Q1) https://github.com/atrakriv/hw4.git

In this question logistic regression has been implemented. The error function is the negative logarithm of the likelihood, namely, Cross-entropy error function:

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = \sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

Newton-Raphson Algorithm uses a local quadratic approximation to the cross-entropy error function to update w iteratively:

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - H^{-1}\nabla E(\mathbf{w})$$

For cross-entropy error function, gradient and Hessian matrix can be calculated as follows:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n = \boldsymbol{\Phi}^T(\mathbf{y}-\mathbf{t})$$

$$H = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T = \boldsymbol{\Phi}^T\mathbf{R}\boldsymbol{\Phi}$$

where $n^{th}$ row of $\Phi$ is $\varphi_n^T$,
and R is an $N \times N$ diagonal matrix, and $R_{nn} = y_n(1 - y_n)$.

The Newton-Raphson update for cross-entropy error function is:

$$\begin{aligned}\mathbf{w}^{(\text{new})} &= \mathbf{w}^{(\text{old})} - (\boldsymbol{\Phi}^T\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T(\mathbf{y}-\mathbf{t}) \\ &= (\boldsymbol{\Phi}^T\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{R}\mathbf{z}\end{aligned}$$

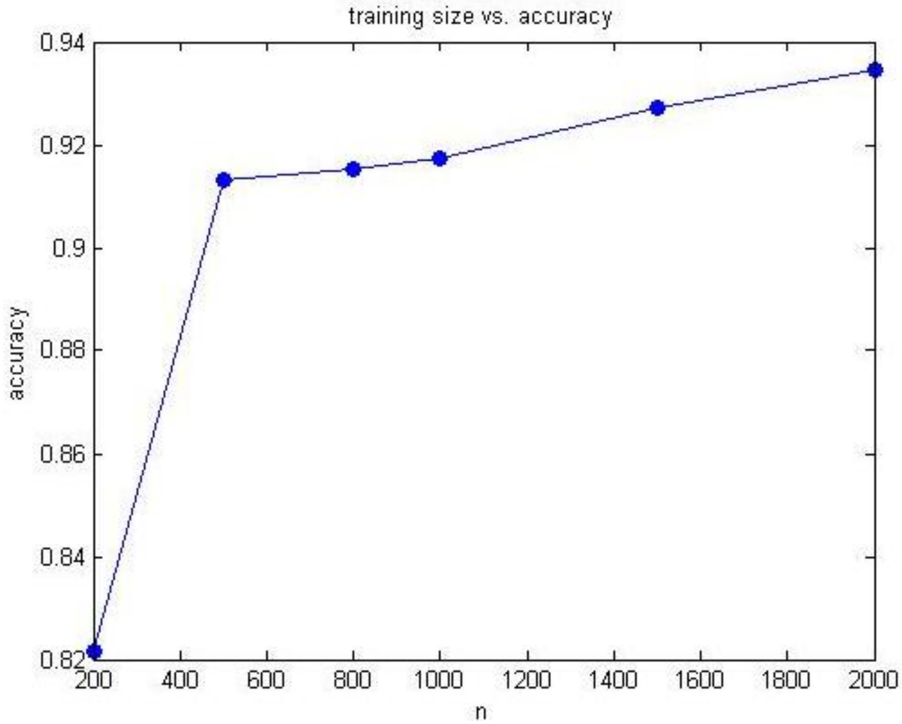
where

$$\mathbf{z} = \boldsymbol{\Phi}\mathbf{w}^{(old)} - \mathbf{R}^{-1}(\mathbf{y}-\mathbf{t})$$

A data set of size 4601 has been provided. Testing size is fixed and contained 2061 samples. Training size n is variable and accuracy on the test data as a function of n is reported.

Note: To get the final result, step function to map output values to labels (probabilities) 0 and 1 has been used.

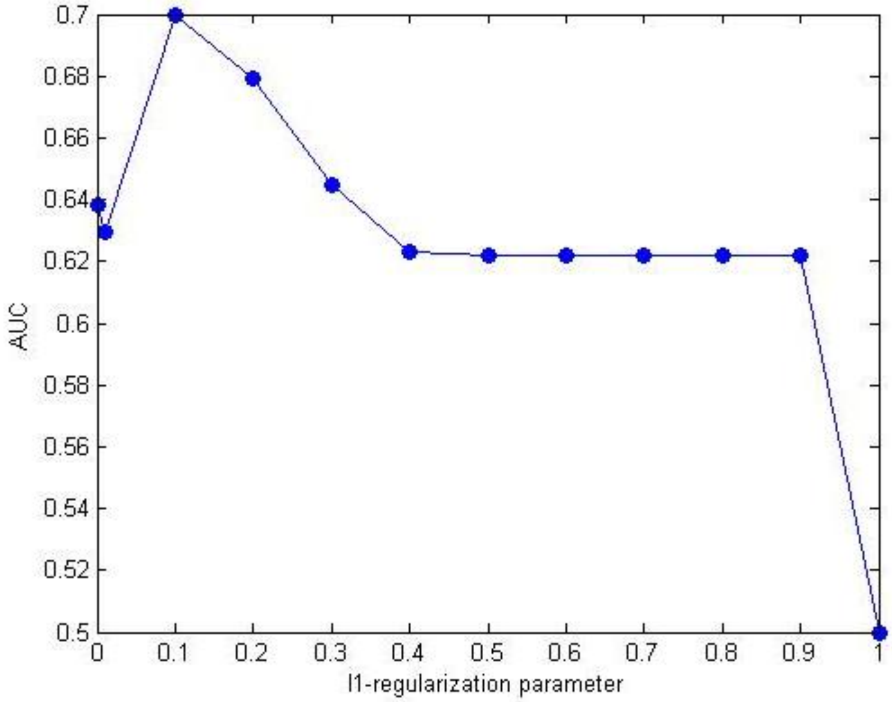
| n | 200 | 500 | 800 | 1000 | 1500 | 2000 |
|---|---|---|---|---|---|---|
| accuracy | 0.82161 | 0.91311 | 0.91542 | 0.91734 | 0.92734 | 0.93464 |

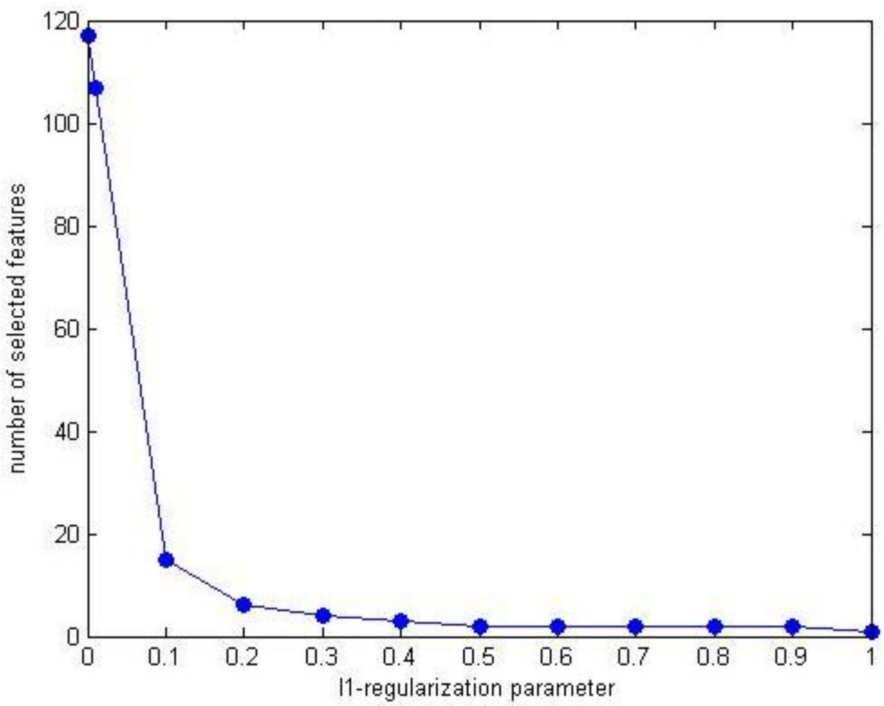



Q2) https://github.com/atrakriv/hw4.git

In this problem sparse logistic regression has been implemented. $LogisticR$ function has been used to perform l1-regularization. Bias is not included in data sent as a parameter to function $logisticR$. Different values of regularization parameter have been tried. As par (l1-regularization parameter) increases the number of selected feature decreases. AUC has been used to measure the performance. In the table below AUC and number of selected feature including bias have been shown. It could be seen that best performance is obtain for par=0.1 and 15 extracted feature. So we can conclude that more features do not necessary give us better performance.

| par | AUC | Number of extracted feature |
|---|---|---|
| 0 | 0.638 | 117 |
| 0.01 | 0.629 | 107 |
| 0.1 | 0.699 | 15 |
| 0.2 | 0.679 | 6 |
| 0.3 | 0.644 | 4 |
| 0.4 | 0.622 | 3 |
| 0.5 | 0.622 | 2 |
| 0.6 | 0.622 | 2 |
| 0.7 | 0.622 | 2 |
| 0.8 | 0.622 | 2 |
| 0.9 | 0.622 | 2 |
| 1 | 0.500 | 1 |


AUC
l1-regularization parameter
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
0.54
0.52
0.5
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


number of selected features
l1-regularization parameter
120
100
80
60
40
20
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1