

Machine Learning

COMP 5630/ COMP 6630/ COMP 6630 - D01

Instructor: Dr. Shubhra (“Santu”) Karmaker
TA 1: Dongji Feng
TA 2: Souvika Sarkar
Department of Computer Science and Software Engineering
Auburn University
Fall, 2022

September 5, 2022

Assignment #4

Logistic Regression

Submission Instructions

This assignment is due Tuesday, September 22, 2022, at 11:59pm. Please submit your solutions via Canvas (<https://auburn.instructure.com/>). You should submit your assignment as a typeset PDF. Please do not include scanned or photographed equations as they are difficult for us to grade.

Late Submission Policy

The late submission policy for assignments will be as follows unless otherwise specified:

1. 75% credit within 0-48 hours after the submission deadline.
2. 50% credit within 48-96 hours after the submission deadline.
3. 0% credit after 96 hours after the submission deadline.

Tasks

1 General Questions About Logistic Regression [60 pts]

1. [10 Points] Explain why logistic regression is a discriminative classifier (as opposed to a generative classifier such as Naive Bayes).
2. [10 Points] Recall the prediction rule for logistic regression is if $p(y^j = 1|x^j) > p(y^j = 0|x^j)$, then predict 1, otherwise predict 0. What does the decision boundary of logistic regression look like? Justify your answer (e.g., try to write out the decision boundary as a function of w_0, w_1, w_2 and x_1^j, x_2^j).

3. In this question, we will derive the logistic regression algorithm (the M(C)LE and its gradient). For simplicity, we assume the dataset is two-dimensional. Given a training set $\{(x^i, y^i); i = 1, \dots, n\}$ where $x^i \in \mathbb{R}^2$ is a feature vector and $y^i \in \{0, 1\}$ is a binary label, we want to find the parameters \hat{w} that maximize the likelihood for the training set, assuming a parametric model of the form.

$$p(y = 1 | x; w) = \frac{1}{1 + \exp(-w_0 - w_1 x_1 - w_2 x_2)} = \frac{\exp(w_0 + w_1 x_1 + w_2 x_2)}{1 + \exp(w_0 + w_1 x_1 + w_2 x_2)}. \quad (1)$$

- (a) **[20 Points]** Below, we give a derivation of the conditional log likelihood. In this derivation, provide a short justification for why each line follows from the previous one.

$$\ell(w) \equiv \ln \prod_{j=1}^n p(y^j | x^j, w) \quad (2)$$

$$= \sum_{j=1}^n \ln p(y^j | x^j, w) \quad (3)$$

$$= \sum_{j=1}^n \ln \left(p(y^j = 1 | x^j, w)^{y^j} p(y^j = 0 | x^j, w)^{1-y^j} \right) \quad (4)$$

$$= \sum_{j=1}^n \left[y^j \ln p(y^j = 1 | x^j, w) + (1 - y^j) \ln p(y^j = 0 | x^j, w) \right] \quad (5)$$

$$= \sum_{j=1}^n \left[y^j \ln \frac{\exp(w_0 + w_1 x_1^j + w_2 x_2^j)}{1 + \exp(w_0 + w_1 x_1^j + w_2 x_2^j)} + (1 - y^j) \ln \frac{1}{1 + \exp(w_0 + w_1 x_1^j + w_2 x_2^j)} \right] \quad (6)$$

$$= \sum_{j=1}^n \left[y^j \ln \left(\exp(w_0 + w_1 x_1^j + w_2 x_2^j) \right) + \ln \left(\frac{1}{1 + \exp(w_0 + w_1 x_1^j + w_2 x_2^j)} \right) \right] \quad (7)$$

$$= \sum_{j=1}^n \left[y^j \left(w_0 + w_1 x_1^j + w_2 x_2^j \right) - \ln \left(1 + \exp(w_0 + w_1 x_1^j + w_2 x_2^j) \right) \right]. \quad (8)$$

Next, we will derive the gradient of the previous expression with respect to w_0, w_1, w_2 , i.e., $\frac{\partial \ell(w)}{\partial w_i}$, where $\ell(w)$ denotes the log likelihood from part 1. We will perform a few steps of the derivation, and then ask you to do one step at the end. If we take the derivative of Expression 8 with respect to w_i for $i \in \{1, 2\}$, we get the following expression:

$$\frac{\partial \ell(w)}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_{j=1}^n \left[y^j \left(w_0 + w_1 x_1^j + w_2 x_2^j \right) \right] - \frac{\partial}{\partial w_i} \sum_{j=1}^n \ln \left[1 + \exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right) \right]. \quad (9)$$

The blue expression is linear in w_i , so it can be simplified to $\sum_{j=1}^n y^j x_i^j$. For the red expression, we use the chain rule as follows (first we consider a single $j \in [1, n]$).

$$\frac{\partial}{\partial w_i} \ln \left[1 + \exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right) \right] \quad (10)$$

$$= \frac{1}{1 + \exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right)} \cdot \frac{\partial}{\partial w_i} \left(1 + \exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right) \right) \quad (11)$$

$$= \frac{1}{1 + \exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right)} \cdot \exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right) \frac{\partial}{\partial w_i} \left(w_0 + w_1 x_1^j + w_2 x_2^j \right) \quad (12)$$

$$= x_i^j \cdot \frac{\exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right)}{1 + \exp \left(w_0 + w_1 x_1^j + w_2 x_2^j \right)} \quad (13)$$

(b) [20 Points] Now, use Equation 13 (and the previous discussion) to show that overall, Expression 9, i.e., $\frac{\partial \ell(w)}{\partial w_i}$, is equal to

$$\frac{\partial \ell(w)}{\partial w_i} = \sum_{j=1}^n x_i^j (y^j - p(y^j = 1 \mid x^j; w)) \quad (14)$$

Hint: does Expression 13 look like a familiar probability?

Since the log likelihood is concave, it is easy to optimize using gradient ascent. The final algorithm is as follows. We pick a step size η , and then perform the following iterations until the change is $< \epsilon$:

$$w_0^{(t+1)} = w_0^{(t)} + \eta \sum_j \left[y^j - p(y^j = 1 \mid x^j; w^{(t)}) \right] \quad (15)$$

$$w_i^{(t+1)} = w_i^{(t)} + \eta \sum_j x_i^j \left[y^j - p(y^j = 1 \mid x^j; w^{(t)}) \right]. \quad (16)$$

2 Logistic Regression Implementation [40 pts]

In this assignment you will implement simple linear classifiers and run them on the following dataset:

Mushroom dataset: a simple categorical binary classification dataset. Please note that the labels in the dataset are 0/1, as opposed to -1/1 as in the lectures, so you may have to change either the labels or the derivations of parameter update rules accordingly.

The goal of this assignment is to help you understand the fundamentals of the classic logistic regression method and become familiar with scientific computing tools in Python. You will also get experience in hyperparameter tuning and using proper train/validation/test data splits.

Download the starting code from the package (“Logistic Regression.zip”) provided to you.

The top-level notebook (“Logistic.ipynb”) will guide you through all of the steps. Setup instructions are below. The format of this assignment is inspired by the Stanford CS231n assignments, and we have borrowed some of their data loading and instructions in our assignment IPython notebook.

None of the parts of this assignment require the use of a machine with a GPU. You should be able to complete the assignment using your local machine.

Environment Setup (Local): You will need a Python environment set up with the appropriate packages.

IPython: The assignment is given to you in the Logistic.ipynb file. ensure that IPython is installed (<https://ipython.org/install.html>). You may then navigate to the assignment directory in the terminal and start a local IPython server using the jupyter notebook command.

Reporting: Describe the hyperparameter tuning you tried for learning rate and number of epochs. Report the optimal hyperparameter setting you found in the list below. Also report your training, validation, and testing accuracy with your optimal hyperparameter setting.

- **Optimal hyperparameters:**
- **Training accuracy:**
- **Validation accuracy:**
- **Test accuracy:**

Also, create two plots as following:

1. Fix the optimal learning rate, then create plot where x axis varies the number of epochs and y-axis plots the training, validation, and testing accuracy.
2. Fix the optimal number of epochs, then create plot where x axis varies the learning rate and y-axis plots the training, validation, and testing accuracy.

Finally, submit “logistic.py” on Canvas as well.

Disclaimers: This assignment re-uses some materials from the publicly available website: [CMU Introduction to Machine Learning Course, 10-315, Spring 2019](#). I personally thank Prof. Maria-Florina Balcan for sharing her teaching materials publicly. This assignment is exclusively used for instructional purposes.

We also thank University of Illinois at Urbana-Champaign for developing this assignment. Credit goes to Justin Lizama (jlizama2@illinois.edu), Daniel Gonzales (dsgonza2@illinois.edu) and Weilin Zhang (weilinz2@illinois.edu).