Tyler Cerilli

EECE 5644

Homework 2

Github: https://github.com/tpcerilli/EECE-5644

**Problem 1**

np.random.seed(1)

Number of samples from Class 1: 16, Class 2: 4

Number of samples from Class 1: 127, Class 2: 73

Number of samples from Class 1: 1309, Class 2: 691

Number of samples from Class 1: 6391, Class 2: 3609

Chart, scatter chart

Description automatically generated

**Part 1**

Threshold value: 1.8571428571428574

Confusion Matrix MAP (rows: Predicted class, columns: True class):

[[5634 908]

[ 757 2701]]

Total Number of Misclassified Samples (MAP): 1665

Gamma MAP (Theoretical): 1.8571428571428574

Probability of Error(MAP): 0.16649999999999998

Best Gamma (ERM): 1.649473324917088

Probability of Error(Empirical): 0.1649

Chart, line chart

Description automatically generatedChart, scatter chart

Description automatically generated

**Part 2**

LINEAR

2 batches of size 10:

Logistic-Linear N=20 GD Theta:

[-0.02831339 -0.50197184 -0.09912327]

Logistic-Linear N=20 NLL: 1.2686217324687203

The total error achieved with this classifier is 0.442

20 batches of size 10:

Logistic-Linear N=200 GD Theta:

[-0.87731068 0.18788255 0.09950977]

Logistic-Linear N=200 NLL: 12.60622593092242

The total error achieved with this classifier is 0.367

200 batches of size 10:

Logistic-Linear N=2000 GD Theta:

[-1.90320918 0.37300914 0.3563989 ]

Logistic-Linear N=2000 NLL: 121.38572234930577

The total error achieved with this classifier is 0.346

A picture containing application

Description automatically generated

QUADRATIC

2 batches of size 10:

Logistic-Linear N=20 GD Theta:

[ 0.08970478 -0.81070515 -1.2917546 -0.01122395 1.0294441 -0.0422405 ]

Logistic-Linear N=20 NLL: 0.4579893514019052

The total error achieved with this classifier is 0.276

20 batches of size 10:

Logistic-Linear N=200 GD Theta:

[ 0.35070649 -0.18782312 -0.47789395 -0.19579952 0.92950008 -0.20815267]

Logistic-Linear N=200 NLL: 8.668329342392768

The total error achieved with this classifier is 0.196

200 batches of size 10:

Logistic-Linear N=2000 GD Theta:

[-1.27830878 0.72130575 0.8245824 -0.32236347 0.55910594 -0.36879039]

Logistic-Linear N=2000 NLL: 81.01347096977517

The total error achieved with this classifier is 0.164

A picture containing diagram

Description automatically generated

How does the performance of your classifiers trained in this part compare to each other considering differences in number of training samples and function form? How do they compare to the theoretically optimal classifier from Part 1? Briefly discuss results and insights.

Let us look at just the quadratic results, as they are the best, and the same concepts apply to both the quadratic and linear case…

The number of training samples increase the performance of the classifiers drastically. From 20 to 200 to 2000 training samples validated on 10000 samples, the probability of error (performance) decreases from 0.276 (Ntrain =20) to 0.196 (Ntrain =200) to 0.164 (Ntrain=2000). Since we want to mitigate error, a decrease in error is proportional to an increase in performance. The same exact pattern/trend exists in the linear function form as well.

The function form also has a drastic effect and is an especially important component to consider. For both the linear and quadratic case, the best performance is achieved using the highest training set (Ntrain=2000). The linear form achieves a probability of error of 0.346, while the quadratic achieves a probability of error of 0.164. Clearly the quadratic case outperforms the linear case. This is the same across all sizes of training samples tested.

In comparison to the theoretically optimal classifier from Part 1 (p(error) = 0.16649999999999998), the linear form performs much worse, whereas the quadratic form performs better (for this seed). The fits and classifies the data just as well as the theoretically optimal classifier when we have sample set of ~2000 samples.

**Problem 2**

np.random.seed(7)

10 batches of size 10:

theta start:

[ 0.20139884 0.80473259 -0.30712783 -0.19619016 -0.69316614 1.56372702

0.3546326 -0.36475778 -0.32149488 0.10854485]

theta MLE:

[ 0.20269365 0.79902827 -0.30291402 -0.00566083 -0.65097462 1.56869389

-0.01630005 0.02123632 -0.27712615 0.12454508]

theta MAP:

[-0.3993506 0.03506325 0.2527343 -0.00449423 -0.01386988 0.60370334

-0.01026672 -0.00096463 -0.0428286 -0.1144727 ]

MSE ML (GD): 40.81441651726255

MSE MAP: 5.583247585813262

**Chart, scatter chart

Description automatically generated**

**Chart

Description automatically generated**

How does your MAP-trained model perform on the validation set as γ is varied? How is the MAP estimate related to the ML estimate? Describe your experiments, visualize, and quantify your analyses (e.g. average squared error on validation dataset as a function of hyperparameter γ) with data from these experiments.

As gamma in the MAP model increases, the MSE decreases rapidly until about ~ gamma = 50-100 where it flattens out. The curve seems to follow a somewhat exponential decay, especially when we take a closer look at it.

The MAP estimate is related to the ML estimate such that the MSE of the ML estimate is theoretically converging to the MSE point for MAP.

Chart

Description automatically generated

**APPENDIX:** Github: https://github.com/tpcerilli/EECE-5644