Machine Learning

Assignment 3

Ajay Kumar Loganathan Ravichandran
USC ID: 1669-4689-06

1)

Given:

$y = x^T\beta^* + \varepsilon$

$\widehat{\beta}_\lambda = argmin_\beta(\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i{}^T\beta)^2 + \lambda||\beta||_2^2$

a)

To Find:

Closed form of $\widehat{\beta}_\lambda$

Solution:

The closed form solution of $\widehat{\beta}_\lambda$ can be written as

$\widehat{\beta}_\lambda = (X^TX + \lambda I)^{-1}X^Ty$

Substituting y we get,

$\widehat{\beta}_\lambda = (X^TX + \lambda I)^{-1}X^Ty + (X\beta^* + \varepsilon)$

$\widehat{\beta}_\lambda = (X + \lambda I)^{-1}X^TX\beta^* + (X^TX + \lambda I)^{-1}X^T\varepsilon$

Since ε is from a multivariate gaussian distribution, By affine transformation property of gaussian distribution

$$\widehat{\beta}_\lambda \sim N((X^TX + \lambda I)^{-1}X^TX\beta^* + (X^TX + \lambda I)^{-1}X^T\Sigma X(X^TX + \lambda I))$$

b)

To Calculate:

Bias

Solution:

Bias $= E[x^T\widehat{\beta}_\lambda] - x^T\beta^*$

$= x^TE[\widehat{\beta}_\lambda] - x^T\beta^*$

Substituting $\widehat{\beta}_\lambda$ in the above equation we get,

Bias $= x^T E[[X^TX + \lambda I]^{-1}X^Ty] - x^T\beta^*$

$= x^T [[X^TX + \lambda I]^{-1}X^T . E[y]] - x^T\beta^*$

$= x^T [[X^TX + \lambda I]^{-1}X^T . X\beta^*] - x^T\beta^*$

$= x^T[[X^TX + \lambda I]^{-1} X^TX - I]\, \beta^*$

c)

Since $x^T\,[\hat{\beta}_\lambda - E\,[\hat{\beta}_\lambda]]$ follows a gaussian distribution

Its variance can be shown as $\Rightarrow x^T\,[X^TX + \lambda I]^{-1} X^T\, X[X^TX + \lambda I]\, x$

Since χ^2 variable is square of a gaussian random variable,

Mean of χ^2 distribution $\Rightarrow \left\| X[X^TX + \lambda I]^{-1}\, x \right\|^2$

Variance $\Rightarrow E[\,[x^T[\beta_\lambda - E\,[\beta_\lambda]\,]]^2\,] = \left\| X[X^TX + \lambda I]^{-1}\, x \right\|^2$

d)

Known: Error $= Bias^{\,2} + variance + Noise$

From the results of (b) and (c),

We see that λ is directly proportional to bias and inversely proportional to variance.

Thus,

When λ is large, the bias term dominates the error.

When λ is low, the variance term dominates the error.

2)

Kernel Construction

Given: $k_1(\cdot,\cdot)\ and\ k_2(\cdot,\cdot)$ are kernel functions, $x\ and\ x'$ are any two samples.

Mercer's theorem states a bivariate function $k(\cdot,\cdot)$ is a positive definite kernel function if and only if the matrix $K\ \{K_{ij} = k(x_i, x_j)\}$ is positive semi definite.

K is positive semidefinite if all its eigenvalues are non-negative.

A)

To Prove:

$k_3(x,x') = a_1k_1(x,x') + a_2k_2(x,x')\ \ where\ a_1, a_2 > 0$

Solution:

Since $k_1 and\ k_2$ are kernel functions, $K_1 and\ K_2$ are positive definite matrices.Then according to the theorems of positive definite matrices, the sum of the positive definite matrices is also a positive definite matrix.

For any $c > 0$, cK_1 will also be a positive semi definite.

Thus, we have some positive definite matrix $K' = K_1 + K_2$

$\Rightarrow K' = a_1 K_1 + a_2 K_2$ is positive semi definite given $a_1 > 0\ and\ a_2 > 0$

$\Rightarrow k_3(x,y) = a_1 k_1(x,y) + a_2 k_2(x,y)$ will be a valid kernel function

B)

To Prove:

$k_4(x,x') = f(x)f(x')$, where $f(\cdot)$ is a real valued function.

Solution:

We know that the matrix form of the kernel function k_n *is* K_n, *where* K_n is:

$K_n = [[k_n(x,x), k_n(x,x')], [k_n(x',x), k_n(x',x')]]$.

Thus, the eigenvalues for the given kernel function can be calculated as

$y = det(K_4 - I\lambda) = 0$

$y = (f(x)^2 - \lambda).(f(x')^2 - \lambda) - (f(x).f(x')).(f(x).f(x')) = 0$

$y = \lambda^2 - \lambda(f(x)^2 + f(x')^2) + D = 0$

Thus solving for λ,

$\frac{dy}{d\lambda} = 2\lambda - (f(x)^2 + f(x')^2) = 0$

$\Rightarrow \lambda = \frac{(f(x)^2 + f(x')^2)}{2}$

In order to verify if the λ is minima , $\frac{d^2y}{d\lambda^2}$ must be greater than 0.

$\frac{d^2y}{d\lambda^2} = 2$

Thus the function y is concave with λ as its minimum value.

Therefore, K_4 is positive semi definite and k_4 is a valid kernel function.

C)

To prove:

$k_5(x,x') = k_1(x,x')k_2(x,x')$

Solution:

According to **Schur product theorem**, The inner product of positive semidefinite matrices is also positive semidefinite.

Since K_1 *and* K_2 are positive semidefinite K_5 *is semidefinite if*,

$K_5 = K_1 \bullet K_2$

Thus k_5 is a valid kernel function where,

$k_5(x,x') = k_1(x,x')k_2(x,x')$

3)

Kernel Regression

Given:

Training data : $(x_1,y_1),(x_2,y_2)....,(x_n,y_n)$, $x_i \in R^D$

Objective function (OF(w)) : $min_w \sum_n (y_i - w^T x_i)^2 + \lambda||w||^2$

Where λ is the regularization parameter

a)

Solution:

The objective function for Linear Ridge regression is given by:

$$RSS(w) = min_w \sum_n (y_i - w^T x_i)^2 + \lambda||w||^2$$

Differentiating and equating to 0 we get,

$$\frac{\partial RSS(w)}{\partial w} = 2\sum_n (y_i - w^T x_i)(-x_i) + 2\lambda w = 0$$

$$\sum_n (y_i x_i) = \lambda w + \sum_n x_i^T x_i w$$

Converting the above equation to Vector form,

$$\Rightarrow X^T Y = (X^T X + \lambda I)w$$

$$\Rightarrow w = (X^T X + \lambda I)^{-1} X^T Y$$

b)

To Prove:

$w = \Phi^T(\Phi\Phi^T + \lambda I_N)^{-1} Y$

Solution:

When a nonlinear feature mapping Φ is applied,

$$RSS(w) = min_w \sum_n (y_i - w^T \Phi(x_i))^2 + \lambda||w||^2$$

Differentiating and equating to 0 we get,

$$\frac{\partial RSS(w)}{\partial w} = 2\sum_n (y_i - w^T \Phi(x_i))(-\Phi(x_i)) + 2\lambda w = 0$$

$$-\sum_n (y_i \Phi(x_i)) + \sum_n \Phi(x_i)\Phi(x_i)^T w = -\lambda w$$

$$\sum_n (y_i \Phi(x_i)) = \lambda w + \sum_n \Phi(x_i)\Phi(x_i)^T w$$

Converting the above equation to Vector form,

$$\Rightarrow \Phi^T Y = (\Phi\Phi^T + \lambda I_N)w$$

$$\Rightarrow w = (\Phi\Phi^T + \lambda I_N)^{-1}\Phi^T Y$$

Applying Inverse Matrix lemma we get,

$w^* = \Phi^T(\Phi\Phi^T + \lambda I_N)^{-1} Y$

c)

Given:

Kernel matrix $K_{ij} = \Phi_i^T \Phi_j$

Prediction $\hat{y} = w^{*T}\Phi(x)$

Nth element $k(x)_n = \Phi(x_n)^T \Phi(x)$

To Prove:

$$\bar{y} = Y^T (K + \lambda I_N)^{-1} \lambda(x)$$

Solution:

From (b), we know

$w^* = \Phi^T (\Phi\Phi^T + \lambda I_N)^{-1} Y$

Hence,

$w^{*T} = [\Phi^T (\Phi\Phi^T + \lambda I_N)^{-1} Y]^T$

$w^{*T} = Y^T ((\Phi^T\Phi + \lambda I_N)^{-1})^T \Phi$

$w^{*T}\Phi(x) = Y^T ((\Phi^T\Phi + \lambda I_N)^{-1})^T \Phi\,\Phi(x)$

$\hat{y} = w^{*T}\Phi(x)$

$\hat{y} = Y^T (K + \lambda I_N)^{-1} \lambda(x)$

d)

<u>For linear Regression</u>

⇒ The computation complexity for Linear Regression → $O(D^3 + ND^2)$

<u>For Kernel Regression</u>

⇒ The computation complexity for Kernel Regression → $O(N^3 + TN^2)$

4)

Support Vector Machines

Given:

Positive examples: (1,1) and (-1,-1)

Negative examples: (1,-1) and (-1,1)

Solution:

a) No

b) Φ(x) = [1, x_1 , x_2 , x_1 x_2] where x_1 x_2 are first and second coordinates of x

The coordinates for the positive examples in the feature space are (1,1,1,1) , (1,-1, -1,1)
and the coordinates for the negative examples in the feature space are (1,1,-1,-1), (1,-1,1,-1).
The final coordinate of each point helps in the decision boundary separating the classes with the maximum margin.

Therefore, W = [0,0,0,1]

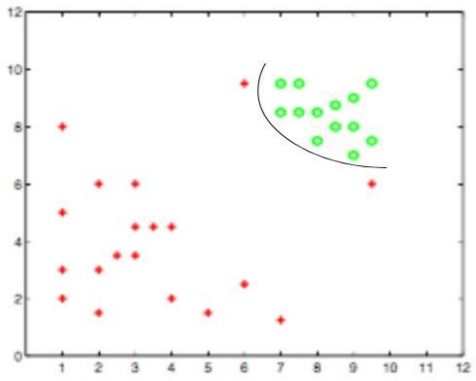
c) Consider the training example (-0.5,0.5), being a positive class, it'll not be classified as positive in the new feature space by W. Adding the example will make them no longer linearly separable.

d) The Kernel $k(x,x)$ corresponds to a polynomial of degree 2 or a quadratic Kernel as it has the product of x_1 x_2 in the kernel function.

5)

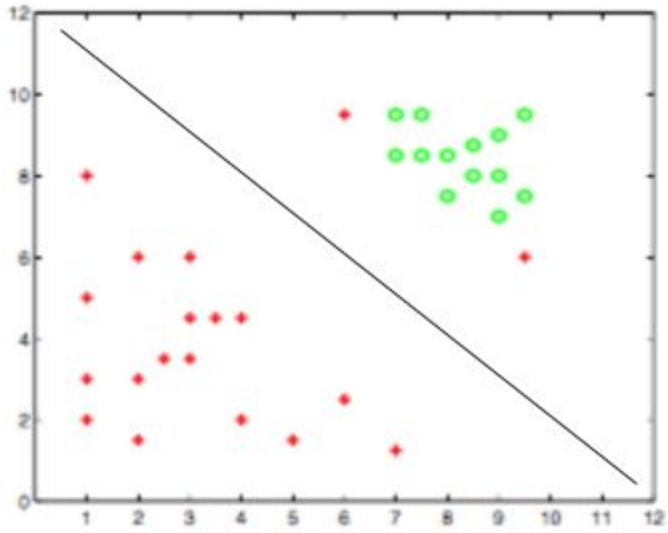
Support Vector Machines and Slack Penalty

a)
When $C \to \infty$, Choosing a quadratic kernel will try to fit the data such that the separation between the data is precise thus classifying the data to its classes accurately. This might lead to overfitting the curve as there is an increase in the variance and decrease in the bias.



(a) Part 1

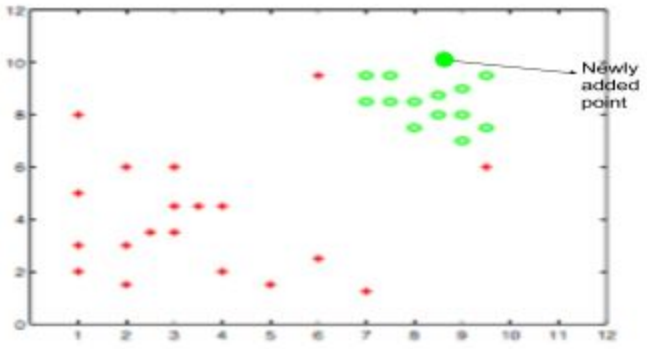
b)
When $C \to 0$, the classes gets separated by a larger margin separating hyperplane even if the hyperplane misclassifies more points. This results in misclassified data points even though the data points are linearly separable. Thus When C tends to zero, bias increases resulting in underfitting.



(b) Part 2

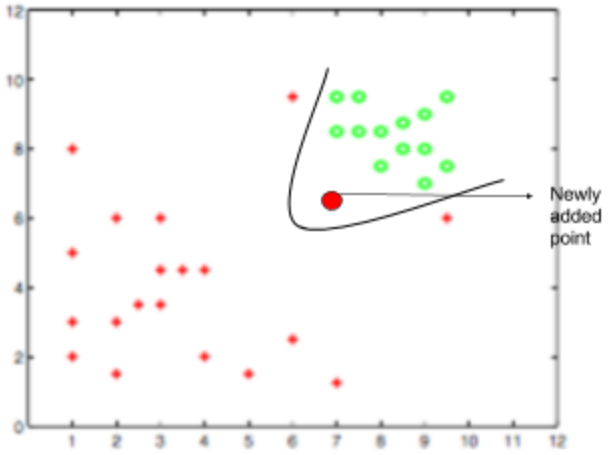
c)
Given that the training data comes from the sensors which can be error-prone, it is highly likely that some data could be misclassified. Therefore, Such outliers shouldn't be considered in the classification process. In such a case, the second model (High bias case) is prefered over the first model (High variance case - which classifies the outliers to fit the curve).

d)
Since as C increases the curve tries to fit the data points more accurately, adding a new point in the corresponding class region away from the decision boundary will not affect the curve.



(c) Part 4

e)
Since as C increases the curve tries to fit the data points more accurately, adding a new point in the negative class region near the decision boundary will affect the curve. The curve should be readjusted in order to accommodate the data points pertaining to the corresponding class alone such that it maintains the maximum margin condition. In the below graph, the curve should be re-adjusted.



(d) Part 5

PROGRAMMING RESULTS

BIAS AND VARIANCE

10 SAMPLE DATASET

MSE

G1 = 587.272367988
G2 = 411.648686634
G3 = 336.406074007
G4 = 65.1567522217
G5 = 56.5368466705
G6 = 47.2907612187

Bias

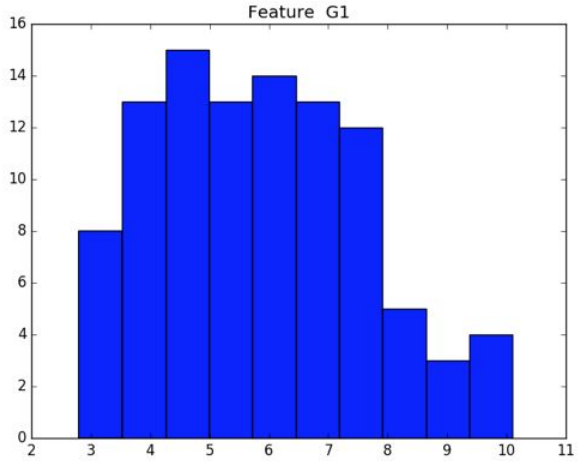
G1 = 0.4022423

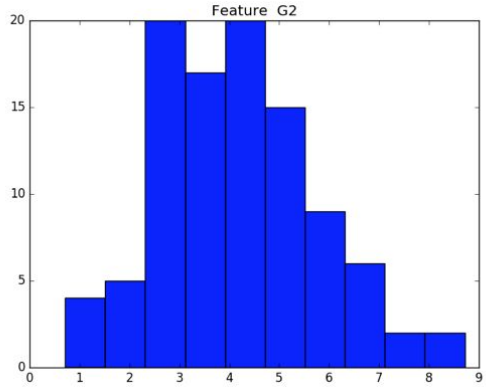
G2 = 0.3236924
G3 = 0.34001104
G4 = 7.83707387e-05
G5 = 0.00936888
G6 = 0.23971696

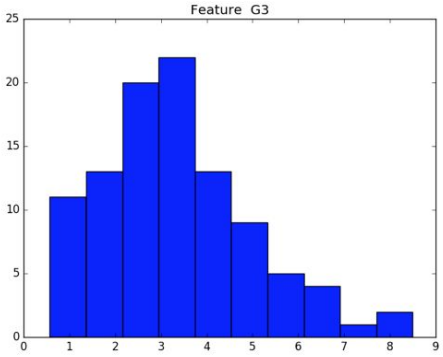
Variance

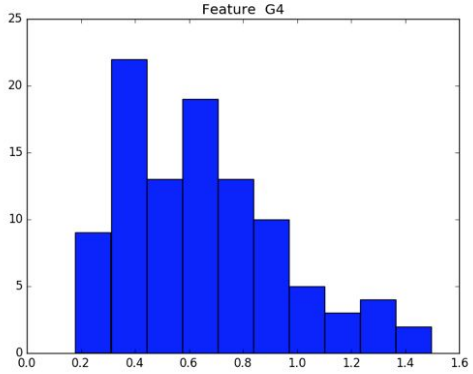
G1 = 0
G2 = 0.0408270
G3 = 0.15396917
G4 = 0.04704384
G5 = 0.78244776
G6 = 10.00480864

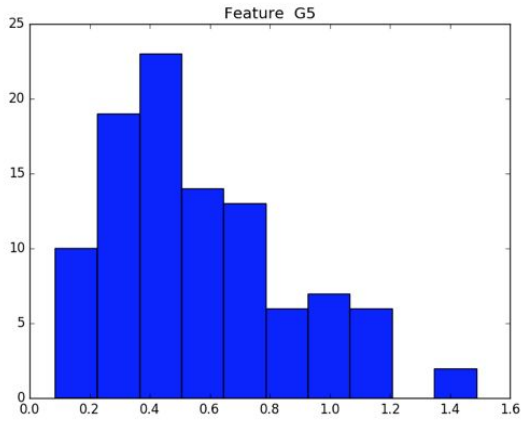
HISTOGRAM FOR 10 SAMPLE DATASET

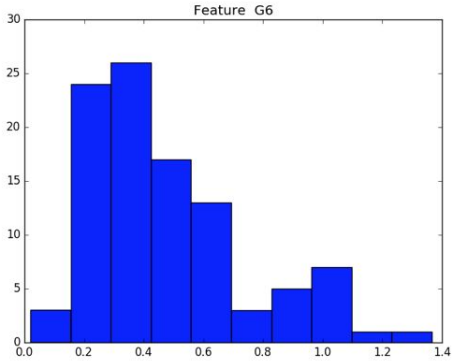



Feature G2
20
15
10
5
0
0 1 2 3 4 5 6 7 8 9


Feature G3
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8 9


Feature G4
25
20
15
10
5
0
0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6





RESULTS:

1) We observe the variance is directly proportional to the degree of the polynomial and inversely proportional to bias.Thus G6 has the highest variance (overfitting) and G0 has the highest bias (under fitting)

2) G4 has the optimal variance and bias values when compared to polynomial of other degrees

3) Since higher order polynomial would fit the dataset better, the sum error is inversely proportional to the degree of polynomial.

100 SAMPLE DATASET

MSE

G1 = 5616.8793049
G2 = 4479.41084177
G3 = 4409.39218108
G4 = 986.874182875
G5 = 977.108079433
G6 = 967.474636932

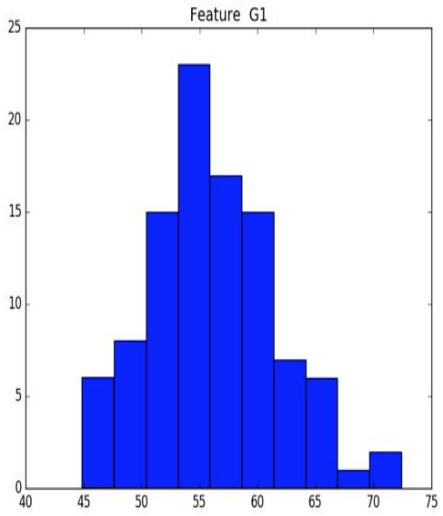
Bias

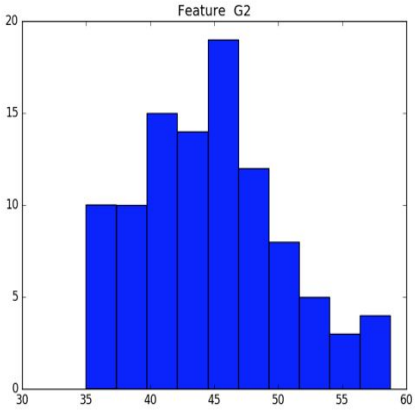
G1 = 0.42149646
G2 = 0.29851561
G3 = 0.33138999
G4 = 2.75616883e-05
G5 = 4.35487746e-05
G6 = 4.16561593e-05

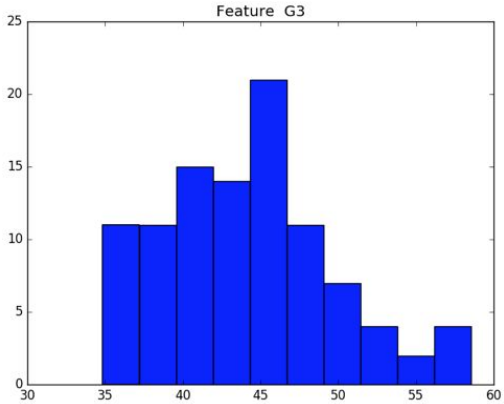
Variance

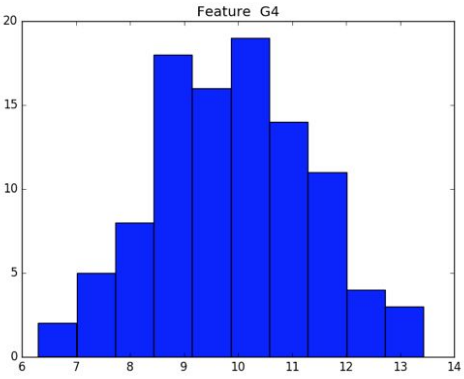
G1 = 0.0
G2 = 0.0057256
G3 = 0.01103276
G4 = 0.00258432
G5 = 0.00370959
G6 = 0.00477902

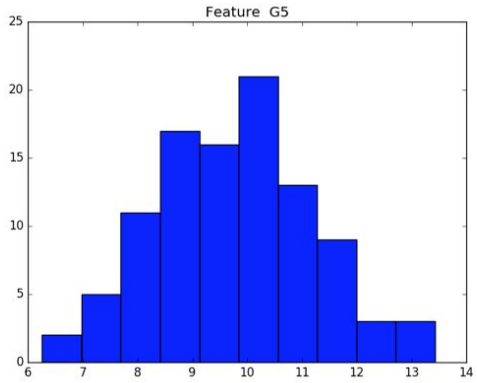
HISTOGRAMS FOR 100 SAMPLE DATASET

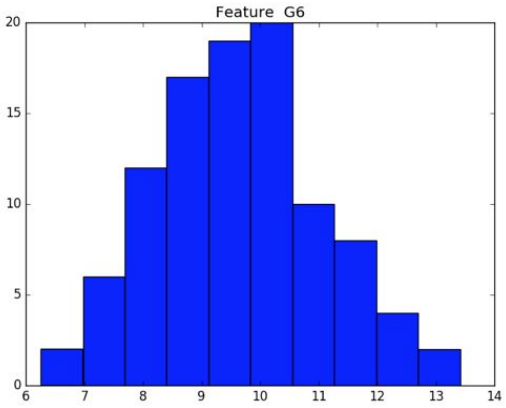



Feature G2
20
15
10
5
0
30
35
40
45
50
55
60


Feature G3
25
20
15
10
5
0
30
35
40
45
50
55
60


Feature G4
20
15
10
5
0
6
7
8
9
10
11
12
13
14





OBSERVATIONS:

1) Sum error varies similar to the previous sample.
2) Bias decreases as the order of polynomial increases
3) Variance is highest for G3

C)

It is observed that G4 better performs in the terms both bias and variance.

Effect of sample size:

1)Variance decreases as the sample size increases because the higher order polynomial functions will fit the data with more precision.
2) High bias models are invariant to increase in the sample size
3) We observe that as we use more sample data, **the variance and bias tradeoff can be observed better.**

RIDGE REGRESSION

For Lambda : 0.001

Sum Squared Error : 986.874230528

Bias : [2.74760396e-05]
variance : [0.0025841]

For Lambda : 0.003
Sum Squared Error : 986.874611548
Bias : [2.73328539e-05]
variance : [0.00258365]

For Lambda : 0.01
Sum Squared Error : 986.878937896
Bias : [2.71261654e-05]
variance : [0.00258211]

For Lambda : 0.03
Sum Squared Error : 986.916772927
Bias : [2.90420305e-05]
variance : [0.00257781]

For Lambda : 0.1
Sum Squared Error : 987.33955735
Bias : [6.43161834e-05]
variance : [0.00256403]

For Lambda : 0.3
Sum Squared Error : 990.870255106
Bias : [0.00039363]
variance : [0.00253448]

For Lambda : 1.0
Sum Squared Error : 1024.85965955
Bias : [0.00367398]
variance : [0.00251727]

LINEAR AND KERNEL SVM

LINEAR SVM
LINEAR SVM
Cross Validation Accuracy = 56.55%
C: 4^-6 Kernel: linear Train Time:0.752753
->
Cross Validation Accuracy = 90.95%
C: 4^-5 Kernel: linear Train Time:0.652666
->
Cross Validation Accuracy = 92.85%
C: 4^-4 Kernel: linear Train Time:0.477928
->
Cross Validation Accuracy = 93.55%
C: 4^-3 Kernel: linear Train Time:0.342659
->
Cross Validation Accuracy = 94.2%
C: 4^-2 Kernel: linear Train Time:0.275248
->
Cross Validation Accuracy = 94.1%
C: 4^-1 Kernel: linear Train Time:0.267233
->
Cross Validation Accuracy = 94.55%
C: 4^0 Kernel: linear Train Time:0.270875
->
Cross Validation Accuracy = 94.65%
C: 4^1 Kernel: linear Train Time:0.36451
->
Cross Validation Accuracy = 93.95%
C: 4^2 Kernel: linear Train Time:0.816529
->
Best c 1 and accuracy is 94.65

2.a)
POLYNOMIAL KERNEL
Cross Validation Accuracy = 73.15%
C: 4^-3 Degree:1 Kernel:polynomial Train Time:1.380443
->
Cross Validation Accuracy = 56.15%
C: 4^-3 Degree:2 Kernel:polynomial Train Time:1.24257
->
Cross Validation Accuracy = 55.75%
C: 4^-3 Degree:3 Kernel:polynomial Train Time:0.887062

->
Cross Validation Accuracy = 92.05%
C: 4^-2 Degree:1 Kernel:polynomial Train Time:0.611414
->
Cross Validation Accuracy = 87.35%
C: 4^-2 Degree:2 Kernel:polynomial Train Time:0.684858
->
Cross Validation Accuracy = 78.05%
C: 4^-2 Degree:3 Kernel:polynomial Train Time:0.766119
->
Cross Validation Accuracy = 92.6%
C: 4^-1 Degree:1 Kernel:polynomial Train Time:0.428575
->
Cross Validation Accuracy = 92.9%
C: 4^-1 Degree:2 Kernel:polynomial Train Time:0.570265
->
Cross Validation Accuracy = 92%
C: 4^-1 Degree:3 Kernel:polynomial Train Time:0.761013
->
Cross Validation Accuracy = 93.35%
C: 4^0 Degree:1 Kernel:polynomial Train Time:0.428616
->
Cross Validation Accuracy = 95.2%
C: 4^0 Degree:2 Kernel:polynomial Train Time:0.37862
->
Cross Validation Accuracy = 95.4%
C: 4^0 Degree:3 Kernel:polynomial Train Time:0.439579
->
Cross Validation Accuracy = 94.4%
C: 4^1 Degree:1 Kernel:polynomial Train Time:0.282767
->
Cross Validation Accuracy = 95.8%
C: 4^1 Degree:2 Kernel:polynomial Train Time:0.285423
->
Cross Validation Accuracy = 96.65%
C: 4^1 Degree:3 Kernel:polynomial Train Time:0.324083
->
Cross Validation Accuracy = 94.4%
C: 4^2 Degree:1 Kernel:polynomial Train Time:0.268133
->
Cross Validation Accuracy = 96%
C: 4^2 Degree:2 Kernel:polynomial Train Time:0.275178
->

Cross Validation Accuracy = 96.15%
C: 4^2 Degree:3 Kernel:polynomial Train Time:0.294088
->
Cross Validation Accuracy = 94.55%
C: 4^3 Degree:1 Kernel:polynomial Train Time:0.319261
->
Cross Validation Accuracy = 96.1%
C: 4^3 Degree:2 Kernel:polynomial Train Time:0.276268
->
Cross Validation Accuracy = 96.9%
C: 4^3 Degree:3 Kernel:polynomial Train Time:0.283088
->
Cross Validation Accuracy = 94.7%
C: 4^4 Degree:1 Kernel:polynomial Train Time:0.46625
->
Cross Validation Accuracy = 96.5%
C: 4^4 Degree:2 Kernel:polynomial Train Time:0.281881
->
Cross Validation Accuracy = 96.6%
C: 4^4 Degree:3 Kernel:polynomial Train Time:0.271867
->
Cross Validation Accuracy = 94.2%
C: 4^5 Degree:1 Kernel:polynomial Train Time:1.7558
->
Cross Validation Accuracy = 96.05%
C: 4^5 Degree:2 Kernel:polynomial Train Time:0.416484
->
Cross Validation Accuracy = 95.7%
C: 4^5 Degree:3 Kernel:polynomial Train Time:0.385776
->
Cross Validation Accuracy = 94.6%
C: 4^6 Degree:1 Kernel:polynomial Train Time:4.169622
->
Cross Validation Accuracy = 95.65%
C: 4^6 Degree:2 Kernel:polynomial Train Time:0.468552
->
Cross Validation Accuracy = 96.05%
C: 4^6 Degree:3 Kernel:polynomial Train Time:0.374166
->
Cross Validation Accuracy = 94%
C: 4^7 Degree:1 Kernel:polynomial Train Time:15.653228
->
Cross Validation Accuracy = 95.55%

C: 4^7 Degree:2 Kernel:polynomial Train Time:0.32797
->
Cross Validation Accuracy = 96.15%
C: 4^7 Degree:3 Kernel:polynomial Train Time:0.279778
->
best c 3 best d 3 and accuracy is96.9

2.b) RBF KERNEL

RBF KERNEL
Cross Validation Accuracy = 55.75%
C: 4^-3Gamma: 4^-7 Kernel:rbf Train Time:0.916404
->
Cross Validation Accuracy = 55.75%
C: 4^-3Gamma: 4^-6 Kernel:rbf Train Time:0.912432
->
Cross Validation Accuracy = 55.75%
C: 4^-3Gamma: 4^-5 Kernel:rbf Train Time:0.941561
->
Cross Validation Accuracy = 55.75%
C: 4^-3Gamma: 4^-4 Kernel:rbf Train Time:0.929828
->
Cross Validation Accuracy = 64.85%
C: 4^-3Gamma: 4^-3 Kernel:rbf Train Time:0.769456
->
Cross Validation Accuracy = 71.75%
C: 4^-3Gamma: 4^-2 Kernel:rbf Train Time:0.860933
->
Cross Validation Accuracy = 55.75%
C: 4^-3Gamma: 4^-1 Kernel:rbf Train Time:0.908052
->
Cross Validation Accuracy = 55.75%
C: 4^-2Gamma: 4^-7 Kernel:rbf Train Time:0.833437
->
Cross Validation Accuracy = 55.75%
C: 4^-2Gamma: 4^-6 Kernel:rbf Train Time:0.862768
->
Cross Validation Accuracy = 55.75%
C: 4^-2Gamma: 4^-5 Kernel:rbf Train Time:1.012484
->
Cross Validation Accuracy = 82.8%
C: 4^-2Gamma: 4^-4 Kernel:rbf Train Time:0.829407
->

Cross Validation Accuracy = 91.4%
C: 4^-2Gamma: 4^-3 Kernel:rbf Train Time:0.784855
->
Cross Validation Accuracy = 91.25%
C: 4^-2Gamma: 4^-2 Kernel:rbf Train Time:0.803296
->
Cross Validation Accuracy = 63.9%
C: 4^-2Gamma: 4^-1 Kernel:rbf Train Time:0.91198
->
Cross Validation Accuracy = 55.75%
C: 4^-1Gamma: 4^-7 Kernel:rbf Train Time:0.784174
->
Cross Validation Accuracy = 55.75%
C: 4^-1Gamma: 4^-6 Kernel:rbf Train Time:0.966062
->
Cross Validation Accuracy = 86.3%
C: 4^-1Gamma: 4^-5 Kernel:rbf Train Time:1.035303
->
Cross Validation Accuracy = 92.2%
C: 4^-1Gamma: 4^-4 Kernel:rbf Train Time:0.675535
->
Cross Validation Accuracy = 93.55%
C: 4^-1Gamma: 4^-3 Kernel:rbf Train Time:0.498155
->
Cross Validation Accuracy = 94.85%
C: 4^-1Gamma: 4^-2 Kernel:rbf Train Time:0.528534
->
Cross Validation Accuracy = 91.55%
C: 4^-1Gamma: 4^-1 Kernel:rbf Train Time:0.795615
->
Cross Validation Accuracy = 55.75%
C: 4^0Gamma: 4^-7 Kernel:rbf Train Time:0.886012
->
Cross Validation Accuracy = 86.45%
C: 4^0Gamma: 4^-6 Kernel:rbf Train Time:0.851997
->
Cross Validation Accuracy = 92.05%
C: 4^0Gamma: 4^-5 Kernel:rbf Train Time:0.630982
->
Cross Validation Accuracy = 93.35%
C: 4^0Gamma: 4^-4 Kernel:rbf Train Time:0.488019
->
Cross Validation Accuracy = 95.1%

C: 4^0Gamma: 4^-3 Kernel:rbf Train Time:0.433029
->
Cross Validation Accuracy = 95.95%
C: 4^0Gamma: 4^-2 Kernel:rbf Train Time:0.361238
->
Cross Validation Accuracy = 96.2%
C: 4^0Gamma: 4^-1 Kernel:rbf Train Time:0.643138
->
Cross Validation Accuracy = 86.8%
C: 4^1Gamma: 4^-7 Kernel:rbf Train Time:0.896615
->
Cross Validation Accuracy = 92.25%
C: 4^1Gamma: 4^-6 Kernel:rbf Train Time:0.797429
->
Cross Validation Accuracy = 93.15%
C: 4^1Gamma: 4^-5 Kernel:rbf Train Time:0.505315
->
Cross Validation Accuracy = 94.15%
C: 4^1Gamma: 4^-4 Kernel:rbf Train Time:0.462413
->
Cross Validation Accuracy = 95.4%
C: 4^1Gamma: 4^-3 Kernel:rbf Train Time:0.333594
->
Cross Validation Accuracy = 96.45%
C: 4^1Gamma: 4^-2 Kernel:rbf Train Time:0.371376
->
Cross Validation Accuracy = 95.6%
C: 4^1Gamma: 4^-1 Kernel:rbf Train Time:0.785944
->
Cross Validation Accuracy = 92.3%
C: 4^2Gamma: 4^-7 Kernel:rbf Train Time:0.617715
->
Cross Validation Accuracy = 93.15%
C: 4^2Gamma: 4^-6 Kernel:rbf Train Time:0.470435
->
Cross Validation Accuracy = 94.15%
C: 4^2Gamma: 4^-5 Kernel:rbf Train Time:0.341246
->
Cross Validation Accuracy = 95.5%
C: 4^2Gamma: 4^-4 Kernel:rbf Train Time:0.26416
->
Cross Validation Accuracy = 96.5%
C: 4^2Gamma: 4^-3 Kernel:rbf Train Time:0.307944

->
Cross Validation Accuracy = 96.25%
C: 4^2Gamma: 4^-2 Kernel:rbf Train Time:0.397333
->
Cross Validation Accuracy = 96%
C: 4^2Gamma: 4^-1 Kernel:rbf Train Time:0.567883
->
Cross Validation Accuracy = 93.2%
C: 4^3Gamma: 4^-7 Kernel:rbf Train Time:0.392471
->
Cross Validation Accuracy = 94%
C: 4^3Gamma: 4^-6 Kernel:rbf Train Time:0.309007
->
Cross Validation Accuracy = 95%
C: 4^3Gamma: 4^-5 Kernel:rbf Train Time:0.269268
->
Cross Validation Accuracy = 95.65%
C: 4^3Gamma: 4^-4 Kernel:rbf Train Time:0.299489
->
Cross Validation Accuracy = 96.95%
C: 4^3Gamma: 4^-3 Kernel:rbf Train Time:0.240649
->
Cross Validation Accuracy = 96.55%
C: 4^3Gamma: 4^-2 Kernel:rbf Train Time:0.287895
->
Cross Validation Accuracy = 95.35%
C: 4^3Gamma: 4^-1 Kernel:rbf Train Time:0.556557
->
Cross Validation Accuracy = 93.85%
C: 4^4Gamma: 4^-7 Kernel:rbf Train Time:0.304848
->
Cross Validation Accuracy = 94.65%
C: 4^4Gamma: 4^-6 Kernel:rbf Train Time:0.275535
->
Cross Validation Accuracy = 94.45%
C: 4^4Gamma: 4^-5 Kernel:rbf Train Time:0.252322
->
Cross Validation Accuracy = 96.1%
C: 4^4Gamma: 4^-4 Kernel:rbf Train Time:0.261733
->
Cross Validation Accuracy = 96.5%
C: 4^4Gamma: 4^-3 Kernel:rbf Train Time:0.251608
->

Cross Validation Accuracy = 96.4%
C: 4^4Gamma: 4^-2 Kernel:rbf Train Time:0.286598
->
Cross Validation Accuracy = 95.35%
C: 4^4Gamma: 4^-1 Kernel:rbf Train Time:0.570393
->
Cross Validation Accuracy = 94.85%
C: 4^5Gamma: 4^-7 Kernel:rbf Train Time:0.293758
->
Cross Validation Accuracy = 94.15%
C: 4^5Gamma: 4^-6 Kernel:rbf Train Time:0.356849
->
Cross Validation Accuracy = 95.7%
C: 4^5Gamma: 4^-5 Kernel:rbf Train Time:0.311351
->
Cross Validation Accuracy = 96.55%
C: 4^5Gamma: 4^-4 Kernel:rbf Train Time:0.280528
->
Cross Validation Accuracy = 96.7%
C: 4^5Gamma: 4^-3 Kernel:rbf Train Time:0.332891
->
Cross Validation Accuracy = 96.55%
C: 4^5Gamma: 4^-2 Kernel:rbf Train Time:0.378194
->
Cross Validation Accuracy = 96.2%
C: 4^5Gamma: 4^-1 Kernel:rbf Train Time:0.677143
->
Cross Validation Accuracy = 94.65%
C: 4^6Gamma: 4^-7 Kernel:rbf Train Time:0.306148
->
Cross Validation Accuracy = 94.95%
C: 4^6Gamma: 4^-6 Kernel:rbf Train Time:0.293762
->
Cross Validation Accuracy = 95.95%
C: 4^6Gamma: 4^-5 Kernel:rbf Train Time:0.455069
->
Cross Validation Accuracy = 96.65%
C: 4^6Gamma: 4^-4 Kernel:rbf Train Time:0.439083
->
Cross Validation Accuracy = 96.6%
C: 4^6Gamma: 4^-3 Kernel:rbf Train Time:0.410561
->
Cross Validation Accuracy = 96.6%

C: 4^6Gamma: 4^-2 Kernel:rbf Train Time:0.404952
->
Cross Validation Accuracy = 95.55%
C: 4^6Gamma: 4^-1 Kernel:rbf Train Time:0.676527
->
Cross Validation Accuracy = 94.75%
C: 4^7Gamma: 4^-7 Kernel:rbf Train Time:0.381684
->
Cross Validation Accuracy = 94.8%
C: 4^7Gamma: 4^-6 Kernel:rbf Train Time:0.498503
->
Cross Validation Accuracy = 96.25%
C: 4^7Gamma: 4^-5 Kernel:rbf Train Time:0.543878
->
Cross Validation Accuracy = 96.55%
C: 4^7Gamma: 4^-4 Kernel:rbf Train Time:0.554352
->
Cross Validation Accuracy = 96.15%
C: 4^7Gamma: 4^-3 Kernel:rbf Train Time:0.337768
->
Cross Validation Accuracy = 96.65%
C: 4^7Gamma: 4^-2 Kernel:rbf Train Time:0.374659
->
Cross Validation Accuracy = 95.25%
C: 4^7Gamma: 4^-1 Kernel:rbf Train Time:0.579096
->
best c 5 best g -2 and acc is 97.15

RBF kernel VS POLYNOMIAL Kernel :

RBF kernel for c = 4^5 and g = -2 has the best accuracy

Accuracy = 96.4% (1928/2000) (classification)
Testing Accuracy 96.4

COLLABORATORS:
SINDHUJHA SETHURAMAN
SHIVRANJINI RAJAGOPAL