2016

Assignment 2 Report

LINEAR MODELS FOR REGRESSION AND CLASSIFICATION

AARON (ARUNA) TILLEKERATNE - 27345483

Part A

Question 1

Section I

Given the gradient descent algorithms for linear regression (discussed in Chapter 2 of Module 2), derive weight update steps of stochastic gradient descent (SGD) as well as batch gradient descent (BGD) for linear regression with L2 regularisation norm. Show your work with enough explanation in your PDF report; you should provide the steps of SGD and BGD, separately.

Section I

Deriving the gradient of the training objective.

Training objective is:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \omega \cdot \phi(x_n)\right)^2 + \frac{\lambda}{2}\sum_{j=0}^{M-1}\omega_j^2$$

The gradient of the training objective is the derivative:

$$\nabla E(W) = -\sum_{n=1}^{N}[t_n - \omega \cdot \phi(x_n)]\phi(x_n) + \sum_{j=0}^{M-1}\lambda \cdot \omega_j$$

Implementation in the BGD Algorithm

1. Initialise the parameters to $\mathbf{w^{(0)}}$ and t = 1
2. While iterations do not exceed limit or the error is less than threshold error:
    - **a.** η' = η
    - **b.** while η' > ϵ
        - **i.** $w := w^{(t-1)} - \eta' * -\sum_{n=1}^{N}[t_n - \omega \cdot \phi(x_n)]\phi(x_n) + \sum_{j=0}^{M-1}\lambda \cdot \omega_j$
        - **ii.** if $E(w) < E(w^{(t-1)})$ then break
        - **iii.** η' = η' / 2
    - **c.** $w^{(t)} := w$
    - **d.** t = t + 1

Implementation in the SGD algorithm

1. Initialise the parameters to $\mathbf{w^{(0)}}$ and t = 1
2. While iterations do not exceed limit or the error is less than threshold error:
    - **a.** Randomly visit a data point (x_n, y_n) in the training set
    - **b.** $W^{(\tau)} := w^{(\tau-1)} - \eta^{(\tau)} \nabla E_n(w^{(\tau-1)})$
    - **c.** τ = τ+1

Section II

Please see iPython notebook for implementation.

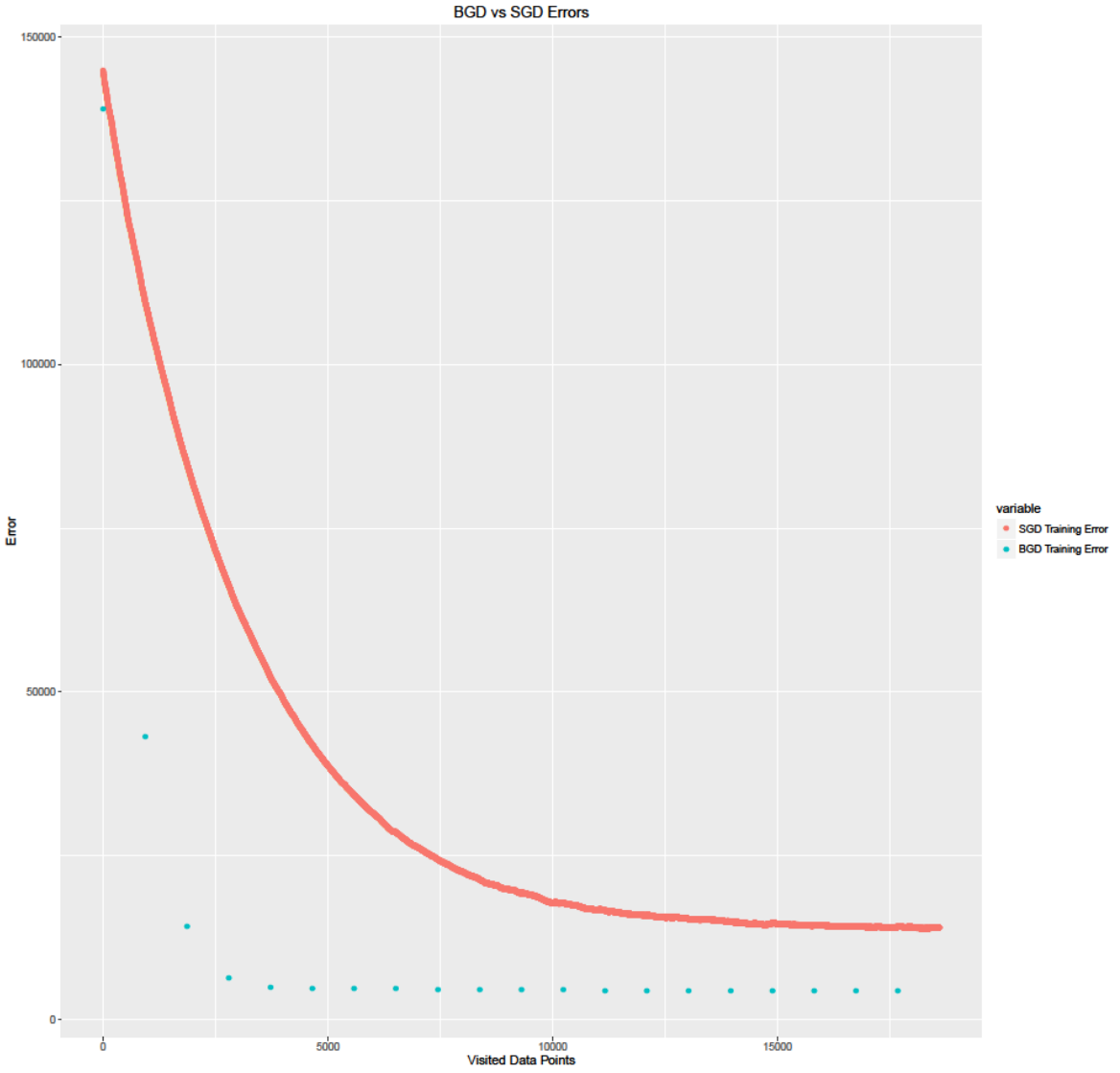
Section III

Subsection C

Run your implementations of SGD and BGD while all parameter settings (initial values, learning rate etc) are exactly the same for both algorithms. During run, record training error rate every time the weights get updated. Create a plot of error rates (use different colors for SGD and BGD), where the x-axis is the number of visited data points and y-axis is the error rate. Save your plot and attach it to

your report. Note that for every N errors for SGD in the plot, you will only have one error for BGD; the total length of the x-axis will be 20 x N.

The following error graph was recovered for BGD and SGD errors.



Subsection D

Explain your observation based on the errors plot you generated in Part C. Particularly, discuss the convergence speed and the fluctuations you see in the error trends.
The SGD seems to have a smooth convergence, with not much fluctuation up and down. It also seems to converge smoother than BGD with small reduction in error as the number of data points visited grows larger.

BGD on the other hand seems to respond longer to reduce error, however it was identified that the BGD was able to reduce error to a smaller value than achieved by SGD. BGD also seems to have fluctuated up and down before reaching a converged solution.

Part B

Question 2

Section 3

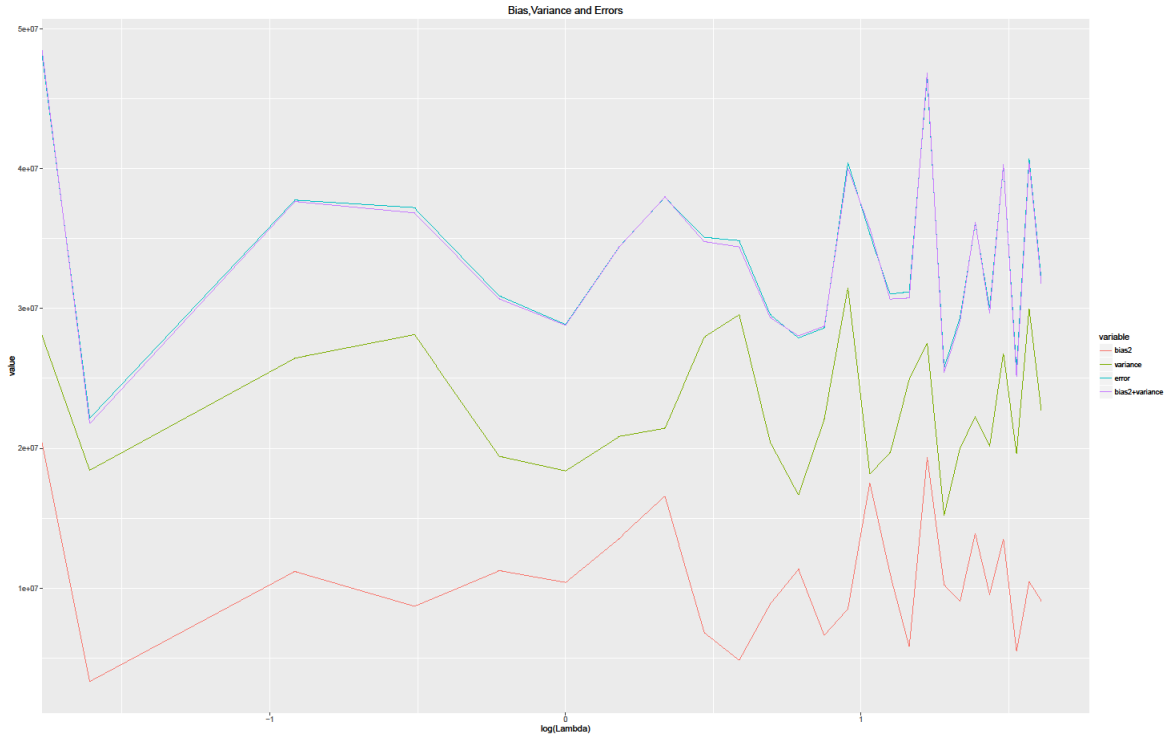
For each lambda in {0, 0.2, 0.4, 0.6, …, 5} do:

A. Build 50 regression models using the sampled sets

B. Based on the predictions of these models on the test set, calculate the (average) test error, variance, (bias) 2, and variance + (bias) 2.

Plot the (average) test error, variance, (bias) 2, and variance + (bias) 2 versus log lambda, and attach the plot to your report.

Plot of bias, variance, error and bias + variance.



Section 4

Based on your plot in the previous step (III), what's the best value for lambda?
Explain your answer in terms of the bias, variance, and test error.

The lowest points for bias and variance and was observed at λ = 0.2. This seems to be the point where bias2, variance, error and the bias2+variance terms seem to be lowest.

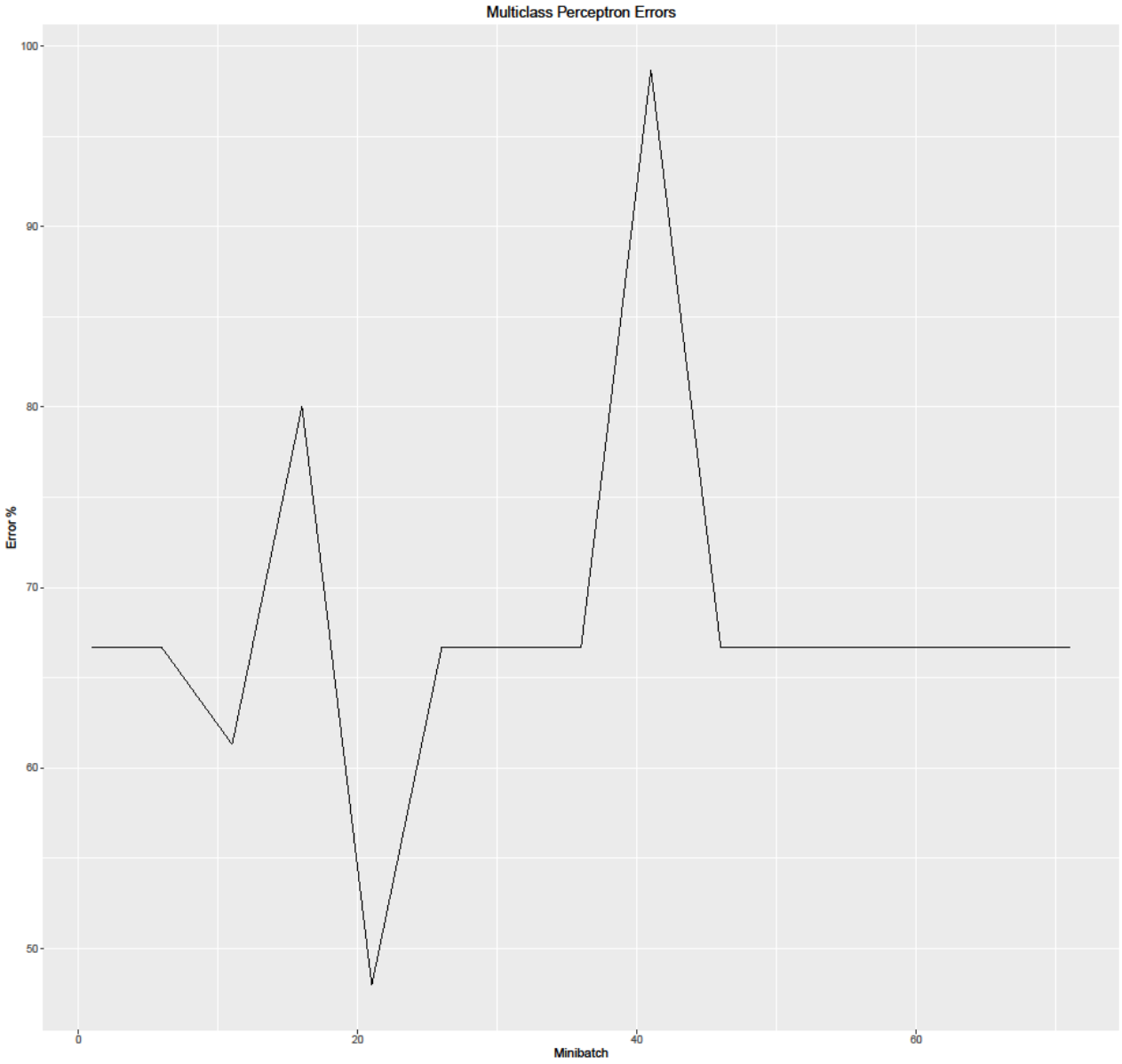
Part C

Question 3

Section 3

Set the learning rate to .1, and train the multiclass perceptron η on the provided training data. After processing every 5 training data points (also known as a mini-batch), evaluate the error of the current model on the test data. Plot the error of the test data vs the number of mini-batches, and attach the plot to your report.

Errors of multiclass perceptron.



Section 4

Suppose we did not want to use multiclass Perceptron, and instead would be interested to use the one-versus-one approach to solve the multi-class classification problem (Chapter 2 in Module 3). The

idea is to build K(K−1)/2 classifiers for each possible pair of classes where K is the number of classes. Each point is then classified according to a majority vote among the discriminant functions.

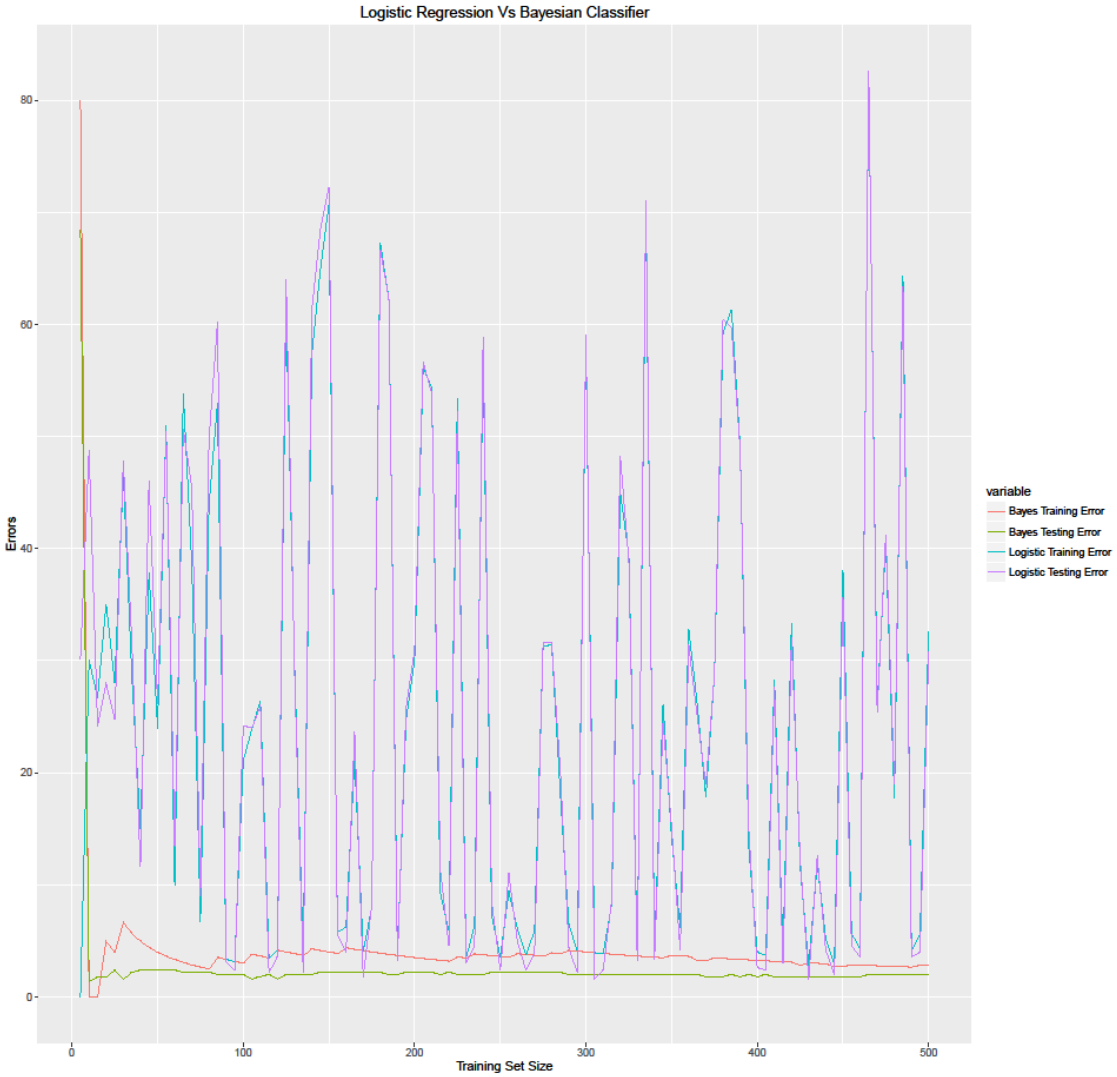
Confusion events could likely be caused by the errors in each model. Since each binary classifier has a chance to incorrectly classify labels, 2 of the 3 binary classifiers may return as they belong to their respective classes.

Part D

Question 4

Section 2

Using the first 5 data points from the training set, train a BC and a LR model, and compute their test errors. In a "for loop", increase the size of training set (5 data points at a time), retrain the models and calculate their test errors until all training data points are used. In one figure, plot the test errors for each model (with different colours) versus the size of the training set; name the plot "Learning Curve" and add it to your report.



Section 3

Explain your observations in your report:

A. What does happen for each classifier when the number of training data points is increased?

B. Which classifier is best suited when the training set is small, and which is best suited when the training set is big?

C. Justify your observations in previous questions (III.A & III.B) by providing some speculations and possible reasons.

A. Seems the Bayesian classifier improves well with the increasing training set size, whilst the logistic regression seems to be very erratic.
B. It seems that the logistic regression is better for smaller data sets and the Bayes classifier does well with larger number of points.
C. Perhaps the Logistic regression is more prone to overfitting and the distribution of data, where the Naïve Bayes is more resilient.