CSCI 6100 Machine Learning From Data
Fall 2018

HOMEWORK 7
Daniel Southwick
661542908
southd@rpi.edu

1. Classifying Handwritten Digits: 1 vs. 5

a) From Problem 3.6(c), we have developed a linear program for nonlinear separable data, which is the same as Problem 3.5. The *optimproblem* function from Matlab was used to do the following minimization

$$\min_{w} \sum_{i=1}^{N} \max(0, 1 - y_n w^T x_n)$$

The computed result vector is $w_{linear} = [4.52251.0666, -13.3283]$ and we plotted $\dfrac{(w_{linear}(1) + w_{linear}(2) * x)}{w_{linear}(3)}$ with x in the same range as the data intensity range, and resulted the following for the training set and the testing set of data:

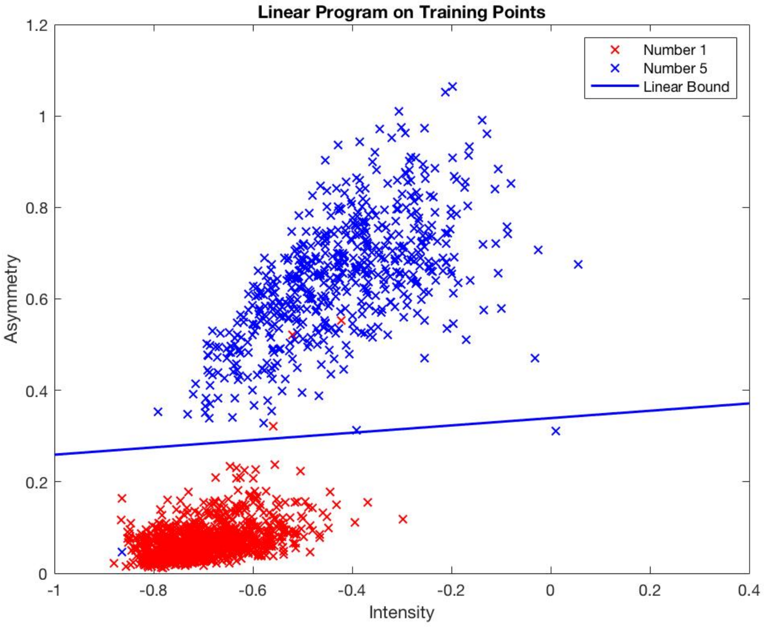

Figure 1: Linear Program on Training Points

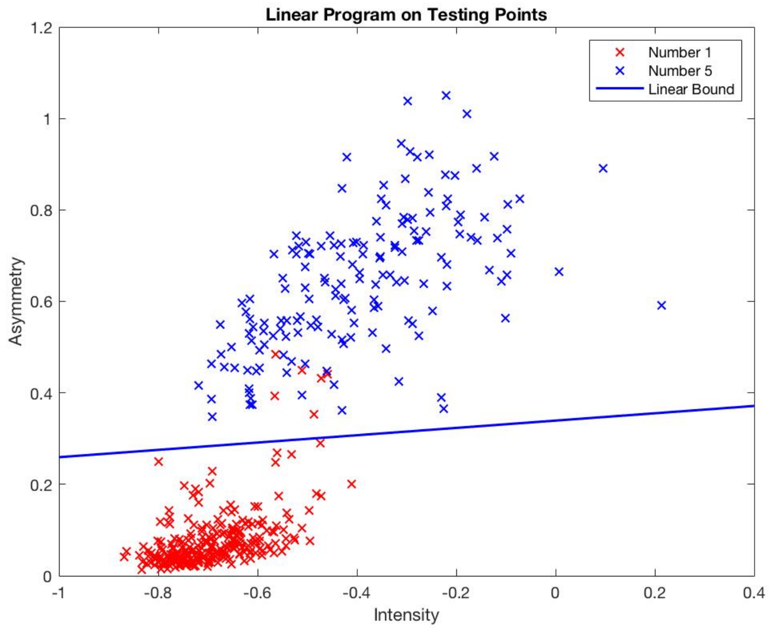

Figure 2: Linear Program on Testing Points

b) We can compute the error by simply counting the number of misclassified samples and divide by the total sample numbers, since we are only considering $+1$ and -1's to categorize our decision. There are 3 numbers of digit 1 and 2 numbers of digit 5 got misclassified in the training points. so the in sample error:

$$E_{in} = \frac{3+2}{1005+556} = 0.0032$$

And there are 6 numbers of digit 1 and 0 numbers of digit 5 got misclassified, so the out sample error:

$$E_{out} = \frac{6+0}{264+160} = 0.0142$$

c) For the training set, there are total of 1561 points and 2 features so:

$$\begin{aligned}
N &= 1561, \qquad \delta = 0.05 \\
d_{vc} &= 2+1 = 3 \\
E_{out} &\leq E_{in} + \sqrt{\frac{8}{N}\ln\left(\frac{4((2N)^{d_{vc}}+1)}{\delta}\right)} \\
&\leq E_{in} + \sqrt{\frac{8}{1561}\ln\left(\frac{4(3121^3+1)}{0.05}\right)} \\
&= 0.0032 + 0.382310 \\
E_{out} &= 0.385510
\end{aligned}$$

For the testing set, there are total of 424 points:

$$\begin{aligned} N &= 424, \qquad \delta = 0.05 \\ d_{vc} &= 2+1 = 3 \\ E_{out} &\leq E_{in} + \sqrt{\frac{1}{2N}\ln\frac{2M}{\delta}} \\ &\leq E_{in} + \sqrt{\frac{1}{848}\ln\frac{2}{0.05}} \\ &= 0.0142 + 0.065955 \\ E_{out} &= 0.080155 \end{aligned}$$

The bound on the testing set is much better than the bound on the training set.

d) We apply the 3rd order polynomial transform to the existing method by implementing $\Phi(x) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1x_2^2, x_1^2x_2)$ and resulted the following for the training set and the testing set of data:

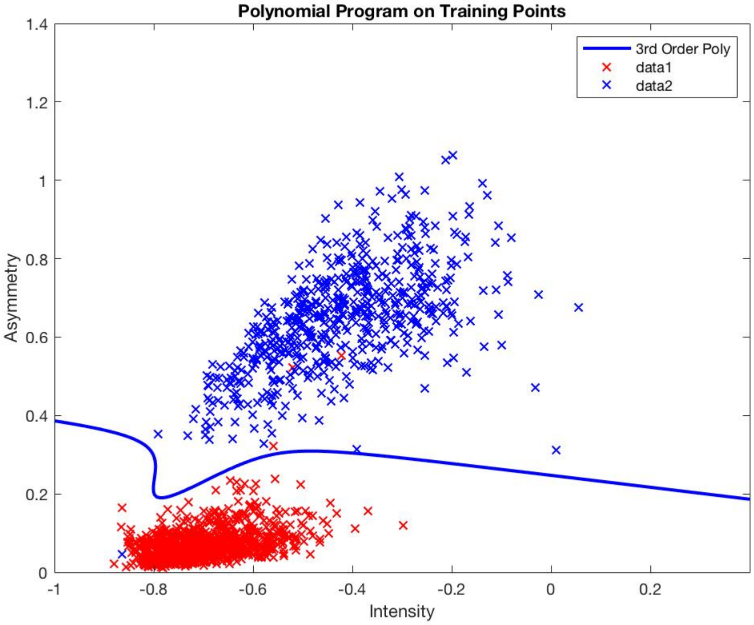

Figure 3: Quadratic Program on Training Points

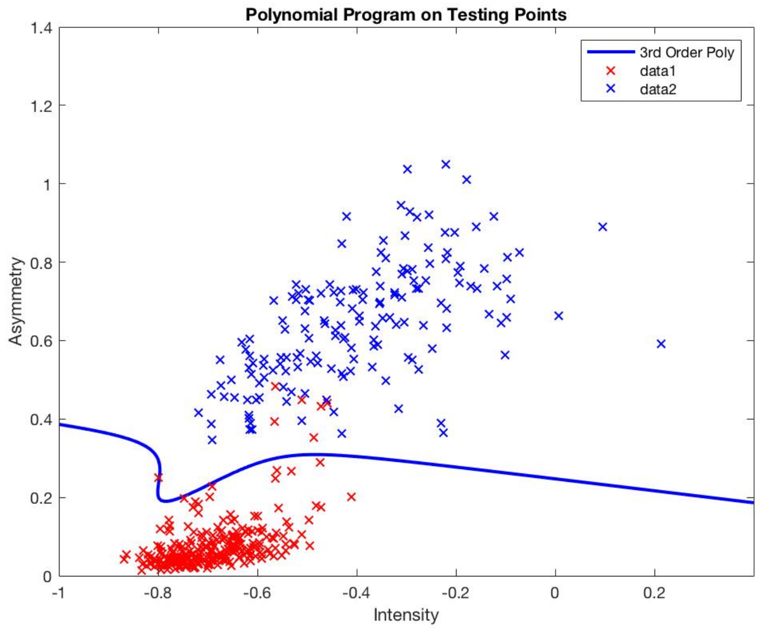

Figure 4: Quadratic Program on Testing Points

Then, there are 3 numbers of digit 1 and 1 numbers of digit 5 got misclassified in the training points. so the in sample error:

$$E_{in} = \frac{3+1}{1005+556} = 0.0026$$

And there are 6 numbers of digit 1 and 0 numbers of digit 5 got misclassified, so the out sample error:

$$E_{out} = \frac{6+0}{264+160} = 0.0142$$

and for the training, testing E_{out} with a new $d_{vc} = 10$

$$\begin{aligned} N &= 1561, \qquad \delta = 0.05 \\ d_{vc} &= 2+1 = 3 \\ E_{out} &\leq E_{in} + \sqrt{\frac{8}{N}\ln\left(\frac{4((2N)^{d_{vc}}+1)}{\delta}\right)} \\ &\leq E_{in} + \sqrt{\frac{8}{1561}\ln\left(\frac{4(3121^{10}+1)}{0.05}\right)} \\ E_{out} &= 0.662032 \end{aligned}$$

$$\begin{aligned}
N &= 424, \qquad \delta = 0.05 \\
d_{vc} &= 2 + 1 = 3 \\
E_{out} &\leq E_{in} + \sqrt{\frac{1}{2N} \ln \frac{2M}{\delta}} \\
&\leq E_{in} + \sqrt{\frac{1}{848} \ln \frac{2}{0.05}} \\
&= 0.0142 + 0.065955 \\
E_{out} &= 0.080155
\end{aligned}$$

e) So I would definitely prefer using the linear model, since the 3rd order polynomial transform did not only result in worse performance on testing set, but also gives a higher bound of E_{out} theoretically, due to the increasing of VC dimension. So in this case, a simpler model is better.

2. Gradient Descent on a "Simple" Function

$$f(x, y) = x^2 + 2y^2 + 2\sin(2\pi x)\sin(2\pi y)$$

$$\nabla f(x, y) = \begin{bmatrix} 2x + 4\pi\cos(2\pi x)\sin(2\pi y) \\ 4y + 4\pi\sin(2\pi x)\cos(2\pi y) \end{bmatrix}$$

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} - \eta \begin{bmatrix} 2x_t + 4\pi\cos(2\pi x_t)\sin(2\pi y_t) \\ 4y_t + 4\pi\sin(2\pi x_t)\cos(2\pi y_t) \end{bmatrix}$$

a) We applied the above gradient descent algorithm with $\eta = 0.01$ and $\eta = 0.1$ and obtained the following results:

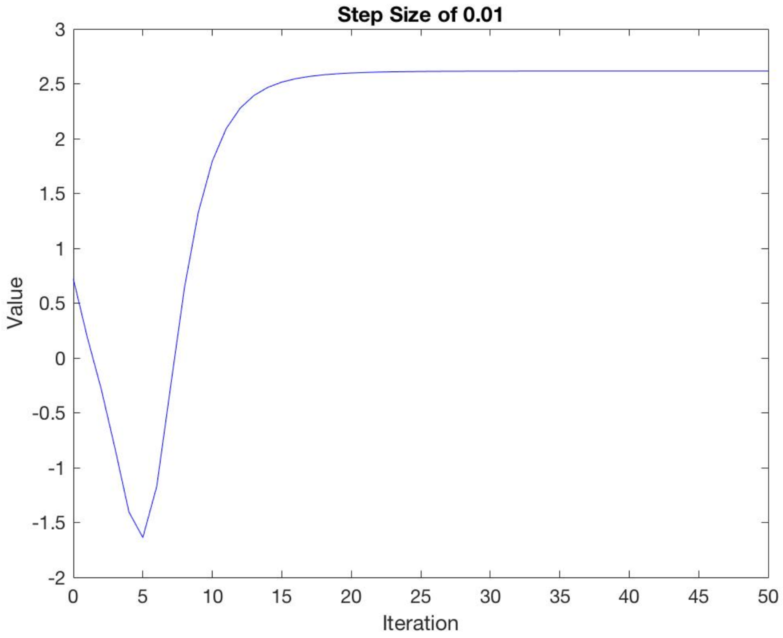

Figure 5: Gradient Descent with $\eta = 0.01$

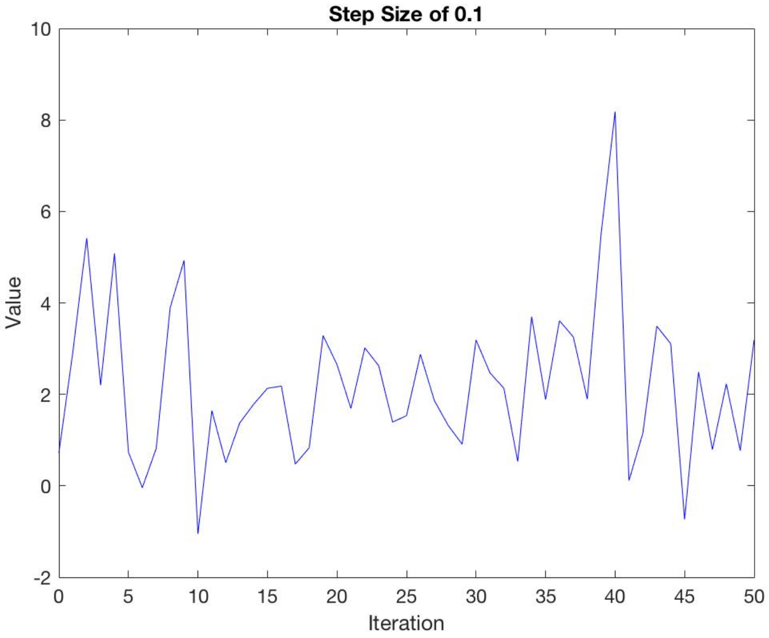

Figure 6: Gradient Descent with $\eta = 0.1$

Initial points	$\eta = 0.01$		$\eta = 0.10$	
(x_0, y_0)	$f(x^*, y^*)$	(x^*, y^*)	$f(x^*, y^*)$	(x^*, y^*)
(0.1, 0.1)	-1.6365	(-0.2807, 0.1809)	-1.0492	(-0.6476, -0.2554)
(1, 1)	2.5926	(0.9800, 0.8437)	-1.7419	(-0.2475, 0.2812)
(-0.5, -0.5)	0.7500	(-0.5, -0.5)	-1.3272	(0.7367, 0.2479)
(-1, -1)	2.5926	(-0.9800, -0.8437)	-1.7419	(0.2475, -0.2812)

We can see that the function converged with $\eta = 0.01$ and bounced around when $\eta = 0.1$, This is mainly caused by the step size. A smaller step size leads to a higher computation cost, but if the step size is not small enough, the function value will be trapped in the local minimum region thus not leading to the global minimum.

Problem 3.16

a) Since $\mathbb{P}[y = +1 \big| \mathbf{x}] = g(\mathbf{x})$, thus $\mathbb{P}[y = -1 \big| \mathbf{x}] = 1 - g(\mathbf{x})$. So:

$$\begin{aligned}
\text{Predicted}(+1) \text{ and actual } y = -1 &: \\
\text{cost(accept)} &= (1 - g(\mathbf{x}))c_a \\
\text{Predicted}(-1) \text{ and actual } y = +1 &: \\
\text{cost(reject)} &= g(\mathbf{x})c_r
\end{aligned}$$

b)So if the cost of reject $\geq$ the cost of accept, we choose to accept the person:

$$\begin{aligned}
g(\mathbf{x})c_r &\geq (1 - g(\mathbf{x}))c_a \\
g(\mathbf{x})(c_r + c_a) &\geq c_a \\
g(\mathbf{x}) &\geq \frac{c_a}{c_r + c_a} \\
\kappa &= \frac{c_a}{c_r + c_a} \\
\text{And } g(\mathbf{x}) &\geq \kappa \\
\text{Thus } \kappa &= \frac{c_a}{c_r + c_a}
\end{aligned}$$

c)

Supermarket

	$+1$	-1
$+1$	0	1
-1	10	0

CIA

	$+1$	-1
$+1$	0	1000
-1	1	0

$$\kappa = \frac{1}{1 + 10} \approx 0.0909 \qquad \kappa = \frac{1000}{1 + 1000} \approx 0.9990$$

The threshold value κ should be higher if the target focus more on the safety level, in this case, it's ok for supermarket to accept some false targets, but for CIA, intruders can cause serious problems.