &= \frac{1+e^{-x}-1}{1+e^{-x}} \\ &= \frac{e^{-x}}{1+e^{-x}}\end{aligned} \tag{2}$$

Substitute (2) into (1) to obtain

$$\frac{\partial \sigma}{\partial x} = \sigma(1-\sigma) \tag{3}$$

2. Derive an expression for the gradient of the log likelihood.

Recall NLL is defined (**Murphy Equation 8.3**)

$$NLL(\mathbf{w}) = \sum_{i=1}^{N}[y_i \log \mu_i + (1-y_i) log(1-\mu_i)] \tag{4}$$

Then

$$\delta NLL(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} \sum_{i=1}^{N} [y_i \log \mu_i + (1-y_i) log(1-\mu_i)] \tag{5}$$

Each term in the sum

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}}[y_i \log \mu_i + (1-y_i) log(1-\mu_i)] &= y_i(\frac{1}{\mu_i}\partial\mu_i\partial\mathbf{w} + (1-y_i)\frac{1}{1-\mu_i}\partial\partial\mathbf{w}(1-\mu_i) \\
&= \frac{y_i\mu_i(1-\mu_i)}{\mu_i} + \frac{(1-y_i)(1-\mu_i)(-1)(1-(1-\mu_i))}{1-\mu_i} \\
&= y_i(1-\mu_i) - (1-y_i)(\mu_i) \\
&= y_i - y_i\mu_i - \mu_i + y_i\mu_i \\
&= y_i - \mu_i
\end{aligned} \tag{6}$$

As expected.

3. 3. Not completed.

■

2 (**Murphy 2.11**) Derive the normalization constant (Z) for a one dimensional zero-mean Gaussian

$$\mathbb{P}(x;\sigma^2) = \frac{1}{Z}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

such that $\mathbb{P}(x;\sigma^2)$ becomes a valid density.

Not completed.

Would reference well-known proof where change of variable from $x- > y$, multiply two functions together, change variables to r, θ, integral is now in $rdrd\theta$ instead of $dxdy$, and thus the integral yields $Z^2 = 2\pi/\sigma^2$

■

3 (**regression**). In this problem, we will use the online news popularity dataset to set up a model for linear regression. In the starter code, we have already parsed the data for you. However, you might need internet connection to access the data and therefore successfully run the starter code.

We split the csv file into a training and test set with the first two thirds of the data in the training set and the rest for testing. Of the testing data, we split the first half into a 'validation set' (used to optimize hyperparameters while leaving your testing data pristine) and the remaining half as your test set. We will use this data for the remainder of the problem. The goal of this data is to predict the **log** number of shares a news article will have given the other features.

(a) (**math**) Show that the maximum a posteriori problem for linear regression with a zero-mean Gaussian prior $\mathbb{P}(\mathbf{w}) = \prod_j \mathcal{N}(w_j|0, \tau^2)$ on the weights,

$$\arg\max_{\mathbf{w}} \sum_{i=1}^{N} \log \mathcal{N}(y_i|w_0 + \mathbf{w}^\top \mathbf{x}_i, \sigma^2) + \sum_{j=1}^{D} \log \mathcal{N}(w_j|0, \tau^2)$$

is equivalent to the ridge regression problem

$$\arg\min \frac{1}{N} \sum_{i=1}^{N} (y_i - (w_0 + \mathbf{w}^\top \mathbf{x}_i))^2 + \lambda ||\mathbf{w}||_2^2$$

with $\lambda = \sigma^2/\tau^2$.

(b) (**math**) Find a closed form solution $\mathbf{x}^\star$ to the ridge regression problem:

$$\text{minimize: } ||A\mathbf{x} - \mathbf{b}||_2^2 + ||\Gamma \mathbf{x}||_2^2.$$

(c) (**implementation**) Attempt to predict the log shares using ridge regression from the previous problem solution. Make sure you include a bias term and *don't regularize the bias term*. Find the optimal regularization parameter λ from the validation set. Plot both λ versus the validation RMSE (you should have tried at least 150 parameter settings randomly chosen between 0.0 and 150.0 because the dataset is small) and λ versus $||\boldsymbol{\theta}^\star||_2$ where $\boldsymbol{\theta}$ is your weight vector. What is the final RMSE on the test set with the optimal $\lambda^\star$?

(continued on the following pages)

■

3 (continued)

(d) (**math**) Consider regularized linear regression where we pull the basis term out of the feature vectors. That is, instead of computing $\hat{\mathbf{y}} = \boldsymbol{\theta}^\top \mathbf{x}$ with $\mathbf{x}_0 = 1$, we compute $\hat{\mathbf{y}} = \boldsymbol{\theta}^\top \mathbf{x} + b$. This corresponds to solving the optimization problem

$$\text{minimize: } ||A\mathbf{x} + b\mathbf{1} - \mathbf{y}||_2^2 + ||\Gamma \mathbf{x}||_2^2.$$

Solve for the optimal $\mathbf{x}^\star$ explicitly. Use this close form to compute the bias term for the previous problem (with the same regularization strategy). Make sure it is the same.

(e) (**implementation**) We can also compute the solution to the least squares problem using gradient descent. Consider the same bias-relocated objective

$$\text{minimize: } f = ||A\mathbf{x} + b\mathbf{1} - \mathbf{y}||_2^2 + ||\Gamma \mathbf{x}||_2^2.$$

Compute the gradients and run gradient descent. Plot the ℓ_2 norm between the optimal $(\mathbf{x}^\star, b^\star)$ vector you computed in closed form and the iterates generated by gradient descent. Hint: your plot should move down and to the left and approach zero as the number of iterations increases. If it doesn't, try decreasing the learning rate.

Partially completed. See attached python code hw2pr3.py. In order to learn the techniques and get un-stuck, I referred to the published solution of a previous class homework:

`https://github.com/math189bigdata/math189bigdata.github.io/homework/hw2/hw2_sol/hw2pr3_sol.py`

■