CS M146 Quiz 4 Solutions

1 Logistic Regression

For this question, $\mathbf{x}$ is the feature vector, x, x_i's are scalar features, and y is the label.

1.1

Suppose we build a logistic regression classifier $P(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$ for a binary classification task. Which one of the following statements about logistic regression is correct?

- In general, logistic regression and perceptron learn the same parameters $\mathbf{w}, b$ for the decision boundary.
- The decision boundary of logistic regression is nonlinear.
- Logistic regression minimizes the negative log likelihood of the training data.
- Logistic regression is guaranteed to maximize the training accuracy.

Answer: 3. 1 is wrong because logistic regression and perceptron optimize different loss functions. 2 is wrong because the decision boundary is linear. 3 is correct as the loss for logistic regression is the negative log likelihood of the training data. 4 is wrong because the loss function for logistic regression, negative log likelihood, does not involve explicitly calculating or optimizing training accuracy. Thus, logistic regression is not guaranteed to maximize the training accuracy.

1.2

Given a binary logistic regression model $P(y = 1|x) = \sigma(2x + 3)$ where σ is the sigmoid function. What is the decision boundary?

- $x = 0$
- $2x = 3$
- $2x = -3$

- $x = -2$

Answer: 3. The decision boundary is 2x+3=0, or 2x=-3.

1.3

Given a logistic regression model $P(y = 1|\mathbf{x}) = \sigma(1 + 0.7x_1 + 0.2x_2 - x_3)$ where σ is the sigmoid function, increasing the value of the first feature (keeping the value of the other features fixed) decreases the model's output probability for $y = 1$.

Answer: False. Increasing x_1 increases $P(y = 1|\mathbf{x})$ since $1+0.7x_1+0.2x_2-x_3$ increases with x_1 and sigmoid is monotonically increasing.

1.4

Given a logistic regression model $P(y = 1|\mathbf{x}) = \sigma(1 + 0.7x_1 + 0.2x_2 - x_3)$ where σ is the sigmoid function, increasing the value of the second feature (keeping the value of the other features fixed) decreases the model's output probability for $y = 1$.

Answer: False. Increasing x_2 increases $P(y = 1|\mathbf{x})$ since $1+0.7x_1+0.2x_2-x_3$ increases with x_2 and sigmoid is monotonically increasing.

1.5

Given a logistic regression model $P(y = 1|\mathbf{x}) = \sigma(1 + 0.7x_1 + 0.2x_2 - x_3)$ where σ is the sigmoid function, when all features take the value 0, the model predicts $y = 1$.

Answer: True. With all features 0, the model outputs $P(y = 1|\mathbf{x}) = \sigma(1) > \sigma(0) = 0.5$ as sigma is monotonically increasing.

2 Convexity

2.1

The following function $f(x) = ax^4$ is convex for any $a \in R$.

Answer: False. $f''(x) = 12ax^2 < 0$ if $a < 0$, so $f(x)$ is not convex if $a < 0$.

2.2

If $g(x)$ is convex, then $h(x) = g(ax + b)$ is also convex for any $a, b \in R$

Answer: True. This is a standard property of convex functions. The proof of this statement is below.

We need to show that h is convex, i.e. for any x, y and any $\lambda, 0 \leq \lambda \leq 1$,

$$h(\lambda x + (1 - \lambda)y) \leq \lambda h(x) + (1 - \lambda)h(y).$$

By definition of h, we have that

$$h(\lambda x + (1-\lambda)y) = g(a(\lambda x + (1-\lambda)y) + b),$$

so

$$h(\lambda x + (1-\lambda)y) = g(\lambda(ax+b) + (1-\lambda)(ay+b))$$

. As g is convex, we have that

$$g(\lambda(ax+b) + (1-\lambda)(ay+b)) \leq \lambda g(ax+b) + (1-\lambda)g(ay+b)$$

and hence,

$$h(\lambda x + (1-\lambda)y) \leq \lambda h(x) + (1-\lambda)h(y).$$

3 Optimization

3.1

In general, one iteration of stochastic gradient descent (SGD) has a greater runtime than one iteration of batch gradient descent (GD).

Answer: False. One iteration of SGD generally has a lower runtime than one iteration of GD because SGD needs to calculate the gradient of the loss function for only one data point while GD calculates this gradient for all data points.

3.2

Which of the following statements about gradient descent is correct?

- Gradient descent, using an appropriate step size, will converge to the global minimum of a nonconvex function.
- Gradient descent, using any step size, will converge to the local minimum of a function.
- Gradient descent is used to maximize a function.
- Gradient descent, using an appropriate step size, will converge to the local minimum of a function.

Answer: 4. 1 is wrong because gradient descent, even with the right step size, will converge only to a local minimum, not necessarily the global minimum, of a nonconvex function. 2 is wrong because if the step size is too large, gradient descent may not converge to a local minimum. 3 is wrong because gradient descent is used to minimize, not maximize, a function. 4 is correct.

4 Linear Regression

Consider fitting a linear regression $h_\theta(x) = \theta_0 + \theta_1 x$ to a training dataset $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$, where x_i's are features and y_i's are labels. Let $\mathcal{D}$ consist of 3 data points: $(x_1, y_1) = (0, 1), (x_2, y_2) = (1, 2), (x_3, y_3) = (-1, 1)$.

4.1

Our current estimate of the parameter is $(\theta_0, \theta_1) = (0, 0)$. Which training instance makes the largest contribution to the gradient of the loss function? (contribution means the magnitude, or 2-norm, of the gradient vector corresponding to the training instance)

Answer: (1, 2). The loss function for a single data point (x, y) is

$$J(\theta_0, \theta_1) = (y - \theta_0 - \theta_1 x)^2$$

. Computing the gradient, we get

$$\frac{\partial J}{\partial \theta_0} = -2(y - \theta_0 - \theta_1 x)$$

and

$$\frac{\partial J}{\partial \theta_1} = -2x(y - \theta_0 - \theta_1 x).$$

If we compute the gradient of the loss function for each data point, we get $(-2, 0), (-4, -4)$, and $(-2, 2)$, respectively. Hence, the second data point $x_2 = (1, 2)$ makes the largest contribution to the gradient of the loss function (for the dataset).

4.2

Our current estimate of the parameter is $(\theta_0, \theta_1) = (0, 0)$. Which vector has the same direction as the updated parameter after one step of gradient descent?

- $[-4, 1]$
- $[-4, -1]$
- $[4, 1]$
- $[4, -1]$

Answer: 3. We use the computation of the gradient from 4.1. The gradient for each data point is $(-2, 0), (-4, -4)$, and $(-2, 2)$, so the gradient of the loss function is $(-8, -2)$. However, as we are running gradient descent, the updated parameter will be $(0, 0) - \eta * (-8, -2)$ for some learning rate η, so $[4, 1]$ will have the same direction as the updated parameter.

4.3

What is the closed form solution for (θ_0, θ_1)?

- $[4, 1]$
- $[4/3, 1/2]$
- $[-4/3, 1/2]$
- $[-4, -1]$

Answer: 2. We use the formula for the 1-D case:

$$\theta_1 = \frac{\sum_n (x_n - \bar{x})(y_n - \bar{y})}{\sum_n (x_n - \bar{x})^2}, \theta_0 = \bar{y} - \theta_1 \bar{x},$$

where

$$\bar{x} = \frac{1}{n}\sum_n x_n, \bar{y} = \frac{1}{n}\sum_n y_n$$

. Applying this formula, we get $[4/3, 1/2]$.

5 Maximum Likelihood Estimation

Suppose we have n nonnegative numbers $x_1, ..., x_n \geq 0$ drawn independently from the same distribution. This distribution, with parameter θ, has probability density function $p(x;\theta) = \theta e^{-\theta x}$ for $x \geq 0, p(x;\theta) = 0$ for $x < 0$ (note that $\theta > 0$ so that $p(x;\theta)$ is a valid density function). What is the likelihood function $L(\theta)$?

- $\theta e^{-\theta \sum_{i=1}^n x_i}$
- $\theta^n e^{-\theta \sum_{i=1}^n x_i}$
- $\theta e^{-\theta x_1}$
- $\theta^n e^{-\theta x_1}$

Answer: 2. Since $x_1, ..., x_n$ are drawn indepndently and identically distributed (iid), we have that

$$L(\theta) = p(x_1, ..., x_n; \theta) = \prod_{i=1}^n p(x_i;\theta) = \theta^n e^{-\theta \sum_{i=1}^n x_i}$$

.