Midterm

May 15, 2020

Submission for: Anmol Srivastava

Exercise 1

Gradient Computation

When $n, d = 1$, the objective function and its gradient are:

$$F(\beta) = \exp(-yx\beta) + \lambda\beta^2 \tag{1}$$

$$\nabla F(\beta) = -yx\exp(-yx\beta) + 2\lambda\beta \tag{2}$$

We expand to the $n, d > 1$ case:

$$F(\beta) = \frac{1}{n}\sum_{i=1}^{n} \exp(-y_i x_i^T \beta) + \lambda\|\beta\|_2^2 \tag{3}$$

Using the linearity of differentiation, and other properties, we have:

$$\nabla F(\beta) = \frac{1}{n}\sum_{i=1}^{n} \nabla(\exp(-y_i x_i^T \beta)) + \nabla(\lambda\|\beta\|_2^2) \tag{4}$$

By chain rule and properties of vector differentiation, we simplify:

$$\nabla F(\beta) = \frac{-1}{n}\sum_{i=1}^{n} (y_i x_i)\exp(-y_i x_i^T \beta) + 2\lambda\beta \tag{5}$$

Data Processing (see code)

Function Definitions (see code, hyperparamsearch is in later section)

Training ($\epsilon = 0.005, \lambda = 1$)

Please Note:

Error is reported below. As we can see, for a binary classification problem our error rates are high. An investigation of the code reveals that the primary issue is the amount by which the weights change in *myclassifier*.

Specifically, the algebraic terms in the algorithm resulted in extremely small changes in β from iteration to iteration. These negligible movements in β (on a 10^{-10} scale) meant that the objective function is not converging towards a minimum at a fast enough rate. So, when *myclassifier* terminates, even the training error will be high. It was difficult to resolve this coding issue, after multiple modifications to the algorithm (randomizing initial weights, etc.).

```
When lambda = 1,
Misclassification Error, Training Data: 34.58%
Misclassification Error, Testing Data: 33.27%
```

Grid-Search (λ)

We re-group the data and create the desired 60%-40%split between training and validation data. We then use grid search on the validation set, via a logarithmic scale for λ, and report our findings. Note that as before, the unresolved issue of minute changes in β hinder the performance of *myclassifier* and the resulting grid search.

```
Optimal value of lambda: 0.001. For this lambda,
Misclassification error is: 34.17%
Sensitivity (in the validation set) is: 54.50%
Specificity (in the validation set) is: 73.35%
```

Exercise 2

Contagion.

It appears impossible to estimate the sensitivity and specificity of the antibody test. Sensitivity is the “true positive rate,” given positive test results and total positive instances. Similarly, specificity is the “true negative rate” given negative test results and total negative instances. To estimate both, it is necessary to know which samples were “actually” positive (had antibodies). We are given only the samples which “tested” positive (and by extension, those which tested negative). Without the “true” counts of total positive or negative instances, Sharon and Billie will be unable to determine the specificity or sensitivity accurately.

Flying saucers.

With UFOs, we must consider that there are a lot of legitimate objects in the sky that might not be discernible by the naked eye or image/video capture. Many planes (at various altitudes), birds, cloud formations, and light effects can be mistaken as UFOs (or deliberately distorted by conspiracy theorists). It is hence important to determine whether a detector is fake by its ability to correctly identify “negative” examples. That is, we care more about its ability to correctly yield negative results for the frequent non-saucers, than to “positively” classify the rare aliens. Even though we are in contact with an alien, the probabilistic likelihood of alien appearances has historically been low. Therefore, we use ***specificity*** (true negative rate over all negative samples) as the chief performance metric. Sensitivity may still have lesser value, as it is important to notice the rare occasion on which an alien is present.