Katie Hughes

Tim Dufala

Jasper LaFortune

Assignment 2

For this project we implemented a logistic regression on handwritten digit recognition. Samples of the handwritten digits four and nine were taken from the USPS handwritten digit data set. Each image was 16x16 pixels, where each pixel's gray-scale value (between 0 and 255) is taken independently as a feature (a total of 256 features).

We found that a learning rate of 10^{-5} was ideal to ensure gradient descent converges; anything beyond this did not see improvement in the test accuracy (see *Figure 1*). We attempted using a threshold on the step size (norm of the gradient) to determine when to converge. However, convergence seems to be chaotic with the chosen numbers (either the step size was on the order of 10^{-4}, or 0). That is, we would "step around" the solution for many steps, until the weights were perfect. Instead, we chose to stop iterating after a fixed number of steps to be more consistent, since performance improved only marginally beyond 30 steps (see *Figure 2*).

To introduce regularization to the logistic regression algorithm, we added a regularization term to our update step ($d = ... + \lambda w$):

Given: training examples (x^i,y^i), i = 1, …, N

Let: $w \leftarrow (0, 0, ..., 0)$

Repeat for a fixed number of (for instance, k = 50) steps.

$d \leftarrow (0, 0, ..., 0)$

For i = 1 to N do

$\hat{y} \leftarrow 1/(1 + e^{-w \cdot x^i})$

$error = y^i - \hat{y}^i$

$d = d + error \cdot x^i + \lambda w$

$w \leftarrow w + \eta d$

We plotted the SSE vs. the regularization term in *Figure 3* for the training and testing data. We found that a regularization term between 10^2 and 10^3 minimizes the SSE (an SSE of approximately 16, vs. 19 without regularization) on the test data. Lower regularization values see no effect on the SSE, whereas higher values cause SSE to skyrocket, as expected. Where the testing performance is minimized, the training performance decreases slightly. This makes sense, since the flexibility that allows the classifier to classify testing data also hinders its ability to perform on the training data (it should perform about as well on both data sets, otherwise it risks underfitting or overfitting).

Figure 1

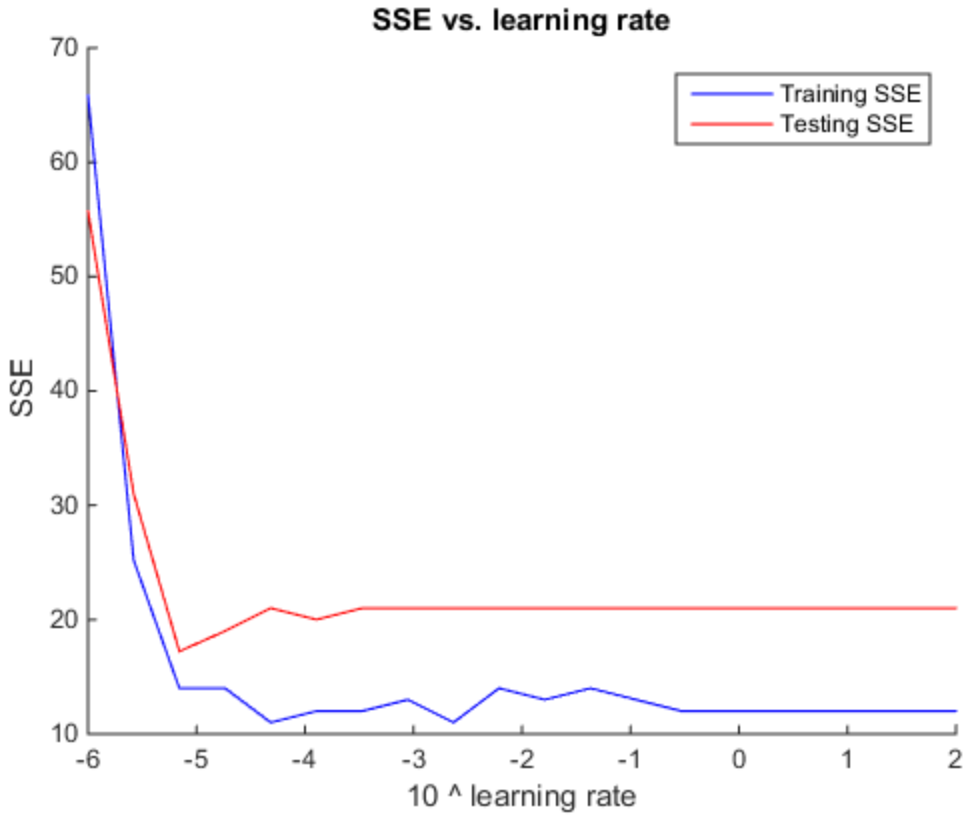

SSE vs. learning rate
Training SSE
Testing SSE
SSE
70
60
50
40
30
20
10
-6
-5
-4
-3
-2
-1
0
1
2
10 ^ learning rate

Figure 2

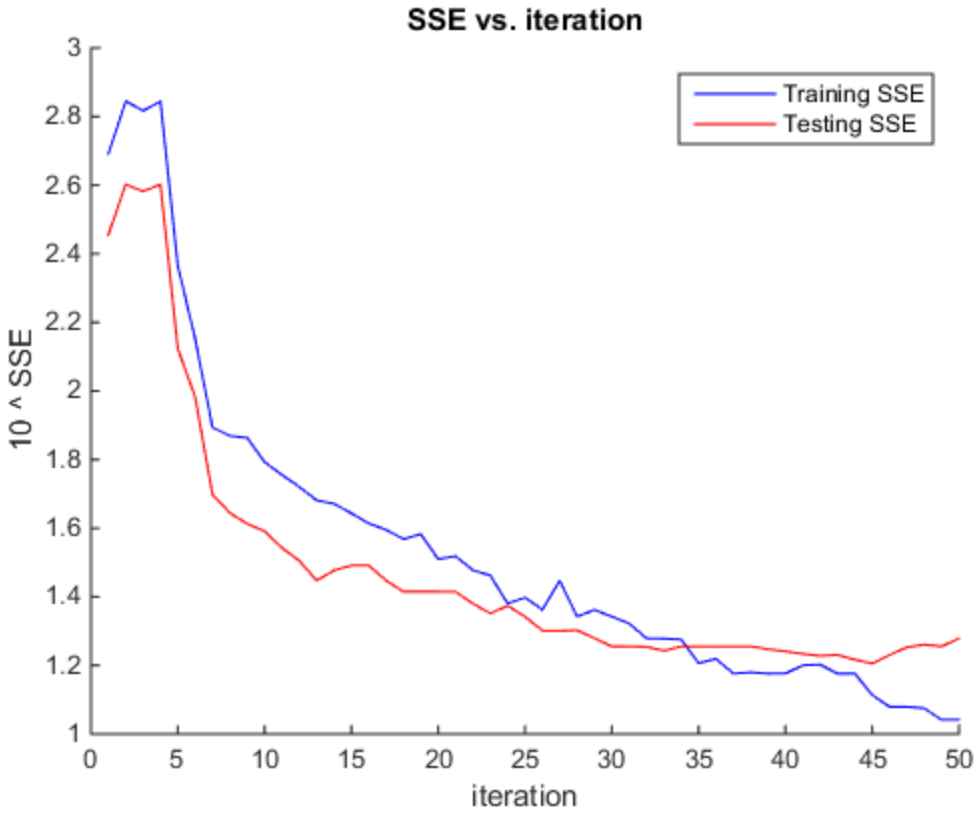
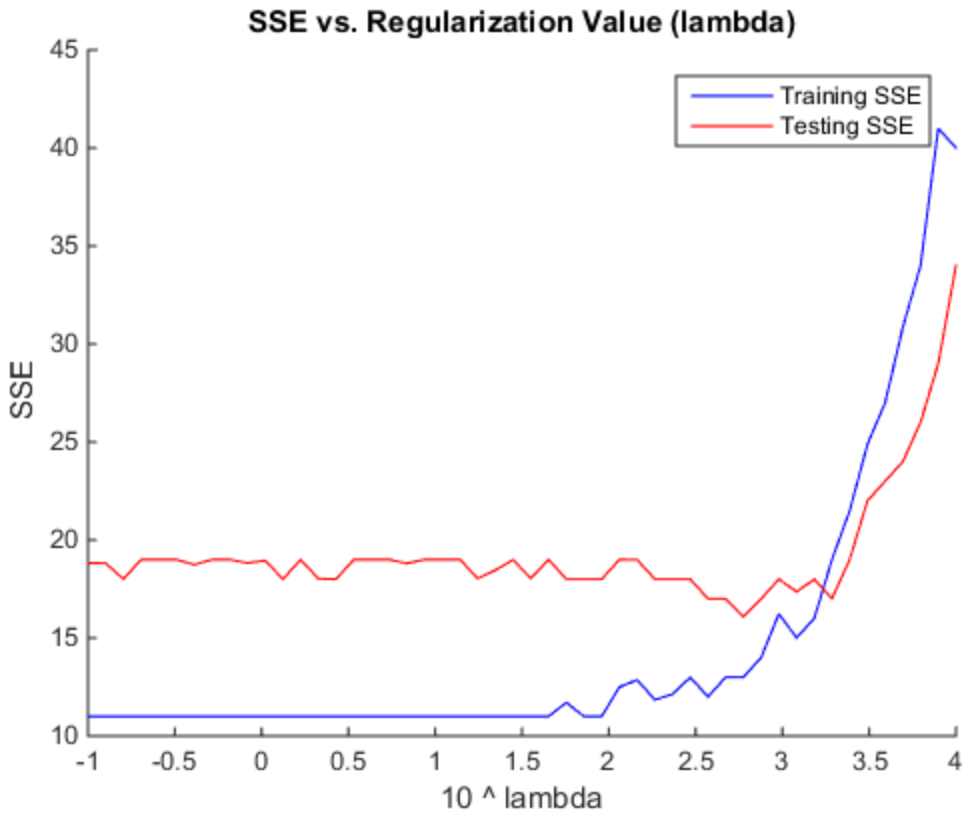

SSE vs. iteration
Training SSE
Testing SSE
10 ^ SSE
iteration
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
5
10
15
20
25
30
35
40
45
50

Figure 3


SSE vs. Regularization Value (lambda)
Training SSE
Testing SSE
SSE
45
40
35
30
25
20
15
10
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
10 ^ lambda