Statistical Machine Learning: Homework #2

Professor Devika Subramanian

Marco Lagos & Apoorv Walia

Problem 1

Gradient and Hessian for binary logistic regression
Compute the gradient and Hessian of $J(\theta)$ for binary logistic regression.

Part 1 - Gradient of the sigmoid
Let $g(z) = \frac{1}{1+e^{-z}}$. Show that $\frac{\delta g(z)}{\delta z} = g(z)(1 - g(z))$.

Solution
By the quotient rule for differentiation:

$$\begin{aligned} g'(z) &= \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right) \\ &= \frac{0 \cdot (1+e^{-z}) - 1 \cdot (-e^{-z})}{(1+e^{-z})^2} \\ &= \frac{e^{-z}}{(1+e^{-z})^2} \end{aligned}$$

With further simplification:

$$\begin{aligned} g'(z) &= \frac{e^{-z}}{(1+e^{-z})^2} \\ &= \frac{e^{-z}}{(1+e^{-z})(1+e^{-z})} \\ &= \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}} \\ &= \frac{1}{1+e^{-z}} \cdot \left(1 - 1 + \frac{e^{-z}}{1+e^{-z}}\right) \\ &= \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1+e^{-z}}{1+e^{-z}} + \frac{e^{-z}}{1+e^{-z}}\right) \\ &= \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1}{1+e^{-z}}\right) \\ &= g(z) \cdot (1 - g(z)) \quad \text{since } g(z) = \frac{1}{1+e^{-z}} \end{aligned}$$

Part 2 - Gradient of L2 penalized binary logistic regression
Using the previous result and the chain rule of calculus, derive the expression for the gradient of the L2 penalized cost function $J(\theta)$ (shown below) for logistic regression. $\lambda > 0$ is the regularization parameter.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)} log(1 - h_\theta(x^{(i)})))) + \frac{\lambda}{2m}\sum_{j=1}^{d}\theta_j^2$$

Solution
The L2 penalized cost function is defined as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))\right) + \frac{\lambda}{2m}\sum_{j=1}^{d}\theta_j^2$$

where:

- m is the number of training examples.

Problem 1 continued on next page. . .

- d is the number of features.
- $y^{(i)}$ is the target value for the i-th example.
- $x^{(i)}$ is the feature vector for the i-th example.
- $h_\theta(x^{(i)})$ is the sigmoid function, which is denoted as $g(z)$, where $z = \theta^T x^{(i)}$.
- $\lambda > 0$ is the regularization parameter.

First, let us recompute Part 1 for $h_\theta(x^{(i)}) = g(\theta^T x^{(i)})$, where $z = \theta^T x^{(i)}$:

$$\begin{aligned}\frac{\partial}{\partial \theta_j} h_\theta(x^{(i)}) &= \frac{\partial}{\partial \theta_j} g(\theta^T x^{(i)}) \\ &= g(\theta^T x^{(i)}) \cdot (1 - g(\theta^T x^{(i)})) \cdot \frac{\partial}{\partial \theta_j}(\theta^T x^{(i)}) \\ &= g(\theta^T x^{(i)}) \cdot (1 - g(\theta^T x^{(i)}))x_j^{(i)}\end{aligned}$$

Deriving the gradient of $J(\theta)$ with respect to θ by the chain rule and by Part 1 on non-regularized term first

$$\begin{aligned}\nabla J(\theta) &= \frac{\partial}{\partial \theta_j}\left(-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right)\right) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\frac{1}{h_\theta(x^{(i)})}\frac{\partial}{\partial \theta_j}h_\theta(x^{(i)}) + (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\frac{\partial}{\partial \theta_j}(1-h_\theta(x^{(i)}))\right) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\frac{1}{h_\theta(x^{(i)})}h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))x_j^{(i)} - (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}(-h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))x_j^{(i)})\right) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}(1-h_\theta(x^{(i)})) - (1-y^{(i)})h_\theta(x^{(i)})\right)x_j^{(i)} \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - y^{(i)}h_\theta(x^{(i)}) - h_\theta(x^{(i)}) + y^{(i)}h_\theta(x^{(i)})\right)x_j^{(i)} \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - h_\theta(x^{(i)})\right)x_j^{(i)}\end{aligned}$$

Computing the gradient of the regularization term:

$$\frac{\partial}{\partial \theta_j}\left(\frac{\lambda}{2m}\sum_{j=1}^{d}\theta_j^2\right) = \frac{\lambda}{2m}\frac{d}{d\theta_j}\left(\sum_{j=1}^{d}\theta_j^2\right) = \frac{\lambda}{2m}\cdot 2\theta_j = \frac{\lambda}{m}\theta_j$$

Combining both parts, we get:

$$\nabla J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} + \frac{\lambda}{m}\theta_j$$

Part 3 - Vector form of gradient for L2 penalized binary logistic regression
Derive the vector form of the first derivative of the L2-penalized $J(\theta)$ with respect to θ.

Solution
The gradient for this cost function is:

$$\nabla J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x^{(i)} + \frac{\lambda}{m}\theta$$

- $h_\theta(X)$ is an m-row prediction vector for all training examples, where X is matrix of training examples (each row is an example).
- y is an $m \times 1$ vector of true target values for training examples
- X is matrix of all input features for training examples
- θ is parameter vector that has all model parameters

We can re-write as:

$$\nabla J(\theta) = \frac{1}{m} X^T (h_\theta(X) - y) + \frac{\lambda}{m}\theta$$

Part 4 - Hessian of L2 penalized binary logistic regression
Show that the Hessian or second derivative of $J(\theta)$ can be written as

$$H = \frac{1}{m}(X^T S X + \lambda I)$$

$$S = diag(h_\theta(x^{(1)})(1 - h_\theta(x^{(1)})), ..., h_\theta(x^{(m)})(1 - h_\theta(x^{(m)})))$$

Show that H is positive definite. You may assume that $0 < h_\theta(x^{(i)}) < 1$ so the elements of S are strictly positive and that X is full rank.

Solution
S is a diagonal matrix with the elements $h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))$ on the diagonal. This can be represented as:

$$S = \text{diag}(h_\theta(X)(1 - h_\theta(X)))$$

Where $h_\theta(X)$ is a vector containing the predicted values for all training examples. Let us rewrite the gradient $\nabla J(\theta)$ using H, S, X, and θ:

$$\nabla J(\theta) = \frac{1}{m} X^T (h_\theta(X) - y) + \frac{\lambda}{m}\theta = \frac{1}{m} X^T S(X\theta - y) + \frac{\lambda}{m}\theta$$

where $X\theta$ represents the predictions $h_\theta(X)$ for all training examples. Let compute the Hessian of $J(\theta)$:

$$\begin{aligned}
H &= \frac{\partial}{\partial\theta}\left(\frac{1}{m} X^T S(X\theta - y) + \frac{\lambda}{m}\theta\right) \\
&= \frac{1}{m}\left(\frac{\partial}{\partial\theta}\left(X^T S X\theta - X^T S y\right) + \frac{\partial}{\partial\theta}\lambda\theta\right) \\
&= \frac{1}{m}\left(X^T S X + \lambda I\right)
\end{aligned}$$

A matrix is positive definite if it's symmetric and all its eigenvalues are positive. First, let us prove that H is symmetric. By definition, a symmetric matrix is equal to its transpose:

$$\begin{aligned}
H^T &= \left(\frac{1}{m}(X^T S X + \lambda I)\right)^T \\
&= \frac{1}{m}(X^T S X + \lambda I)^T \\
&= \frac{1}{m}(X^T (SX)^T + (\lambda I)^T) \\
&= \frac{1}{m}(X^T S X + \lambda I) = H
\end{aligned}$$

Given that $0 < h_\theta(x^{(i)}) < 1$ for all i, then all elements on the diagonal of the matrix S are strictly positive since they are of the form $h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))$.

Now, let's prove that all eigenvalues of H are strictly positive. Consider $H = \frac{1}{m}(X^TSX + \lambda I)$. Let $A = X^TSX$ and $B = \lambda I$. Since X is full rank, all of its columns of linearly independent; thus, X^TX is positive definite (all of its eigenvalues are strictly positive). Similarly, since all elements of S are strictly positive, S is also positive definite.

Now, consider λ_A and λ_B, the eigenvalues of A and B, respectively. Then, the eigenvalues of H are:

$$\lambda_H = \frac{1}{m}(\lambda_A + \lambda_B)$$

Since both λ_A and λ_B are positive, then λ_H is also positive. Therefore, all eigenvalues of H are strictly positive.
Since $H = H^T$ and $\lambda_H > 0$, then we can conclude that H is positive definite.

Part 5 - Newton's method
Now use these results to update the θ vector using Newton's method. We have a 2D training set composed of the data matrix X and the vector y.
The matrix X is:

$$\begin{bmatrix} 0 & 3 \\ 1 & 3 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

The vector y is:

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Prepend a 1 to each $x^{(i)}$ in the training set so that we can model the intercept or bias term in θ.

- State the θ update equation for an iteration of Newton's method for this problem
- Assume a starting $\theta = [0, -1, 1]^T$ and a regularization parameter $\lambda = 0.07$. Compute and provide the values of θ after the first and second iteration of Newton's method, using a Python script.

Solution
By Newton's method:

$$\theta^{(t+1)} = \theta^{(t)} - \left(H^{-1}\nabla J(\theta^{(t)})\right)$$

where

- $\theta^{(t)}$ is the parameter vector at iteration t.
- H is the Hessian matrix of the cost function $J(\theta)$.
- $\nabla J(\theta)$ is the gradient vector of the cost function.

Gradient vector:

$$\nabla J(\theta) = \frac{1}{m}X^T(h_\theta(X) - y) + \frac{\lambda}{m}\theta$$

Hessian matrix:

$$H = \frac{1}{m}X^TSX + \frac{\lambda}{m}I$$

See **newtons_method.py**

Problem 2

Estimating the parameter of a Bernoulli distribution

Consider a data set $D = \{x^{(i)} | 1 \leq i \leq m\}$ where $x^{(i)}$ is drawn from a Bernoulli distribution with parameter θ. The elements of the data set are the results of the flips of a coin where $x^{(i)} = 1$ represents *heads* and $x^{(i)} = 0$ represents *tails*. We will estimate the parameter θ, which is the probability of the coin coming up heads, using the data set D.

Part 1 - MLE estimation
Use the mthod of MLE to derive an estimate for θ from the coin flip results in D.

Part 2 - MAP estimation
Assume a beta prior distribution on θ with hyperparameters a and b. The beta distribution is chosen because it has the same form as the likelihood function for D derived under the Bernoulli model (such a prior is called a conjugate prior).

$$Beta(\theta|a, b) \propto \theta^{a-1}(1-\theta)^{b-1}$$

Derive the MAP estimate for θ using D and this prior distribution. Show that under a uniform prior (Beta distribution with $a = b = 1$), the MAP and MLE estimates of θ are equal.

Problem 3

Logistic regression and Gaussian Naive Bayes
Consider a binary classification problem with dataset $D = \{(x^{(i)}, y^{(i)}) | 1 \leq i \leq m; x^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{0, 1\}\}$. You will derive a connection between logistic regression and Gaussian Naive Bayes for this classification problem.

For logistic regression, we use the sigmoid function $g(\theta^T x)$, where $\theta \in \mathbb{R}^{d+1}$ and we augment x with a 1 in front to account for the intercept term θ_0. For the Gaussian Naive Bayes model, assume that the y's are drawn from a Bernoulli distribution with parameter γ, and that each x_j from class 1 is drawn from a univariate Gaussian distribution with mean μ_j^1 and variance σ_j^2, and each x_j from class 0 is drawn from a univariate Gaussian distribution with mean μ_j^0 and variance σ_j^2. Note that the variance is the same for both classes, just he means are different.

Part 1 - Posterior probabilities in logistic regression
For logistic regression, what is the posterior probability for each class, i.e., $P(y = 1|x)$ and $P(y = 0|x)$? Write the expression in terms of the parameters θ and the sigmoid function.

Part 2 - Posterior probabilities in Gaussian Naive Bayes
Derive the posterior probabilities for each class, $P(y = 1|x)$ and $P(y = 0|x)$, for the Gaussian Naive Bayes model, using Bayes rule, the (Gaussian) distribution on the x_j's, $j = 1, ..., d$ and the Naive Bayes assumption.

Part 3 - Relating LR and GNB
Assuming that class 1 and class 0 are equally likely (uniform class priors), simplify the expression for $P(y = 1|x)$ for Gaussian Naive Bayes. Show that with appropriate parameterization, $P(y = 1|x)$ for Gaussian Naive Bayes with uniform priors is equivalent to $P(y = 1|x)$ for logistic regression.

Problem 4

See **softmax_cifar10.ipynb**