HW2

Aarshay Jain (aj2713)

1 Problem 1 (Written)

1.1 MLE for π

Since only one part of the objective depends on π:

$$\begin{aligned}\hat{\pi} &= arg\max_{\pi}\sum_{i=1}^{n} lnp(y_i|\pi)\\ &= arg\max_{\pi}\sum_{i=1}^{n} y_i ln\pi + (1-y_i)ln(1-\pi)\end{aligned}$$

Taking the derivative and setting it to zero, we get the following solution:

$$\hat{\pi} = \frac{\sum_{i=1}^{n} y_i}{n}$$

1.2 MLE for θ_y^1

Since only one part of the objective depends on θ_y^1:

$$\begin{aligned}\hat{\theta_y^1} &= arg\max_{\theta_y^1}\sum_{i=1}^{n} lnp(x_{i1}|\theta_y^1)\\ &= arg\max_{\theta_y^1}\sum_{i=1}^{n} x_{i1}ln\theta_{yi}^1 + (1-x_{i1})ln(1-\theta_{yi}^1)\end{aligned}$$

Here for each i, θ_{yi}^1 can be either θ_{y0}^1 or θ_{y1}^1. So we can separate them using an indicator variable

$$= arg\max_{\theta_y^1}\sum_{i=1}^{n} x_{i1}ln\theta_y^1 + (1-x_{i1})ln(1-\theta_y^1)\ \mathbb{1}(y_i = y)$$

Taking the derivative and setting it to zero, we get the following solution:

$$\hat{\theta_y^1} = \frac{\sum_{i=1}^n x_{i1} \ \mathbb{1}(y_i = y)}{n_y}$$

where, $n_y = \sum_{i=1}^n \ \mathbb{1}(y_i = y)$

1.3 MLE for θ_y^2

Since only one part of the objective depends on θ_y^2:

$$\hat{\theta_y^2} = arg \max_{\theta_y^2} \sum_{i=1}^n lnp(x_{i2}|\theta_y^2)$$
$$= arg \max_{\theta_y^2} \sum_{i=1}^n ln\theta_{yi}^2 - (1 - \theta_{yi}^2) lnx_{i2}$$

Here for each i, θ_{yi}^2 can be either θ_{y0}^2 or θ_{y1}^2. So we can separate them using an indicator variable

$$= arg \max_{\theta_y^2} \sum_{i=1}^n ln\theta_y^2 - (1 - \theta_y^2) lnx_{i2} \ \mathbb{1}(y_i = y)$$

Taking the derivative and setting it to zero, we get the following solution:

$$\hat{\theta_y^2} = n_y \sum_{i=1}^n \frac{1}{lnx_{i2}} \ \mathbb{1}(y_i = y)$$

where, $n_y = \sum_{i=1}^n \ \mathbb{1}(y_i = y)$

2 Problem 2 (coding)

2.1 Naive Bayes 2x2 table

After running the algorithm, the accuracy and 2x2 table for the train and test data are shown in Figure 1.

		Actual	
		0	1
Predicted	0	54	5
	1	2	32

Figure 1: 2x2 table for naive bayes classifier on training and test set

Accuracy acheived: 92.47%

2.2 Naive Bayes weights chart

After running the algorithm, the step plot containing the weights for Bernoulli parameters is showin in Figure 2.

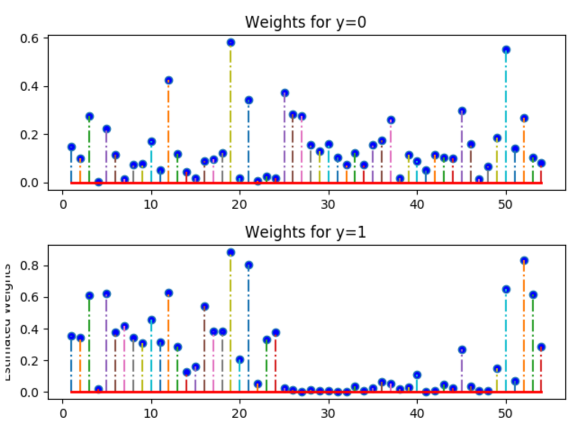

Figure 2: Bernoulli Parameters

Interpretation for dimensions 16 and 52:

- Dimension 16 represents presence of the word "free". The parameters show that the probability of an email being spam is about 5x when an email contains the word free as compared to when it doesn't contain that word. This is intuitive because most of the promotional emails would use the word free.

- Dimension 52 represents presence of the character "!". The parameters show that the probability of an email being spam is about 3x when an email contains an exclamation mark as compared to when it doesn't contain it. This might be because spam emails might use an ! to have an emotional appeal on the audience..

2.3 KNN Accuracy chart

After running the KNN algorithm, the accuracies found for k=1 through 20 are reported in Figure 3. Note that the if equal number of 0 and 1 are found, then a 0 is assigned because its the most prevelant class.

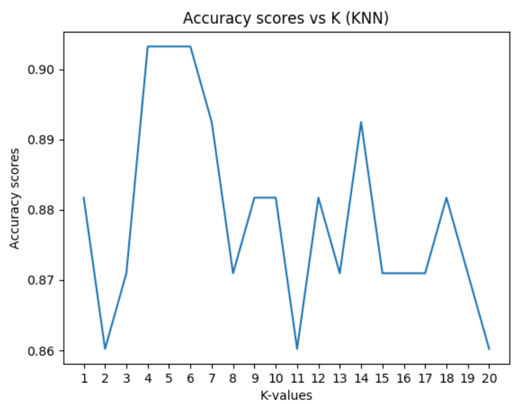

Figure 3: Accuracy vs k for KNN

Note: I have used np.argsort for sorting the distances so the choice of selecting a row is the distance is same is decided by that function by default. If a different function is used, then the results might varu slightly.

2.4 Logistic Regression Steepest Ascent

After running the logistic regression algorithm, the reduction in objective function found are shown in Figure 4.

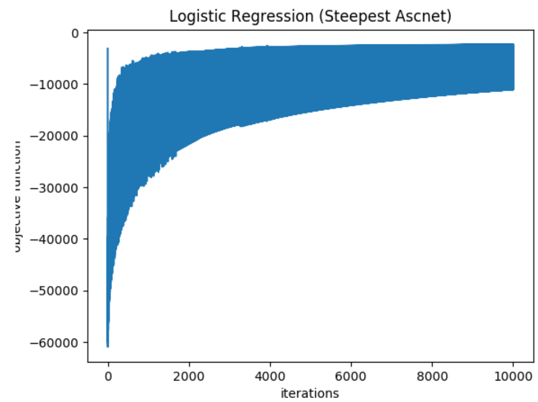

Figure 4: Objective reduction in logistic regression with steepest ascent

The prediction accuracy for test set using this method is: 68.82%.

2.5 Logistic Regression Newton Method

After running the logistic Newton method, the reduction in objective function found was shown 5.

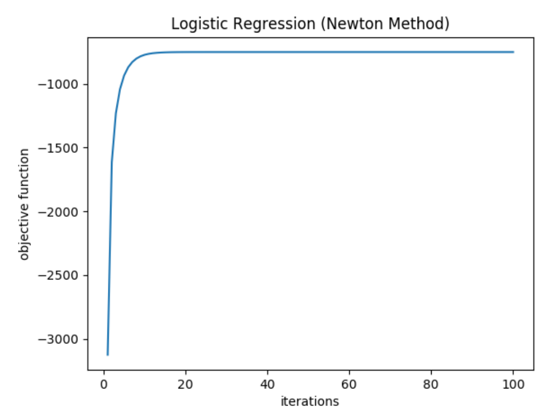

Figure 5: Objective reduction in logistic regression with Newton method

The prediction accuracy for test set using this method is: 91.4%.