Homework Assignment # 4

Submitted by Vinay Vernekar (vrvernek@iu.edu)

Question 1. [30 marks]
In this question, you will implement a radial basis function network. The main difference is simply that the data is transformed using radial basis functions before being passed to the algorithms. Implement a radial basis transformation. Use Gaussian kernels, and try three different band-widths. Compare the performance on SUSY with the reduced feature set (8 features), comparing standard logistic regression and logistic regression with this radial basis transformation.

Answer:

Implemented RBF network function as “RBFLogitReg”. The function takes the initial matrix and transforms it based on the user given parameters and then passes this matrix to the Logistic regression.

A little theory about RBF: (not sure if this is needed or not, I have put this info, so I don’t lose any marks, Thanks)

A radial basis function, RBF, $\phi(x)$ is a function with respect to the origin or a certain point *c*, ie,

$$\phi(x)=f(\|x-c\|)$$

Where the norm is usually the Euclidean norm but can be other type of measure.

The RBF learning model assumes that the dataset $D=(xn,yn),n=1\ldots N$ influences the hypothesis set $h(x)$, for a new observation *x*, in the following way:

$$h(x)=\sum n=1Nwn\times exp(-\gamma\|x-xn\|2)$$

Which means that each *xi* of the dataset influences the observation in a Gaussian shape. Of course, if a data point is far away from the observation its influence is residual (the exponential decay of the tails of the Gaussian make it so).

The choice of **w** should follow the goal of minimizing the in-sample error of the dataset D, *Ein*(D), or simply *Ein* This means **w** should satisfy:

$$\sum_{m=1}^{N} w_m \times exp(-\gamma \|x_n - x_m\|^2) = y_n$$

for each data point *xn*∈D

There are *N* equations for *N* unknowns. In matrix form:

or simply: **w**=Φ−1*y* which is an exact interpolation of the dataset.

$$\underbrace{\begin{bmatrix} \exp(-\gamma\|x_1 - x_1\|^2) & \cdots & \exp(-\gamma\|x_1 - x_N\|^2) \\ \exp(-\gamma\|x_2 - x_1\|^2) & \cdots & \exp(-\gamma\|x_2 - x_N\|^2) \\ \vdots & \ddots & \vdots \\ \exp(-\gamma\|x_N - x_1\|^2) & \cdots & \exp(-\gamma\|x_N - x_N\|^2) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{y}} .$$

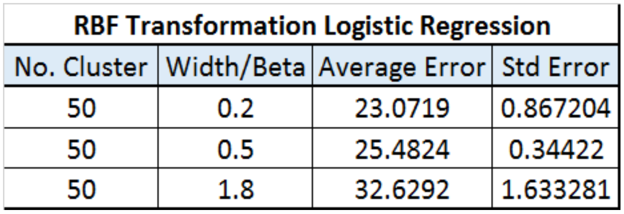
Notice now that Φ, is not a square matrix, so we need to compute the pseudo-inverse:

w=(Φ*T*Φ)−1Φ*Ty*

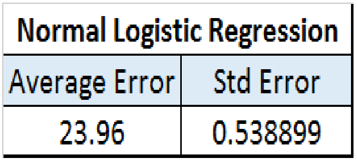
As we increase the number of clusters than the error reduces, however the computation time increases.

The performance of the above transformation is as below:

Different Center numbers were selected and different widths and the results are as per the table below



RBF Transformation Logistic Regression			
No. Cluster	Width/Beta	Average Error	Std Error
50	0.2	23.0719	0.867204
50	0.5	25.4824	0.34422
50	1.8	32.6292	1.633281



Normal Logistic Regression	
Average Error	Std Error
23.96	0.538899

Note: Number of runs were 15 for each setting. Also I had used K means form SKlearn and found that when K means is used for determining centers the results improve. The step size is 0.001 and the threshold for convergence is 0. 000001(1e-06) for Logistic Regression. And for RBF I have used a threshold of 0.00001(1e-05) with step size of 0.0001.

As the dimensions for RBF are high the step size has to be carefully selected for RBF.

****Please run “script_classify_Q1” for the first question.****

Observations:

We can see that as we increase the number of centers the accuracy increases for the RBF.
As we increase the variance or width the accuracy decreases.

Increasing the dimensions help improve the answer marginally, but is computationally expensive and step size needs to be carefully selected. It took many tries to understand this issue. With higher dimension and lower epsilon threshold, we can get appropriate answer, provided we take care of the step size.

Note: It takes a while for the RBF Logistic regression to give an output.

Question 2. [70 marks]
Implement any three learning methods (e.g., that you implemented from the last assignment) and run them on a dataset or problem of your choice. Use your knowledge about model comparison to formally conclude which of the two algorithms is better. This includes proper training-test splits, statistical significance tests, and proper meta-parameter selection techniques (e.g., cross-validation). You can use statistical significance tests built-in to python (or other languages). Provide a precise conclusion of your experiment. You can use any code and implementation you previously completed, but you are still prohibited from using packages that implement the learning algorithm directly. You can now use optimization software, such as lbfgs in scipy.

Answer:

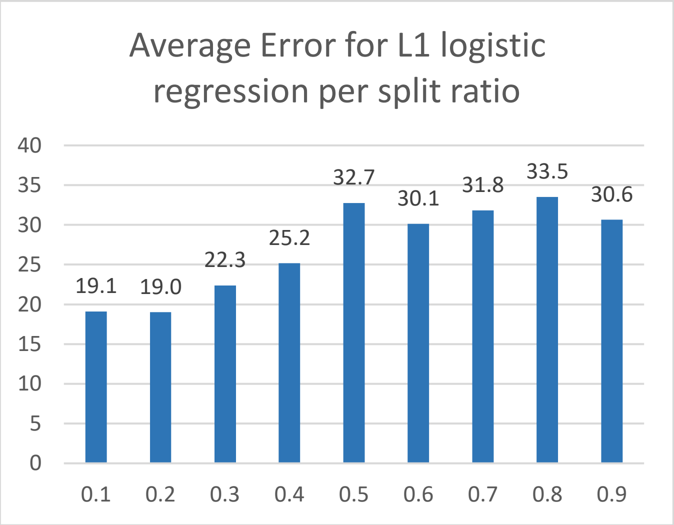
The data selected is from UCI machine learning repository. The data pertains to card payment default in Taiwan. The task is to define a potential clients as credible or not credible clients.

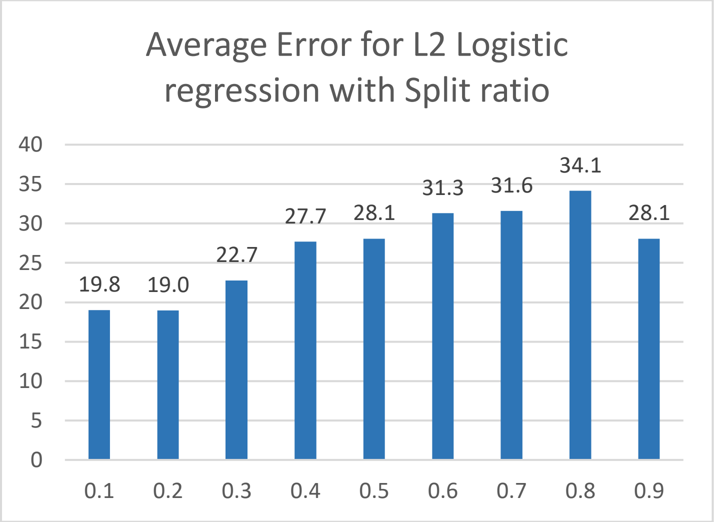
Data Description:

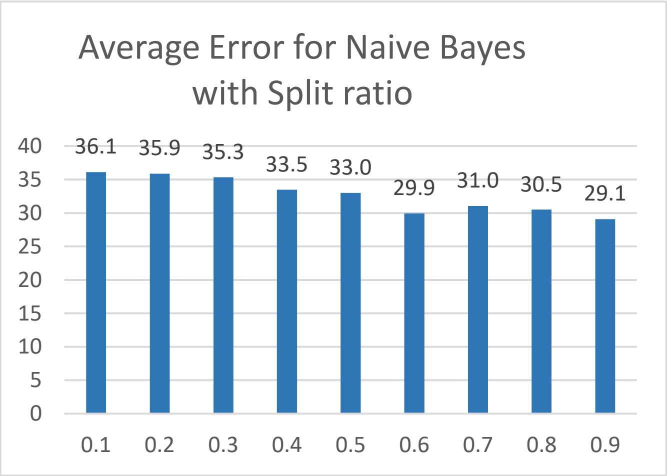
- Number of data points: 30,000
- Number of Features: 24
- The last column indicates default or No default encoded as 1 and 0 respectively

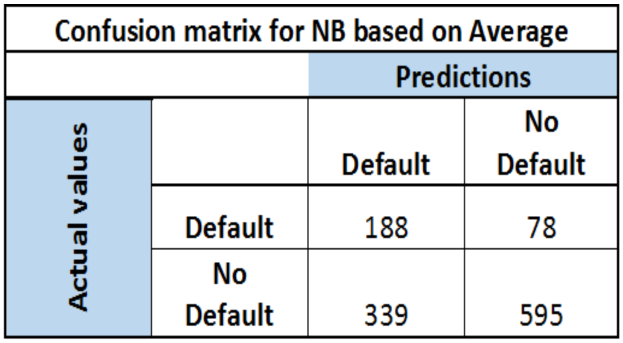
The three algorithms selected are Naïve Bayes (NB), L-2 Logistic regression (L2 or ridge) and L-1 Logistic regression (L1 or lasso).

1. Identifying the best split: To identify the best split various combination of split ratio (train: test) were tried for multiple iterations and below are the results









Confusion matrix for NB based on Average			
		Predictions	
Actual values		Default	No Default
	Default	188	78
	No Default	339	595

Note: The x –axis represents the train split.

Observations:

- Naïve Bayes requires a lot of data to train, before its accuracy increases
- For L2 and L1 logistic regression we can see that the best split is when we train the model on 20-25% of the data and test on rest of the data.

2. Selecting the Meta parameter: For this a range of parameter or tuning or penalty parameters were used to check the L2 and L1 logistic regressions performance. The range of tuning parameters value were generated by using np.logspace function

   After the parameters were generated a K fold -validation with reference to each tuning parameter is done and the average error is calculated. After this the two best algorithms are selected with reference to minimum error. The K size selected for this validation is 5.

   These algorithms mean error for each iterations was stored and now we compare these mean errors using scipy.stats.ttest_ind from Sklearn library. The error associated with each Meta parameters is checked and the min error associated with each Meta parameter value with reference to each algorithm is as follows.

   - The generalization error of L2 is 20.81666 with a penalty parameter of 0.21
   - The generalization error of L1 is 26.4033 with a penalty parameter of 4.6
   - The generalization error for Naïve Bayes is 29.6899

   Since Naïve Bayes results are not that good, we can compare L2 and L1 for T-test.

   E.g. scipy.stats.ttest_ind(means error list Alg1, means error list Alg2)

   This test returns the t-test value and the P-value. We then compare the P- value with the significance level of 0.05.

   Based on the results after K-fold validation testing, the two best algorithms based on their performance (accuracy) for the given data are, L1- Logistic regression and L2 Logistic regression. The performance of Naïve Bayes is less than these two algorithms. Hence L1 and L2 are selected for the T-test.

3. The results of the t-test are as follows:

   T-test = -1.0044

   P- Value = 0.3445

   Based on the rule “If the *P*-value is less than (or equal to) α, reject the null hypothesis in favor of the alternative hypothesis. If the *P*-value is greater than α, do not reject the null hypothesis.”

   The P-value is greater than significance level, hence we do NOT reject the Null Hypothesis

   These two algorithms are not that different statistically. But we can be advised to choose one which has less generalization error.

****Please run “script_classify_Q2” for the second question****

Thank You

References:

https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
https://docs.scipy.org/doc/scipy-0.18.1/reference/index.html
https://onlinecourses.science.psu.edu/statprogram/node/138
http://www.di.fc.ul.pt/~jpn/r/rbf/rbf.html