Programming Assignment 2

Jayalakshmi Sureshkumar, Eihal Alowaisheq,Tan-Hsun Weng

**Answer 1.**

Learning rate: 0.01

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Observed** | **0** | **1** |
| **0** | 20 | 1 |
| **1** | 0 | 14 |

Accuracy: 97.14%

Learning rate: 0.005

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Observed** | **0** | **1** |
| **0** | 19 | 2 |
| **1** | 0 | 14 |

Accuracy: 94.28%

Learning rate: 0.005

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Observed** | **0** | **1** |
| **0** | 18 | 3 |
| **1** | 0 | 14 |

Accuracy: 91.42%

**Answer 2.**

|  |  |  |
| --- | --- | --- |
|  | **Predicted** | |
| **Observed** | **0** | **1** |
| **0** | 21 | 0 |
| **1** | 0 | 14 |

Accuracy: 100%

**Answer 3.**

Naïve Bayes performs better than Logistic regression in the given dataset,

That could be cause generally Naïve Bayes out performs logistic regression when the dataset is scarce. This could be because in Naïve Bayes the model is based on counting the examples while in logistic regression the weights are initialized to zero and then get updated periodically based on example it observes. Also, if there are missing values not seen in training data and seen in testing data, Naïve Bayes handles it with Laplace counting, while logistic regressions weighting system may not actually have a good weight on the unseen possible value (as in the case of feature 12).

**Answer 4.**

**Logistic regression**

In Weka, it shows that 94% accuracy rate, and 19 examples were classified as A, 14 examples were classified as B, and two were misclassified. In our algorithm, we got 97% accuracy rate when we set 0.01 as our learning rate, and 20 examples were classified as A, 14 examples were classified as B, and one was misclassified. We tried different learning rate for testing the accuracy, it shows that from 0.0001 to 0.009, we have lower accuracy. The difference results from we don’t know when the convergent condition stopped in Weka, and we don’t know the default learning rate in Weka.

With very low eta values we get high accuracy then when increasing the eta value “step size” the accuracy decreases, at this point behavior is as we expected. However, when we further increase the eta the accuracy increases again. This may indicate that the last increase in the accuracy is because the algorithm was lucky enough to hit the minima and learned weights that led to high accuracy.

|  |  |
| --- | --- |
| Weka | Macintosh HD:Users:tanhsunweng:Desktop:Screen Shot 2015-10-19 at 6.32.25 PM.png |
| Our algorithm | [jsureshk@degu LR]$ python LR.py **0.01** Trained the model with learning rate : 0.01 The confusion matrix is         PL1     PL2 TL1     20      1   TL2     0       14  Accuracy:  0.971428571429 [jsureshk@degu LR]$ python LR.py **0.005** Trained the model with learning rate : 0.005 The confusion matrix is         PL1     PL2 TL1     19      2   TL2     0       14  Accuracy:  0.942857142857 [jsureshk@degu LR]$ python LR.py **0.001** Trained the model with learning rate : 0.001 The confusion matrix is         PL1     PL2 TL1     18      3   TL2     0       14  Accuracy:  0.914285714286 [jsureshk@degu LR]$ python LR.py **0.0005** Trained the model with learning rate : 0.0005 The confusion matrix is         PL1     PL2 TL1     19      2   TL2     0       14  Accuracy:  0.942857142857 [jsureshk@degu LR]$ python LR.py **0.0003** Trained the model with learning rate : 0.0003 The confusion matrix is         PL1     PL2 TL1     20      1   TL2     0       14  Accuracy:  0.971428571429 |

**Naïve Bayes**

The results in Weka and our algorithm are the same. It shows that 100% accuracy, and 21 examples were classified as A, 14 examples were classified as B.

The results are the same, possibly because the method of computing the conditional probabilities and priors are the same. We are simply counting the values of the combinations of features and label, and training our model. Even the Laplace correction would do a similar sort of correction.

We do not have to iterate to fix some sort of parameter in Naïve Bayes as opposed to other classifiers like Logistic Regression, thus we are getting a similar result to Weka.

|  |  |
| --- | --- |
| Weka | Macintosh HD:Users:tanhsunweng:Desktop:Screen Shot 2015-10-19 at 6.52.33 PM.png |
| Our algorithm | [jsureshk@degu PA2]$ cd NB [jsureshk@degu NB]$ python NB.py The confusion matrix is         PL1     PL2 TL1     21      0   TL2     0       14  Accuracy:  1.0 |