Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining — Assignment 6

Due: March 15, 2023(100 points)

YOUR NAME
YOUR GIT USERNAME
YOUR E-MAIL

Gradient Descent - Logistic Regression

The sinking of the RMS Titanic is the most notorious shipwrecks in history, sinking in the early hours of April 15, 1912. During her maiden voyage, the Titanic was widely considered "unsinkable", but ultimately sank after colliding with an iceberg. In accordance with existing practice, Titanic's lifeboats did not account for all passengers onboard, and of the 2,224 passnegers, 1502 of them perished, most of whom were still on board when the ship sank. While seemingly random, it seems that some groups of people were more likely to survive than others.

In this homework question, which is mirrored on Data World (though the data samples are different)[1], we ask you to build a predictive model that answers the question: "what sorts of people were more likely to survive?" using passenger data (i.e., name, age, gender, socio-economic class, etc). It's important that you review the descriptions of the features on Data World Data World. Also, you can try getting hints on what features predicted survival from the Wikipedia page.

We'll be using logistic regression with binary cross-entropy cost function appearing in the following form:

$$\mathcal{L}(W, b) = -\sum_i y_i \log h_{W,b}(\mathbf{x}) + (1 - y_i) \log \left(1 - h_{W,b}(\mathbf{x})\right)$$

where

$$h_{W,b}(\mathbf{x}) = \sigma\left(W^T \mathbf{x} + b\right)$$

[1] This data has been circulated quite frequently in different iterations (with different data splits, and subsequently seen on Kaggle.

and

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

1. Download the data from the homework 5 data folder. Train a logistic regression predicting who would survive with `titanic.train.csv`. Test to see your accuracy on `titanic.test.csv`, where accuracy is defined by:

$$\text{acc} = \frac{\text{num correct}}{\text{total}} \tag{0.1}$$

   Make sure you're not including the `survived` column in the data as one of your features (that'd be cheating!) If you've done the lab in class, then this simply becomes an exercise of data transformations and preprocessing. (Since there is only one label that you are predicting, you may also have to ensure that the dimensions are correct.)

Some tips and tricks:

- For features with multiple values, try making one-hot encoded data. For example, number of siblings, you can try to have a five column array.

No Siblings	1 Sibling	2 Siblings	3 Siblings	4+ Siblings
0	1	0	0	0

- Play around with your learning rate. If your learning is too high or too low, then you won't converge to the right answer.
- You don't use all the features. Just because they're there, doesn't mean they're always useful. (I ignored using the names.)
- Make sure you normalize your features to have similar ranges. You needn't use the standard scalar (i.e., z-scores), but if you don't transform some features, the logistic regression will not work well.
- While Kaggle provides good descriptions of what the data is, do *NOT* use the Kaggle dataset. Only use data provided via course website.

See if you can do better than I can with a **logistic regression** by adopting the code from the lab in lecture. On the test set, with that code, I reached 79.2% accuracy at a threshold of 0.5, using just a few features.

For this question, submit a plot of your loss curve and print your overall accuracy on training and test sets. Include what features you used, what you did to transform them, and also include what you used as a learning rate. Submit your code to Gradescope.

2. Using a toolbox (like Keras), try benchmarking your logistic regression and then adding a few more layers (i.e., a neural network). Diagram your neural network architecture and print out your overall **test set** accuracy. Make sure you do *not* train on any samples from your test set.