Spam Classification Using Logistic Regression, Ex.8.1

João Caetano, 75218

May 22, 2017

1 Introduction

This exercise focuses in fitting a Ridge Logistic Regression classifier to handle the email spam problem. It can be found in the book: Machine Learning, A Probabilistic Pespectic by Kevin Murphy. The Matlab code is provided in the Ex_8.1 folder.

2 Features Preprocessing

The program lets the user choose between not changing the input data and 3 different types of preprocessing.

2.1 Standardization

Every feature is standardized to have zero mean and unit variance. This is accomplished by dividing each element of the data by the total sum and by the standard deviation of the corresponding feature.

2.2 Logarithmic Transform

Each element of the data is set to the following logarithmic transformation:

$$log(x_{ij} + 0.1)$$

2.3 Conversion to Binary

Every element of the data is converted to 1 if it is bigger than 0, otherwise converted to 0.

3 Fitting the Ridge Logistic Regression classifier

Fitting the Ridge Logistic Regression classifier means solving an optimization problem (Fig. 1). This was done using the Newton algorithm with step progression as shown in Fig. 2.

$$\hat{\mathbf{w}}_{\mathsf{ridge}} = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Figure 1: Optimization problem to fit the classifier to the data

$$\mathbf{w}_{(t+1)} \leftarrow \mathbf{w}_{(t)} - \big(\mathbf{H}(\mathbf{w}_{(t)}) + \lambda\mathbf{I}\big)^{-1}\big(\mathbf{g}(\mathbf{w}_{(t)}) + \lambda\mathbf{w}_{(t)}\big)$$

Figure 2: Newton Algorithm step progression

It was done with a matricial approach instead of for loops for better performance.

4 Strength of l_2 regularizer

The strength of the l_2 regularizer (λ) is chosen using cross validation.

For that propose the training and test sets are portioned in N parts. In my tests N=10. One of the N parts is used for validation and the renaming ones for training.

The values I tested for λ are $10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}$.

For each λ the validation set shifts N times, passing through all N partitions. A model is fitted using the training sets and a misclassification error is calculated using the validation set. When all validation sets have been ran for a specific λ these errors are summed and averaged (divided by N). This way we have an average error for every strength of λ.

The graphs with the average errors for the three types of preprocessing are shown in Figs. 3, 4 and 5.

5 Fitting with the best λ

After picking the λ that provided the smallest error in the previous section I fitted a model now using the whole train set and calculated the mean error rate with the test set.

The mean error rates for the different preprocessings in the train and test set are shown in Table 1.

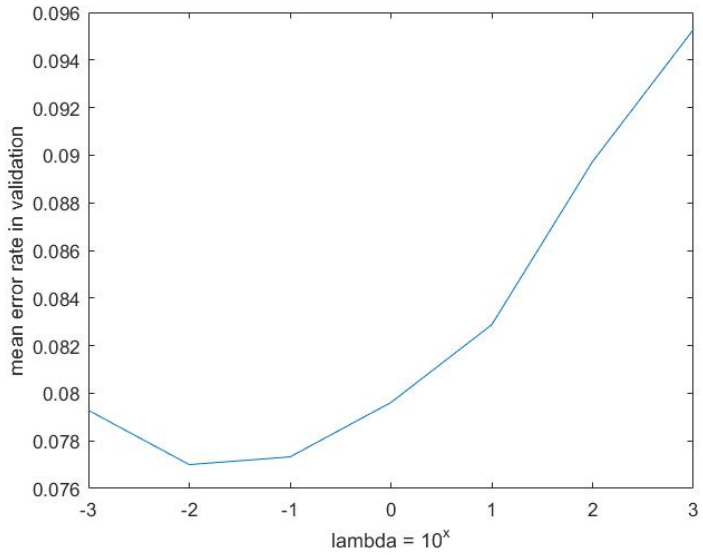

Figure 3: Standardization preprocessing - mean error rate for each λ. Best $\lambda = 10^{-2}$.

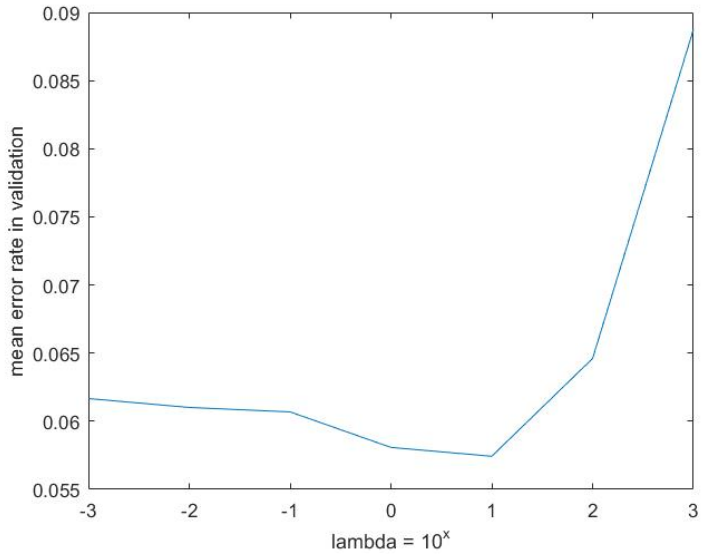

Figure 4: Log preprocessing - mean error rate for each λ. Best $\lambda = 10$.

method	train	test
stnd	0.0701	0.0865
log	0.0531	0.0566
binary	0.0636	0.0722

Table 1: Mean error rates for the different preprocessings in the training and test set.

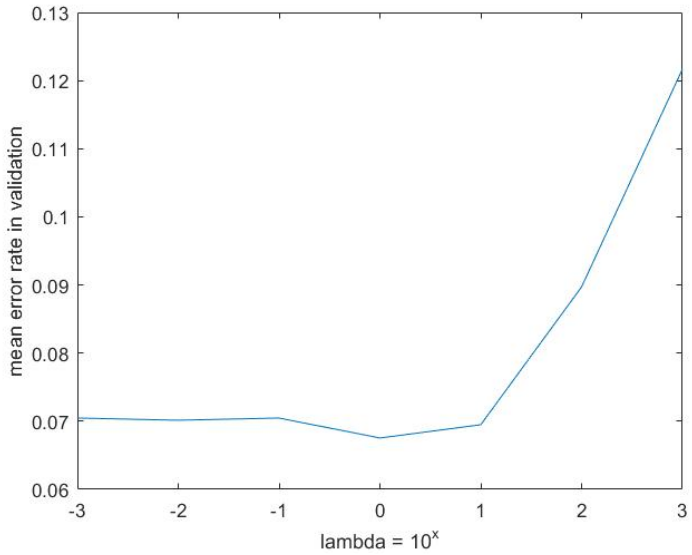

Figure 5: Binarization preprocessing - mean error rate for each λ. Best $\lambda = 1$.