Week 04 Quiz: Logistic Regression Solutions

Name SID

December 11, 2020

Exercise 1

This question is to ensure a mathematical understanding of how the loss function leads to training a logistic regression model. Complete this section without referring to the notes.

1. Write down the cross entropy loss for logistic regression.

   Solution:

$$L(\vec{\theta}, X, \vec{y}) = \frac{1}{n}\sum_{i=1}^{n} -y_i \ln s_i - (1 - y_i)\ln(1 - s_i)$$

   where s(·) is the sigmoid function, and

$$s_i = s(X_i^T \vec{\theta})$$

2. Now take the derivative of this loss with respect to the weights $\vec{\theta}$.

   Solution:

$$\begin{aligned}
\nabla_{\vec{\theta}} L(\vec{\theta}, X, \vec{y}) &= \frac{1}{n}\sum_{i=1}^{n} -\frac{y_i}{s_i}\nabla_{\vec{\theta}} s_i + \frac{1 - y_i}{1 - s_i}\nabla_{\vec{\theta}} s_i \\
&= -\frac{1}{n}\sum_{i=1}^{n} (\frac{y_i}{s_i} - \frac{1 - y_i}{1 - s_i})\nabla_{\vec{\theta}} s_i \\
&= -\frac{1}{n}\sum_{i=1}^{n} (\frac{y_i}{s_i} - \frac{1 - y_i}{1 - s_i}) s_i(1 - s_i) X_i \\
&= -\frac{1}{n}\sum_{i=1}^{n} (y_i(1 - s_i) - (1 - y_i)(s_i)) X_i \\
&= -\frac{1}{n}\sum_{i=1}^{n} (y_i - s_i) X_i
\end{aligned}$$

Exercise 2

In this exercise, you will compute the predictions and the cross entropy loss of a logistic regression model by hand, so that you will get familiar with the definitions and properties of logistic regression.

Suppose we are given the data set $\{((x_i, z_i), y_i)\}$ for $i = 1, ..., 5$. There are 5 data points in total, each data point has a feature vector $(x, z) \in \mathbb{R}^2$ and label value $y \in \{0, 1\}$.

x_i	-5	-4	-3	-2	-1
z_i	1	2	3	4	5
y_i	0	0	1	0	1

Suppose our logistic regression model for binary classification is given by

$$f_\theta(x, z) = \frac{1}{1 + \exp\{-(\theta_1 x + \theta_2 z)\}}$$

, where $\theta_1 = -1, \theta_2 = 2$. You should remember from the notes that there's no closed form formula to optimize the model parameter θ. Now we want to evaluate performance of this model on the given data set.

1. Compute predictions of this logistic regression model on each data point in this given data set. You may leave your answer as unsimplified fractions

   Solution:

$$f_\theta(x_1, z_1) = \frac{1}{1 + \exp\{-(5 + 2)\})} = \frac{1}{1 + e^{-7}}$$
$$f_\theta(x_2, z_2) = \frac{1}{1 + \exp\{-(4 + 4)\})} = \frac{1}{1 + e^{-8}}$$
$$f_\theta(x_3, z_3) = \frac{1}{1 + \exp\{-(3 + 6)\})} = \frac{1}{1 + e^{-9}}$$
$$f_\theta(x_4, z_4) = \frac{1}{1 + \exp\{-(2 + 8)\})} = \frac{1}{1 + e^{-10}}$$
$$f_\theta(x_5, z_5) = \frac{1}{1 + \exp\{-(1 + 10)\})} = \frac{1}{1 + e^{-11}}$$

2. Compute the cross entropy loss of this logistic regression model on the whole dataset. You may refer to your computed numerical answers in previous part.

   Solution:

$$L(\theta, (x, z), y) = \sum_{i=1}^{5} -y_i \log f_\theta(x_i, z_i) - (1 - y_i) \log(1 - f_\theta(x_i, z_i))$$

$$= -\log(1 - f_\theta(x_1, z_1)) - \log(1 - f_\theta(x_2, z_2)) - \log f_\theta(x_3, z_3) - \log(1 - f_\theta(x_4, z_4)) - \log f_\theta(x_5, z_5)$$

where $f_\theta(x_i, z_i)$ are values computed in the previous part.

Exercise 3

In this question we want to work with different evaluation metrics of binary classifiers. Suppose we have a binary classifier that classifies the given data set with 10 points in total as follows.

y_i	1	0	1	0	1	0	1	0	1	1
$\hat{y}_i$	0	0	1	0	0	1	1	0	1	0

1. Draw confusion matrix of this logistic regression model on the given dataset

   Solution:

   First we compute the exact counts of true positives, true negatives, false positives, and false negatives.

   $$\text{TP} = 3, \text{TN} = 3, \text{FP} = 1, \text{FN} = 3,$$

   Filling in the corresponding entries of the confusion matrix, we get

actual vs predicted	0	1
0	3	1
1	3	3

2. Find the accuracy, precision, recall, true positive rate, and false negative rate of this logistic regression model

   Solution:

   $$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of data points}} = \frac{6}{10}$$

   .

   $$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{3}{3+1} = \frac{3}{4}$$

   $$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{3}{3+3} = \frac{1}{2}$$

   $$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{1}{2}$$

   $$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{1}{1+3} = \frac{1}{4}$$

3. Comment on the performance of this classifier. Is it better or worse than a random guessing classifier? Justify your answer, or leave it blank if there's not enough information.

   Solution:

   We use ROC curve to evaluate performance of a classifier in binary classification problem. The larger the area under the curve of ROC (AUC-ROC), the better our classifier can distinguish positive case and begative case. ROC curve plots TPR vs FPR at different thresholds of a classifier, that is FPR is on x-axis and TPR is on y-axis.

   Recall we know a random guessing classifier has ROC curve given by $y = x$ line between 0 and 1 in the 2-dimensional plane. If ROC curve of our classifier is higher than $y = x$ line, then we conclude our binary classifier performs better than random guessing.

We have TPR = $\frac{1}{2}$ and FPR = $\frac{1}{4}$ of our classifier computer above. So its ROC curve must pass through this point. Notice $(\frac{1}{4}, \frac{1}{2})$ stands higher than $y = x$ line, so the curve we are looking at goes above the random guessing classifier baseline. Thus our given classifier performs better than random guessing.

Exercise 4

In this question, we'll compare the different classification models you have seen so far on linearly separable data.

1. First we consider perceptrons. Recall that perceptrons works as intended only on linearly separable data for binary classification problems.

   Describe the decision boundary found by perceptron algorithms on linearly separable data in binary classification. What are its properties and what's special?

   Solution:

   Since our data is linearly separable, there are infinitely many separating hyperplanes with $d-1$ dimensions in the d dimensional feature space. The perceptron algorithm doesn't have any heuristics to differentiate between them. It randomly selects a separating hyperplane that perfectly classifies training points with different labels.

   As you should remember from our discussion in the note, this will lead to generalization issues, since the decision boundary found by perceptrons is so different every time that the algorithm itself is not stable and reliable.

2. Second we discuss the hard-margin SVM on linearly separable data. Describe the decision boundary found by the hard-margin SVM on linearly separable data in binary classification. Compare its shape and properties to perceptrons. What are some advantages and disadvantages of hard-margin SVM?

   Solution:

   Hard-margin SVM does implement a heuristic rule to select one unqiue solution among an infinite number of possible decision boundaries. It selects the separating hyperplane the maximizes the margin between the boundary to the closest data point of each label. This is the distance from the separating hyperplane to the closest data point with each label, pointing in the direction perpendicular to the decision boundary.

3. Third consider logistic regression on linearly separable data. You should remember that the optimal weight vector of logistic regression minimizes the cross entropy loss on the training dataset.

   (a) For linearly separable data, what is the value of cross entropy loss? What can you say about the decision boundary found by a logistic regression model?

   Solution:

   Since our data is linearly separable, from what we know about perceptrons and hard-margin SVM, they can both find linear decision boundaries that perfectly classify the training set. Thus, for logistic regression with more advanced nonlinear decision boundary, it can fit an optimal weight vector such that cross entropy loss goes to 0. This is possible since cross entropy loss is 0 if and only if all data points are correctly classified.

   In this case, the decision boundary found by logistic regression also perfectly separates the data, and we can set a threshold for the probability (which is what we interpret the sigmoid function to output) that we want to use to make classifica-

tions. This threshold would then define a hyperplane in feature space that makes classifications.

(b) Logistic regression given by

$$f_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)}$$

is completely determined by weight vector θ. If you want to force cross entropy loss as close to the value you computed in part (a), what will the value of entries in optimal weight vector $\hat{\theta}$ be?

If we use stochastic gradient descent to iteratively approximate optimal θ, how will value of cross entropy loss change throughout the process?

Solution:

Our dataset is linearly separable, which means that the absolute values of components in $\vec{\theta}$ diverge to ∞, and the optimal cross entropy loss is 0. Suppose θ is the vector that makes the logistic regression model perfectly classify all data points. Then if you multiply θ by any positive scalar value c, the new logistic regression model parametrized by the new weight vector $c\theta$ still perfectly classifies the data. And this new model will incur a lower cross entropy loss because of the shape and properties of the log function.

A cross entropy loss of 0 can never actually be achieved, because $s(t) = \frac{1}{1+e^{-t}}$ never outputs exactly 0 or 1. There's so single value of θ that will "minimize" the cross entropy loss, because we can always pick a value of $\vec{\theta}$ that has an even smaller loss. However, if we perform gradient descent to approximate θ, gradient updates will bring this value arbitrarily close to 0 and it will ultimate converge to a value close enough to ∞ depending on our stopping criteria.

4. Describe why logistic regression can be understood as a "generalized linear classification model". Where does "linear model" come from? Why can it create "nonlinear decision boundaries"?

Solution:

Given feature vector $x \in \mathbb{R}^d$, logistic regression outputs a probability between 0 and 1 in following process. x is first transformed to a "score" given by linear transformation with model parameter θ: $\theta^T x$. This is where the linear model part of logistic regression comes from. Then we apply the sigmoid function on the raw score; the range of the sigmoid function ensures that our output prediction is always in $(0, 1)$. This sigmoid is a nonlinear function mapping, so this is why logistic regression can generate nonlinear decision boundary.

Exercise 5

Potpourri. Long answers aren't necessary for this part.

1. Is there a closed form solution to logistic regression like there is for linear regression?

   Solution:

   No, so we use gradient descent instead.

2. What's the name of the encoding we use for multi-class classification?

   Solution:

   One-hot encoding.

3. What is the function that generalises the sigmoid to multi-class classifcation problems?

   Solution:

   Softmax.