COMP4434 Big Data Analytics

Assignment 1

Author
Wang Yuqi
[redacted]

Lecturer
Prof. Xiao HUANG

BsC in Computer Science
Academic year: 2024/2025 Sem 2

Problems

Problem 1 (3 points)

Assuming that we are training a multivariate linear regression model using three instances as follows.

instance id	(x_1, x_2) aka $\mathbf{x}$	target value y
instance 1:	(1, 1.5)	2
instance 2:	$(-0.5, -1)$	0
instance 3:	$(2, 0.5)$	1

The cost function is based on mean squared error as follows.

$$\min_{\theta_0,\theta_1,\theta_2} \frac{1}{6}\sum_{i=1}^{3}\left(h_\theta(\mathbf{x}^i) - y^{(i)}\right)^2.$$

Assume that we have initialized

$$\mathbf{w} = [\theta_0, \theta_1, \theta_2] = [0, 0.5, 1].$$

You apply gradient descent algorithm to compute a $\mathbf{w}$ that minimizes the cost function w.r.t. the three given instances. Assume that in the first iteration, we set learning rate $\alpha = 0.6$. In the second iteration, we set learning rate $\alpha = 0.4$. Please concisely show how $\mathbf{w}$ is updated in the first and second iterations.

Problem 2 (1 points)

Assume that there are 1,500 documents in total. Among them, 800 documents are related to big data analysis. You build a model to identify documents related to big data analysis. As a result, your model returns 900 documents, but only 600 of them are relevant to big data analysis. What is the recall of your model? What is the F1 score of your model? Briefly justify your answer.

Problem 3 (0.5 points)

Explain why, in the context of cross-validation, test data and training data are randomly selected from the same dataset, even though it is generally stated that test data should be independent of training data. Provide a detailed explanation that reconciles this apparent contradiction.

Problem 4 (3.5 points)

Assume that we are training a logistic regression classifier using three instances as follows.

instance id	(x_1, x_2) aka $\mathbf{x}$	target value y
instance 1:	(1, 1.5)	1
instance 2:	$(-0.5, -1)$	0
instance 3:	$(2, 0.5)$	1

The logistic regression classifier is defined as follows.

$$h_\theta(\mathbf{x}) = h_\theta(x_1, x_2) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2)}}$$

The cost function is based on the logistic loss as follows.

$$\min_{\theta_0, \theta_1, \theta_2} -\frac{1}{3} \sum_{i=1}^{3} \left[y^{(i)} \log\left(h_\theta\left(\mathbf{x}^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(\mathbf{x}^{(i)}\right)\right) \right]$$

Assume that we have initialized

$$\mathbf{w} = [\theta_0, \theta_1, \theta_2] = [0, 0.5, 1].$$

You apply gradient descent algorithm to compute a **w** that minimizes the cost function w.r.t. the three given instances. Assume that in the first iteration, we set learning rate $\alpha = 0.6$. In the second iteration, we set learning rate $\alpha = 0.4$. Please concisely show how **w** is updated in the first and second iterations.

(See Next Page)

Solutions

Solution to Problem 1

Below is a step-by-step update of $\mathbf{w}$ given the instances, based on the update rule. (See **Appendix** at the bottom of this document for more detail)

Initialization

Let's start by padding the instances with a bias term, $x_0 = 1$:

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}. \text{ Instance } 1,\ 2,\ 3 = \begin{pmatrix} 1 \\ 1 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 1 \\ -0.5 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 0.5 \end{pmatrix},$$

First Iteration

$$\mathbf{w} = \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix}, \alpha = 0.6$$

	Instance 1	**Instance 2**	**Instance 3**
$\hat{y}$ or $\mathbf{w}^T\mathbf{x}$	2	-1.25	1.5
$\hat{y} - y$	0	-1.25	0.5
$\left(\hat{y}^{(i)} - y^{(i)}\right) \cdot \mathbf{x}$	$\begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$	$\begin{pmatrix} -1.25 \\ 0.625 \\ 1.25 \end{pmatrix}$	$\begin{pmatrix} 0.5 \\ 1.0 \\ 0.25 \end{pmatrix}$

Accumulate and averaging:

$$\begin{aligned} \mathbf{w} &\leftarrow \mathbf{w} - \alpha \cdot \frac{1}{3} \sum_{i=1}^{3} \left(\hat{y}^{(i)} - y^{(i)}\right) \cdot \mathbf{x} \\ &\leftarrow \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix} - 0.6 \cdot \frac{1}{3} \left(\begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} + \begin{pmatrix} -1.25 \\ 0.625 \\ 1.25 \end{pmatrix} + \begin{pmatrix} 0.5 \\ 1.0 \\ 0.25 \end{pmatrix} \right) \\ &\leftarrow \begin{pmatrix} 0.15 \\ 0.175 \\ 0.7 \end{pmatrix} \end{aligned}$$

Second Iteration

$$\mathbf{w} = \begin{pmatrix} 0.15 \\ 0.175 \\ 0.7 \end{pmatrix}, \alpha = 0.4$$

	Instance 1	Instance 2	Instance 3
$\hat{y}$ or $\mathbf{w}^T\mathbf{x}$	1.375	-0.6375	0.85
$\hat{y} - y$	-0.625	-0.6375	-0.15
$\left(\hat{y}^{(i)} - y^{(i)}\right) \cdot \mathbf{x}$	$\begin{pmatrix} -0.625 \\ -0.625 \\ -0.9375 \end{pmatrix}$	$\begin{pmatrix} -0.6375 \\ 0.31875 \\ 0.6375 \end{pmatrix}$	$\begin{pmatrix} -0.15 \\ -0.3 \\ -0.075 \end{pmatrix}$

Accumulate and averaging:

$$\begin{aligned} \mathbf{w} &\leftarrow \mathbf{w} - \alpha \cdot \frac{1}{3} \sum_{i=1}^{3} \left(\hat{y}^{(i)} - y^{(i)}\right) \cdot \mathbf{x} \\ &\leftarrow \begin{pmatrix} 0.15 \\ 0.175 \\ 0.7 \end{pmatrix} - 0.4 \cdot \frac{1}{3} \left(\begin{pmatrix} -0.625 \\ -0.625 \\ -0.9375 \end{pmatrix} + \begin{pmatrix} -0.6375 \\ 0.31875 \\ 0.6375 \end{pmatrix} + \begin{pmatrix} -0.15 \\ -0.3 \\ -0.075 \end{pmatrix} \right) \\ &\leftarrow \begin{pmatrix} 0.338\dot{3} \\ 0.2558\dot{3} \\ 0.7500 \end{pmatrix} \approx \begin{pmatrix} 0.3383 \\ 0.2558 \\ 0.7500 \end{pmatrix} \end{aligned}$$

Answer:
- Initially, $\mathbf{w} = [\theta_0, \theta_1, \theta_2] = [0, 0.5, 1]$
- $\mathbf{w}$ After 1st iteration: $[\mathbf{0.1500, 0.1750, 0.7000}]$
- $\mathbf{w}$ After 2nd iteration: $[\mathbf{0.3383, 0.2558, 0.7500}]$

Solution to Problem 2

True Positive (TP): samples correctly classified False Positive (FP): samples falsely labeled as positive. False Negative (FN): samples incorrectly classified as negative.

$$\text{TP} = 600 \text{ (Given)}$$
$$\text{FP} = 900 - 600 = 300,$$
$$\text{FN} = 800 - 600 = 200,$$

Let Recall be R and Precision be P. Then:

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{3}{4} = 0.75 \text{ or } 75\%$$
$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{2}{3} \approx 0.667 \text{ or } 66.7\%$$
$$F_1 = \frac{2PR}{P + R} = \frac{12}{17} \approx 0.706 \text{ or } 70.6\%$$

Solution to Problem 3

Explanation

The question confused between: *independence of data-points* and *independence of datasets.*

The idea that "test date should be independent of training data" is to avoid *leakage* by *independence of datapoints.* Whereas the *independence of dataset* is not pursued as it would mean training and testing the model on two different tasks, which does not logically make sense.

To prevent *leakage*, the training and test sets must be statistically independent and identically distributed (i.i.d.). This means: 1) Test set is a disjoint set of sample from the train set. 2) No information from the test set (e.g., its statistics, labels) is used during training (by independence of samples).

If leakage occurs, the model gains unfair knowledge about the test data, causing unreliable performance estimates.

Mathematical Formulation

Let the entire dataset be dataset be

$$\mathcal{D} = \{(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)\} \overset{\text{i.i.d}}{\sim} P(X, Y)$$

where $P(X, Y)$ is the true underlying data-generating distribution. Then, to satisfiy *independence of data-points*, the training and testing sets must satisfy the folloiwng properties:

1. Disjointness $\mathcal{D}^{(i)}_{\text{train}} \cap \mathcal{D}^{(i)}_{\text{test}} = \emptyset, \forall i$
2. Independence $P\left(X_i, Y_i, X_j, Y_j\right) = P(X_i, Y_i)P\left(X_j, Y_j\right), \forall i, j$

Final Verdict

1. Cross-validation splits the dataset into disjoint folds, satisfying the *disjointness.*
2. Random selection ensures that each fold is an independent sample from the same underlying distribution (i.i.d), which ensures that no additional information is leaked between sets, thereby satisfying the *independence.*

Therefore cross-validation w/ random selection successfully achieved *independence of data-points.* The original question is the <u>**result of a misunderstanding of *independence* in the context of test-train splits**</u>.

Solution to Question 4

Below is a step-by-step update of $\mathbf{w}$ given the instances, based on the update rule. (See **Appendix** at the bottom of this document for more detail)

Initialization

Let's start by padding the instances with a bias term, $x_0 = 1$:

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}. \text{ Instance 1, 2, 3} = \begin{pmatrix} 1 \\ 1 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 1 \\ -0.5 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 0.5 \end{pmatrix}$$

Update Rule:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha(\sigma(z) - y) \cdot \mathbf{x}$$

First Iteration

$$\mathbf{w} = \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix}, \alpha = 0.6$$

	Instance 1	**Instance 2**	**Instance 3**
z or $\mathbf{w}^T\mathbf{x}$	2	-1.25	1.5
$\sigma(z)$	0.8808	0.2227	0.8176
$\sigma(z) - y$	-0.1192	0.2227	-0.1824
$(\sigma(z) - y) \cdot \mathbf{x}$	$\begin{pmatrix} -0.1192 \\ -0.1192 \\ -0.1788 \end{pmatrix}$	$\begin{pmatrix} 0.2227 \\ -0.1114 \\ -0.2227 \end{pmatrix}$	$\begin{pmatrix} -0.1824 \\ -0.3648 \\ -0.0912 \end{pmatrix}$

Accumulate and averaging:

$$\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} - \alpha \cdot \frac{1}{3} \sum_{i=1}^{3} \left(\sigma(z)^{(i)} - y^{(i)}\right) \cdot \mathbf{x} \\
&\leftarrow \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix} - 0.6 \cdot \frac{1}{3} \left(\begin{pmatrix} -0.1192 \\ -0.1192 \\ -0.1788 \end{pmatrix} + \begin{pmatrix} 0.2227 \\ -0.1114 \\ -0.2227 \end{pmatrix} + \begin{pmatrix} -0.1824 \\ -0.3648 \\ -0.0912 \end{pmatrix} \right) \\
&\leftarrow \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix} - \begin{pmatrix} -0.0158 \\ -0.1191 \\ -0.0985 \end{pmatrix} = \begin{pmatrix} 0.0158 \\ 0.6191 \\ 1.0985 \end{pmatrix}
\end{aligned}$$

Second Iteration

$$\mathbf{w} = \begin{pmatrix} 0.0158 \\ 0.6191 \\ 1.0985 \end{pmatrix}, \alpha = 0.4$$

	Instance 1	**Instance 2**	**Instance 3**
z or $\mathbf{w}^T\mathbf{x}$	2.283	-1.3923	1.8033
$\sigma(z)$	0.9075	0.1990	0.8586
$\sigma(z) - y$	-0.0925	0.1990	-0.1414
$(\sigma(z) - y) \cdot \mathbf{x}$	$\begin{pmatrix} -0.0925 \\ -0.0925 \\ -0.1388 \end{pmatrix}$	$\begin{pmatrix} 0.1990 \\ -0.0995 \\ -0.1990 \end{pmatrix}$	$\begin{pmatrix} -0.1414 \\ -0.2828 \\ -0.0707 \end{pmatrix}$

Accumulate and averaging:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \frac{1}{3} \sum_{i=1}^{3} \left(\sigma(z)^{(i)} - y^{(i)}\right) \cdot \mathbf{x}$$

$$\leftarrow \begin{pmatrix} 0.0158 \\ 0.6191 \\ 1.0985 \end{pmatrix} - 0.4 \cdot \frac{1}{3} \left(\begin{pmatrix} -0.0925 \\ -0.0925 \\ -0.1388 \end{pmatrix} + \begin{pmatrix} 0.1990 \\ -0.0995 \\ -0.1990 \end{pmatrix} + \begin{pmatrix} -0.1414 \\ -0.2828 \\ -0.0707 \end{pmatrix} \right)$$

$$\leftarrow \begin{pmatrix} 0.0158 \\ 0.6191 \\ 1.0985 \end{pmatrix} - \begin{pmatrix} -0.0047 \\ -0.0633 \\ -0.0545 \end{pmatrix} = \begin{pmatrix} 0.0205 \\ 0.6824 \\ 1.1530 \end{pmatrix}$$

Answer:
- Initially, $\mathbf{w} = [\theta_0, \theta_1, \theta_2] = [0, 0.5, 1]$
- $\mathbf{w}$ After 1st iteration: $[\mathbf{0.0158, 0.6191, 1.0985}]$
- $\mathbf{w}$ After 2nd iteration: $[\mathbf{0.0205, 0.6824, 1.1530}]$

(See Next Page)

Appendix

Reformulation

The Update Rule of $\mathbf{w}$ is given as:

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial \mathbf{w}}$$

$$\because \frac{\partial z}{\partial \mathbf{w}} = \mathbf{x}$$

$$\therefore \frac{\partial L}{\partial \mathbf{w}} = \left(\frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z}\right) \cdot \mathbf{x} = \delta \cdot \mathbf{x}$$

Therefore, the **Generic Update Rule** can be written as,

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha(\delta \cdot \mathbf{x})$$

Linear w/ MSE Loss *(for problem 1)*

In the case of linear regression w/ MSE Loss in *problem (1)*,

$$\hat{y} = z = \mathbf{w}^T \mathbf{x} + \theta_0$$

$$L(y, \hat{y}) = (\hat{y} - y)^2$$

$$\delta = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} = \frac{\partial L}{\partial \hat{y}} \cdot 1 = 2(\hat{y} - y)$$

Weight Update *(everything together)*:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha 2(\hat{y} - y) \cdot \mathbf{x}$$

Sigmoid w/ BCE Loss *(for problem 4)*

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$L(y, \hat{y}) = -(y \cdot \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

Derivatives:

$$\text{Loss}: \frac{\partial L}{\partial \hat{y}} = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right) = \frac{\hat{y}-y}{\hat{y}(1-\hat{y})}$$

$$\text{Activation}: \frac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y}) \textbf{ or } \sigma(z)(1-\sigma(z))$$

$$\text{Error Term}: \delta = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} = \frac{\hat{y}-y}{\hat{y}(1-\hat{y})} \cdot \hat{y}(1-\hat{y})$$

$$= \hat{y} - y \textbf{ or } \sigma(z) - y$$

Weight Update *(everything together)*:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha(\sigma(z) - y) \cdot \mathbf{x}$$

Batch Gradient Descent

We only need to consider $(y - \hat{y})^2$ when calculating the derivatives of MSE. There is no need to calculate the derivative of $\frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$, until the very last step.

Intuitively, this is because n samples $\rightarrow$ n different $\hat{y}$ $\rightarrow$ n different gradients for each weight down the chain rule, which allows us to compute them individually and only summing and averaging them at the last step.Mathematically, simply move the constant term $\frac{1}{n}$ outside the derivative.

TL;DR:
For any loss function L in the form of $\frac{1}{n}\sum f(\hat{y}_i, y_i)$, we can compute each $f(\hat{y}_i, y_i)$ individually. Then, all we need to do is to sum up the derivatives of each sample and average the results at the very last step. This simplifies calculation.