Supervised Learning Coursework 1

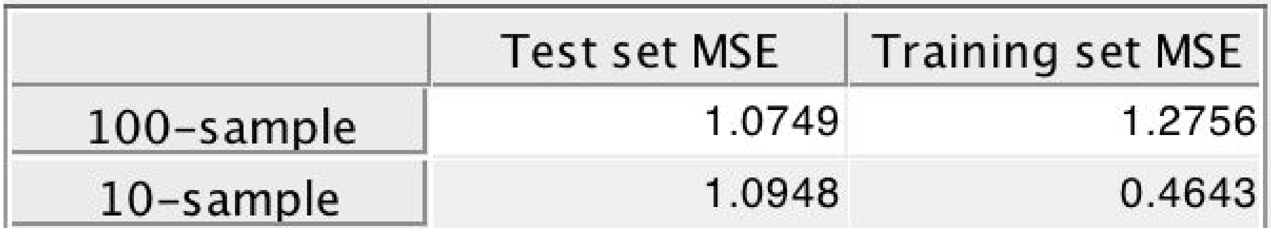
Jack Shipway and Marcin Cuber

15th January 2015

Linear Regression

Exercise 1 - Least Squares Regression: effect of the training set size.

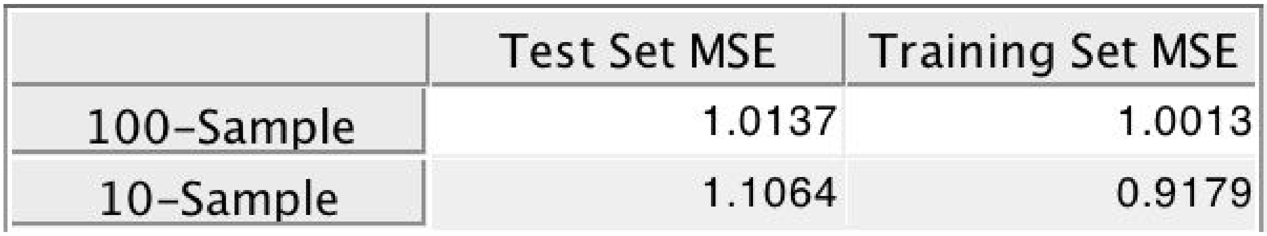
- **a.** See part 1(a) in ***coursework_part1.m.***
- **b.** See part 1(b) in ***coursework_part1.m*** and ***Figure 1.***
- **c.** See part 1(c) in ***coursework_part1.m*** and ***Figure 1.***



	Test set MSE	Training set MSE
100-sample	1.0749	1.2756
10-sample	1.0948	0.4643

Figure 1: *Test and Training set MSE[1] for 100 and 10-sample training sets, calculated for a single iteration.*

- **d.** See Part 1(d) in ***coursework_part1d.m*** and ***Figure 2.***



	Test Set MSE	Training Set MSE
100-Sample	1.0137	1.0013
10-Sample	1.1064	0.9179

Figure 2: *Average Test and Training set MSE across 10 and 100-sample training sets, calculated over 200 iterations.*

[1] *Mean Squared Error will be referred to as MSE throughout this report.*

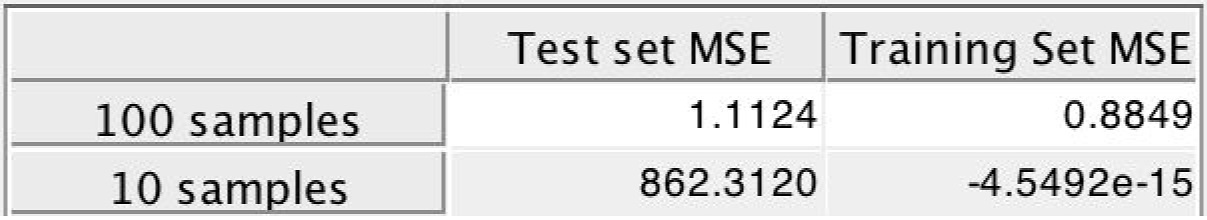
Observations

1. **What is the effect of the training set size on training and test set errors?**
   The test set errors for both training set sample-sizes are very similar when averaged over 200 iterations, despite the large difference in training set sample-size. The size of the training set therefore would not seem to significantly influence the MSE on its corresponding test set. We would expect the MSE on the training set of size 10 to fluctuate more than its equivalent 100-sample training set. In practice this is true, but we never achieve an MSE from the 100-sample training set smaller than the 10-sample version.

2. **What is larger, the training or the test set error? Explain.**
   The average training set error is in both cases, smaller than its corresponding test set error. This is because we are 'fitting' the training set data. Regardless of the fitting, there will be error generated when applying that fitting to unseen data which presents room for error.

Exercise 2 (Least squares regression - effect of the dimensionality)

a. See Part 2(a) in ***coursework_part2.m*** and ***Figure* 3.**
b. See Part 2(b) in ***coursework_part2.m*** and ***Figure* 3.**



	Test set MSE	Training Set MSE
100 samples	1.1124	0.8849
10 samples	862.3120	-4.5492e-15

Figure 3: *Average Test and Training set MSE across 10 and 100-sample training sets for 10-dimensional data.*

Observations

1. **Provide an interpretation of your new table in light of the fact that the data is now 10-dimensional.**

There is a greater margin for error here as there is a greater number of dimensions compared to the number of samples. The test set error demonstrates this because the training set data minimally represents the complexity of the total test set. This does however mean that it is considerably easier to fit a function through the training set, which is why the training set error is so low.

Exercise 3 (Ridge Regression)

1. Prove that solving optimisation problem (6) yields $\boldsymbol{w}^{*} = (\mathbf{X}^{T}\mathbf{X} + \gamma\, l\mathbf{I})^{-1}\mathbf{X}^{T}\mathbf{y}$.

$$\underline{w}^{*} = argmin_{\underline{w}}\ \gamma \underline{w}^{T}\underline{w} + \frac{1}{l}\sum_{i=1}^{l}(\underline{x}_{i}^{T}\underline{w} - y_{i})^{2}$$

$$First,\ note\ that:\ \gamma \underline{w}^{T}\underline{w} = \gamma\sum_{i=1}^{n} w_{i}^{2}$$

$$\therefore\ \nabla(\gamma\sum_{i=1}^{n} w_{i}^{2}) = 2\gamma w$$

$$Second:\ \nabla(\frac{1}{l}\sum_{i=1}^{l}(\underline{x}_{i}^{T}\underline{w} - y_{i})^{2}) = \frac{2}{l}\sum_{i=1}^{l}(x_{i}^{T}\underline{w} - y_{i})\, x_{i} = \frac{2}{l}(\underline{X}^{T}\underline{X}\,\underline{w} - \underline{X}^{T}\underline{y})$$

$$Set\ \ \nabla(\gamma \underline{w}^{T}\underline{w} + \frac{1}{l}\sum_{i=1}^{l}(\underline{x}_{i}^{T}\underline{w} - y_{i})^{2}) = 0$$

$$2\gamma \underline{w} + \frac{2}{l}(\underline{X}^{T}\underline{X}\,\underline{w} - \underline{X}^{T}\underline{y}) = 0$$

$$\gamma l\underline{w} + (\underline{X}^{T}\underline{Xw} - \underline{X}^{T}\underline{y}) = 0$$

$$\underline{X}^{T}\underline{X}\,\underline{w} + \gamma l\underline{w} = \underline{X}^{T}\underline{y}$$

$$(\underline{X}^{T}\underline{X} + \gamma l\,\underline{I})\underline{w} = \underline{X}^{T}\underline{y}$$

$$\therefore\ \underline{w} = (\underline{X}^{T}\underline{X} + \gamma l\,\underline{I})^{-1}\underline{X}^{T}\underline{y} = \underline{w}^{*}$$

□

2. Prove that $\mathbf{X'X} + \gamma\, l\mathbf{I}$ is a positive definite matrix.

For a matrix A to be positive definite, we must show that for any vector v ⊂ R^{n}, that $v^{t}Av > 0$. Using this, we take $v^{t}(\mathbf{X'X} + \gamma\, l\mathbf{I})v = v^{t}\mathbf{X'X}v + v^{t}(\gamma\, l\mathbf{I})v > 0$.

Now, we know that $v^{t}\mathbf{X'X}v$ is positive semi-definite, *l* is a positive integer because it denotes the number of data samples (which we assume to be at least 1), and γ is assumed to be positive. Therefore we are summing two positive definite matrices. To prove that the sum of a positive definite and a positive semidefinite matrix is positive definite, we do the following:

$$\textit{Let } A \textit{ be a positive definite matrix and } B \textit{ be positive semi}-\textit{definite, then}:$$
$$\forall x \in \mathbb{R}^n,\ x^T A x > 0,\ x^T B x \geq 0$$
$$x^T A x > 0,\ x^T B x \geq 0 \Rightarrow x^T A x + x^T B x > 0$$
$$x^T A x + x^T B x = x^T (A + B)\, x > 0,\ \textit{by the distributive law of matrix multiplication}$$
$$\therefore \textit{Let } C = A + B,\ \textit{then } C \textit{ satisfies the definition of being positive definite.}$$

□

Exercise 4 (Effect of the regularisation parameter)

a. See Part 4(a) in ***coursework_part4.m***, *and* ***Figure 4***.

i. Mean squared error on a 500-sample test set with a 100-sample training set, for a single iteration, over all γ.

[46.1325 46.1325 46.1316 46.1232 46.0398 45.293 44.7649 186.312 529.419 621.689]

ii. Mean squared error on the 100-sample training set, for a single iteration, over all γ.

[8.3424 8.3424 8.3424 8.3424 8.3426 8.3558 9.4420 38.6003 106.2121 125.0391]

iii. Mean squared error on a 500-sample test set, using a 10-sample training set, over a single iteration, over all γ.

[1.3397 1.3358 1.2974 0.9944 0.2496 0.0920 0.2435 0.5360 0.6216 0.6324]

iv. Mean squared error on the 10-sample training set, over a single iteration, over all γ.

[0.0000 0.0000 0.0000 0.0035 0.0649 0.2668 2.7878 9.0346 11.7233 12.1063]

b. See Part 4(b) in ***coursework_part4.m***, and ***Figure 4.***

c. See Part 4(c) in ***coursework_part4c.m***, ***Figure 5.***

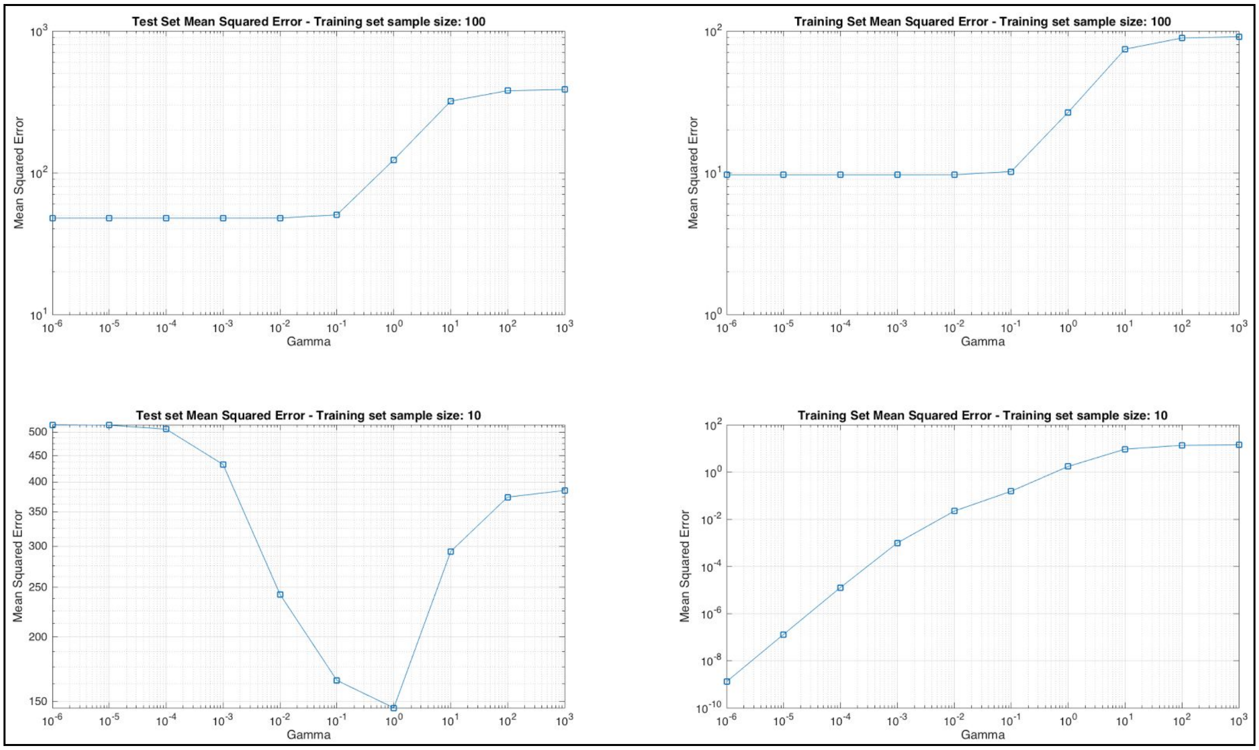


Figure 4: *Test and training set mean squared error for 100 and 10-sample training sets, over each gamma value, over a single iteration.*

Observations

1. **Do your results suggest a method to set the regularisation parameter to minimise the test set error?**

There are no immediately obvious observations from the graph that would indicate whether one particular regularisation parameter is better than another. We do notice that the MSE for the training set with sample size 10, is a lot smaller than that of the one with sample size of 100 as a consequence of our results in exercise 2. However, the test set errors are so different that there is no clear indication of which regularisation parameter would yield the best result. There are obvious minima in the graphs pertaining to the test sets, but they do not in any way correlate to the results of the training set MSEs. Therefore, we can conclude that the training set error is not sufficiently good guidance in order to select the optimal $\gamma \in (10^{-6} : 10^{3})$.

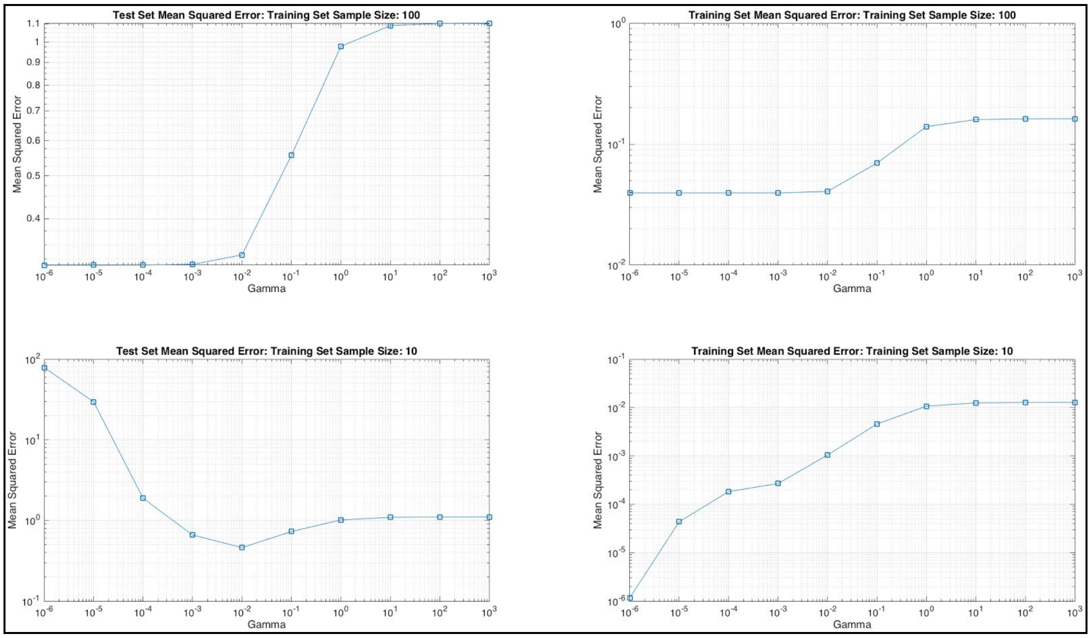


Figure 5: *Plots of the average training and test set errors for 100 and 10-sample training sets, over each gamma value, performed over 200 iterations.*

Exercise 5 (Tuning the regularisation parameter using a validation set)

a. See Part 5(a) in ***coursework_part5a.m***, ***Figure 6*** and ***Figure 7.***

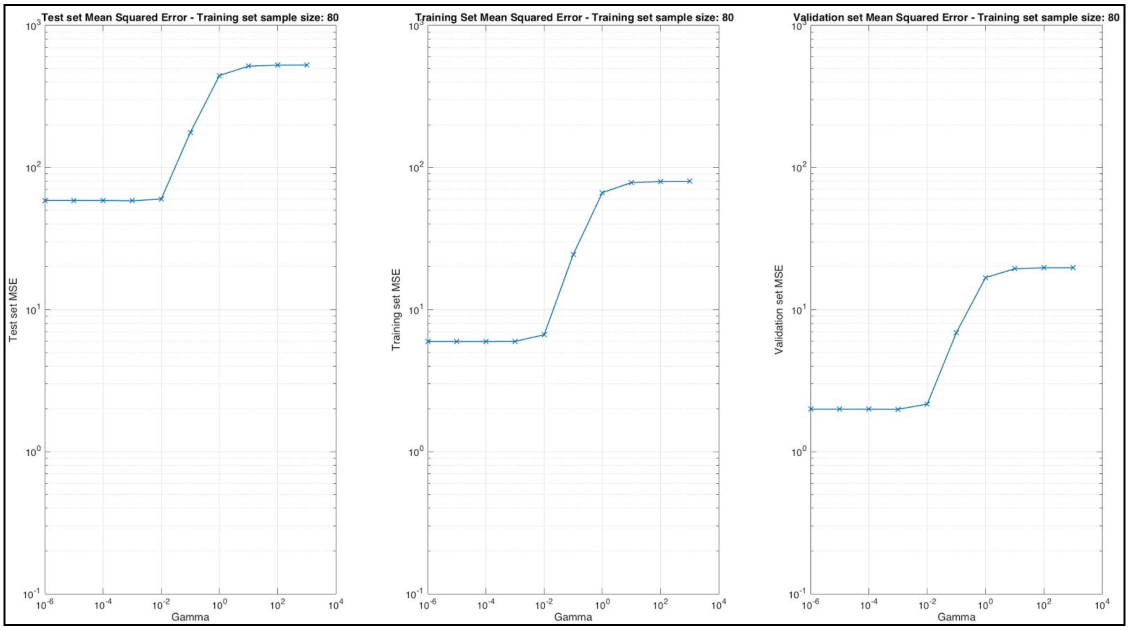
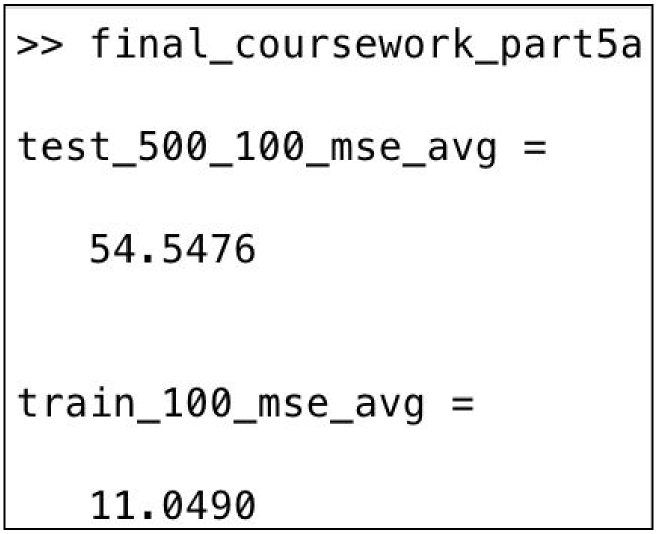


Figure 6: Average test, training and validation set mean squared errors, using a training set of size 80, over 200 iterations



```
>> final_coursework_part5a

test_500_100_mse_avg =

   54.5476


train_100_mse_avg =

   11.0490
```

Figure 7: Average test and training set mean squared errors having performed ridge regression on the original 100-sample training set, over 200 iterations.

b. See part 5(b) in ***coursework_part5b.m***, ***Figure 8*** and ***Figure 9***.

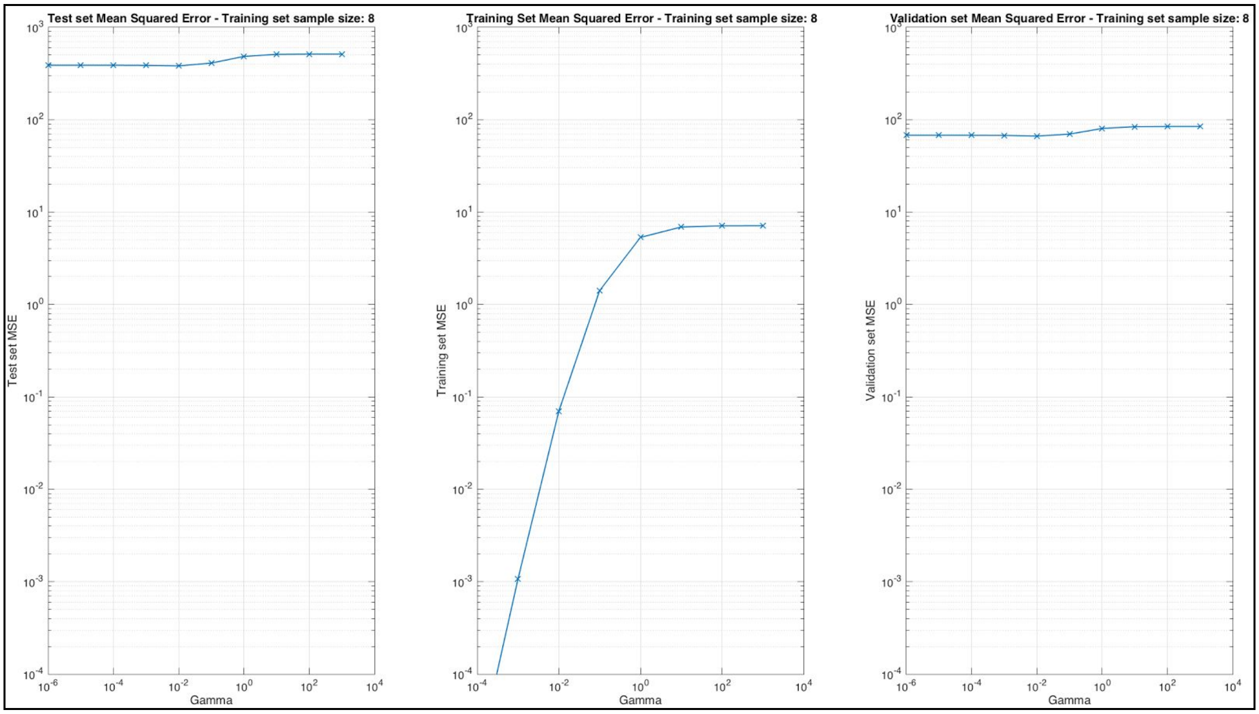
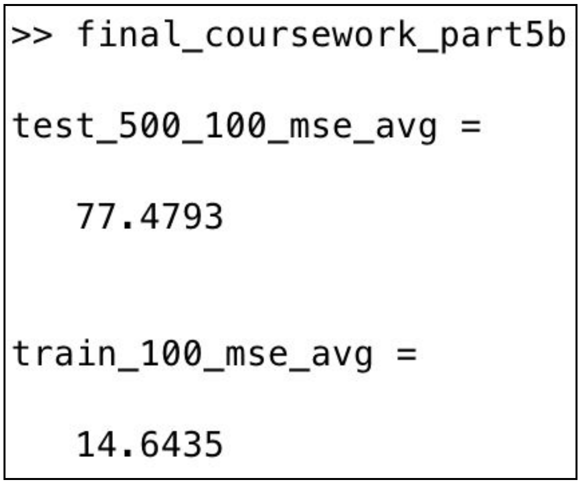


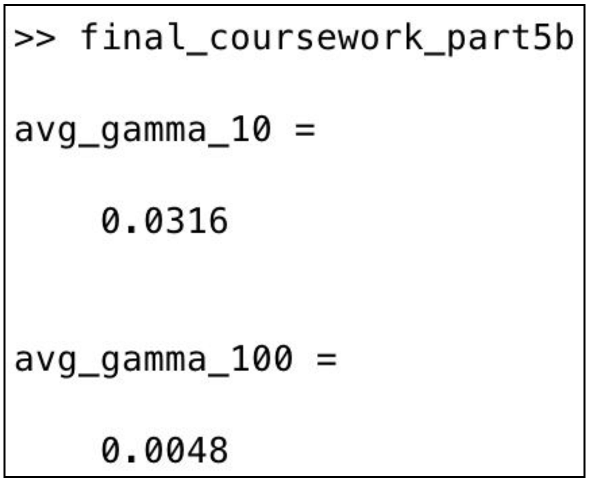
Figure 8: *Average training, validation and test set errors for a training set of size 10.*

c. See parts 5(a) and 5(b) in ***coursework_part5a | b*** and ***Figure 10.***



```
>> final_coursework_part5b

test_500_100_mse_avg =

   77.4793


train_100_mse_avg =

   14.6435
```

Figure 9: *Average test and training set mean squared errors having performed ridge regression on the original 10-sample training set, over 200 iterations.*



```
>> final_coursework_part5b

avg_gamma_10 =

    0.0316


avg_gamma_100 =

    0.0048
```

Figure 10: *Average gamma value over 200 iterations, for 10 and 100-sample training sets respectively.*

Observations

1. **In which case is the average gamma value larger and why?**

The average gamma value for the 10-sample training set is found to be larger than its equivalent 100-sample training set gamma. The reasoning is analogous to drawing a straight line through 10 data points versus 100 points on 2-dimensional data. The latter is going to be far more representative of the whole data set (at least in this case where the total sample size is 510 or 600), and will thus be easier to draw an accurate line. We therefore need a greater 'cancelling' term or dampener, to counteract the overfitting of noise.

2. **Think about what would happen with the 1-dimensional datasets from exercise 1.**

We would predict that the average gamma values would be lower for the 1-dimensional data. On the contrary, perhaps against intuition, we find that in both cases, as the number of dimensions increases, the gamma value substantially increases! It does however remain true that the gamma value for the 10-sample training set is larger than the 100-sample gamma value.

Exercise 6 (Tuning the regularisation parameter using cross-validation)

a. See Exercise 6(a) in ***coursework_part6.m*** and ***Figure 11.***

b. See Exercise 6(b) in ***coursework_part6.m*** and ***Figure 12.***

The average *score* given by performing K-Fold cross-validation on a 100-sample training set was good, ranging from $0.2 < cvs < 1$ approximately. The resulting test set error was also very good, from $60 < test_e < 150$. There was however no clear indication as to which value of the regularisation parameter should be chosen to give the best test set error, other than merely choosing the smallest one.

The 10-sample training set resulted in considerable higher error, ranging from $180 < \text{test}_e < 280$, despite the fact that the cross-validation score remained between $0 < \text{cvs} < 1$. There is a more obvious minima as far as the regularisation parameter is concerned (at 10^{-2}), but this is not the value that gives the optimum test set error (10^{-1}). It is however, considerably better than previous efforts.

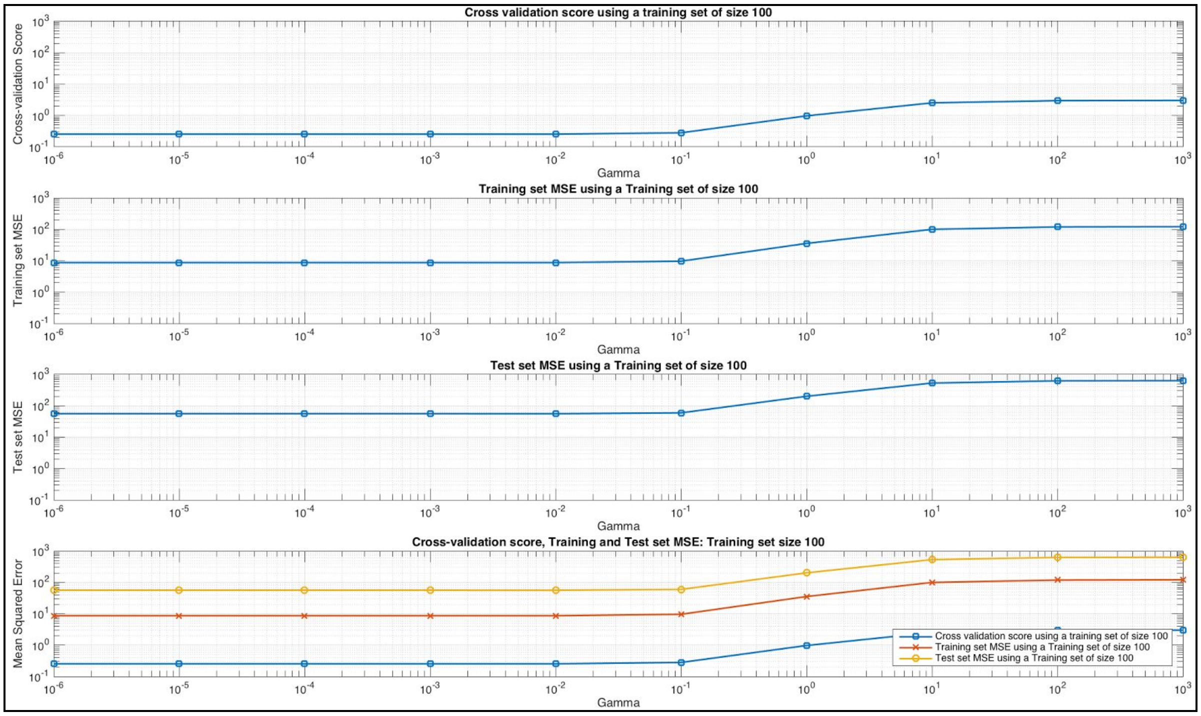


Figure 11: *Plot of the cross-validation score superimposed upon plots of the training and test set MSEs over each value of* $\gamma \in \{10^{-6},\dots, 10^{3}\}$, *using a training set of size* 100.

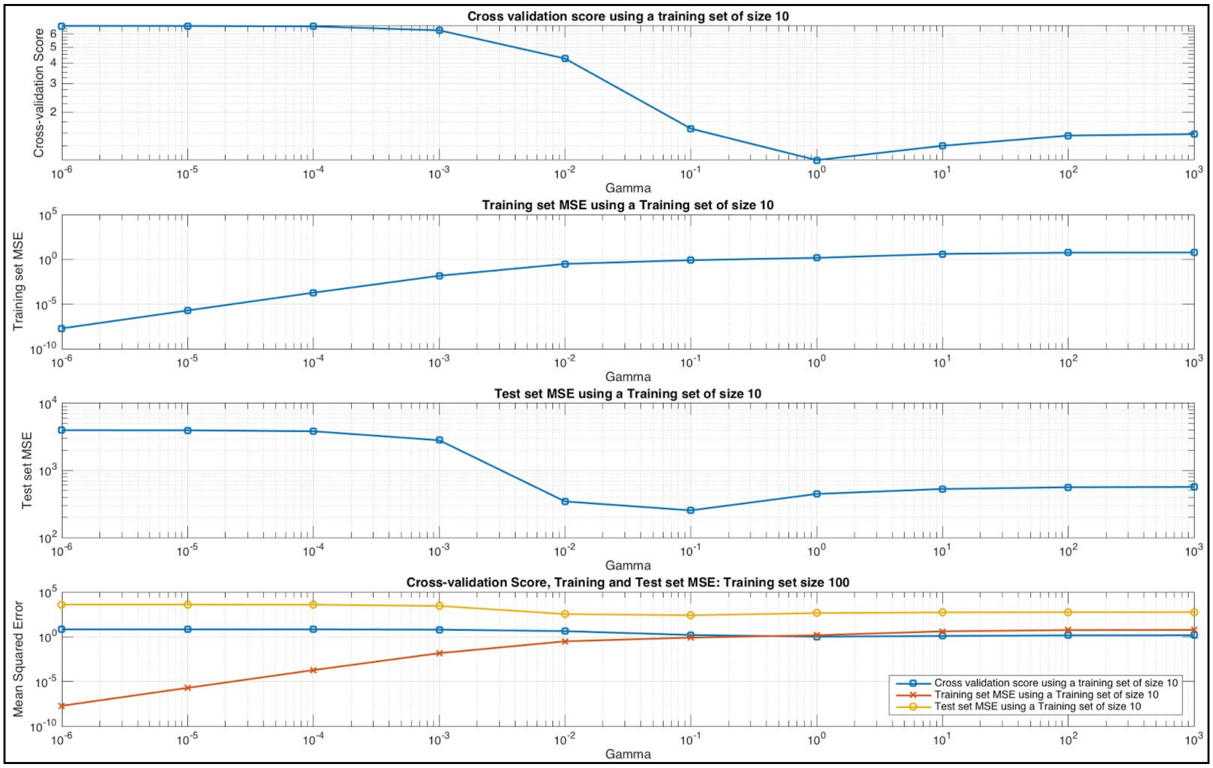
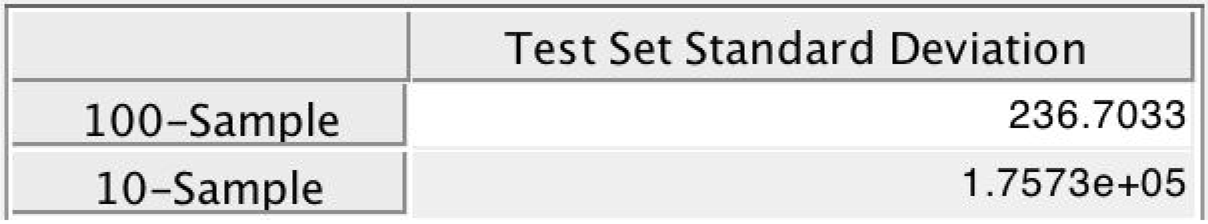


Figure 12: *Plot of the cross-validation score superimposed upon plots of the training and test set MSEs over each value of* γ ∈ {10^{-6},..., 10^3}, *using a training set of size 10.*

Exercise 7 (Comparison of ɣ tuning methods)

a. See ***coursework_part7a.m*** and ***Figures* 13** and **14.**



	Test Set Standard Deviation
100-Sample	236.7033
10-Sample	1.7573e+05

Figure 13: *Table of the standard deviations of the test set MSEs, for both 10 and 100-sample training sets, taken over 200 iterations.*

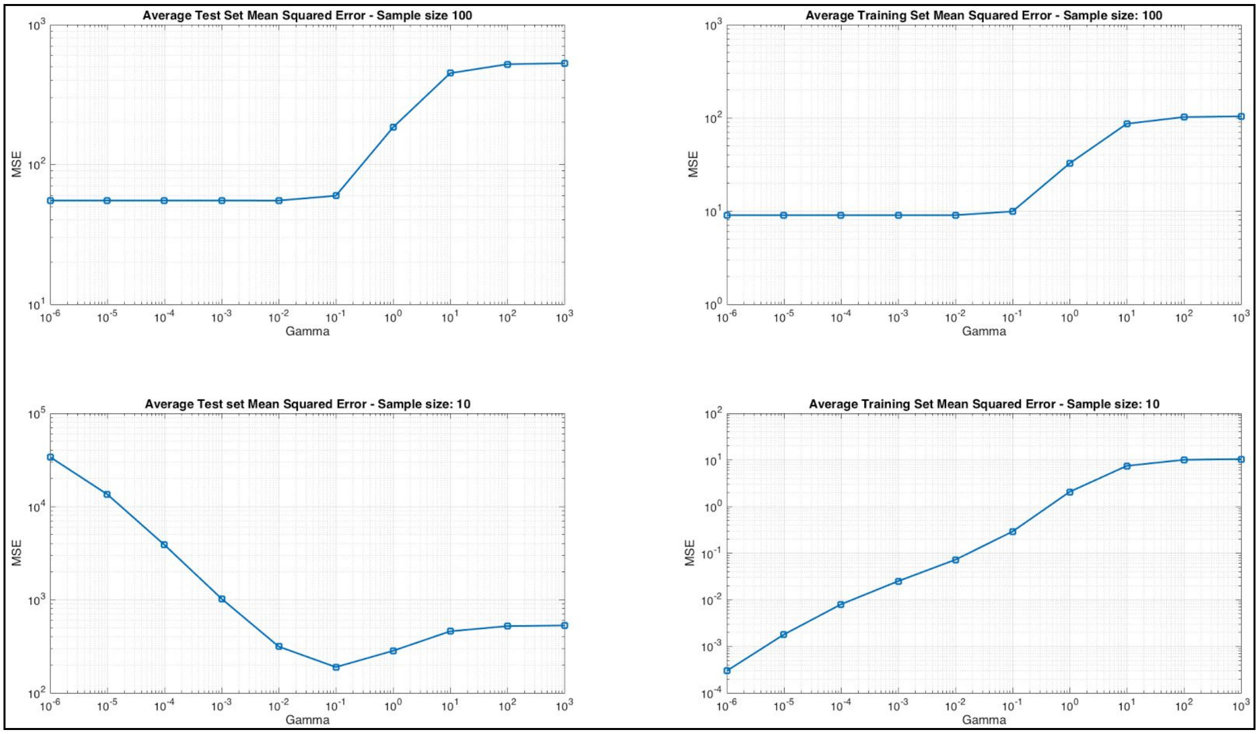
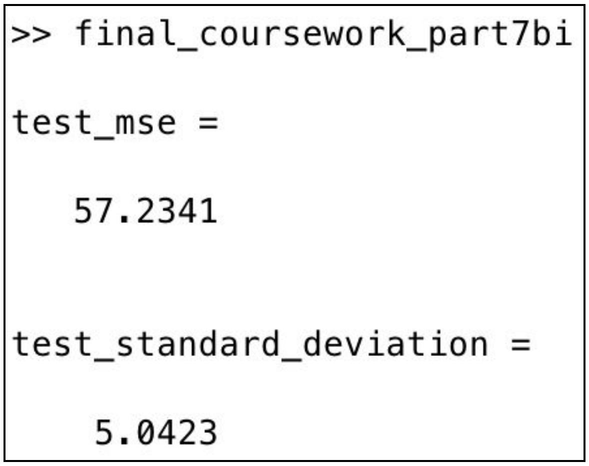
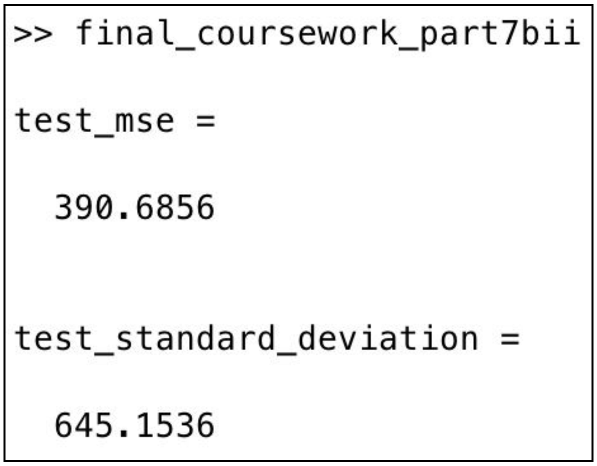


Figure 14: *Comparison of the average training and test set MSEs for both 100 and 10-sample training sets, over 200 iterations.*

b. See ***coursework_part7bi.m***, ***coursework_part7bii.m***, ***Figures* 15** and **16**.



```
>> final_coursework_part7bi

test_mse =

   57.2341


test_standard_deviation =

    5.0423
```



```
>> final_coursework_part7bii

test_mse =

  390.6856


test_standard_deviation =

  645.1536
```

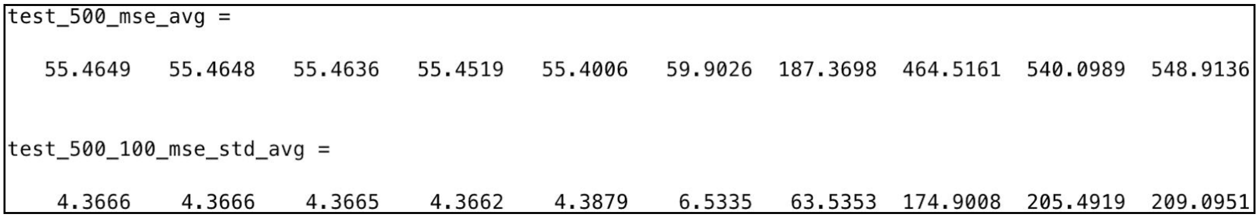
Figure 15: *Average test set MSE and standard deviation computed by performing Ridge Regression on the full* ***100****-sample training set, using the gamma value that resulted in the*

Figure 16: *Average test set MSE and standard deviation computed by performing Ridge Regression on the full* ***10****-sample training set using the gamma value that resulted in the*

minimum validation set error.

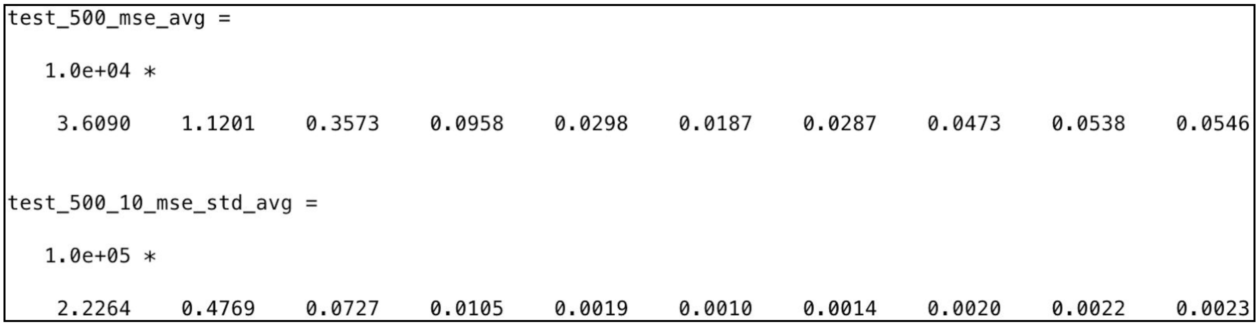
minimum validation set error.

c. See ***coursework_part7c.m*** and ***Figures* 17** and **18**.



```
test_500_mse_avg =

   55.4649   55.4648   55.4636   55.4519   55.4006   59.9026  187.3698  464.5161  540.0989  548.9136


test_500_100_mse_std_avg =

    4.3666    4.3666    4.3665    4.3662    4.3879    6.5335   63.5353  174.9008  205.4919  209.0951
```

***Figure* 17:** *Average test set MSE and standard deviation computed by performing Ridge Regression on the full 100-sample training set, for all values of gamma.*



```
test_500_mse_avg =

   1.0e+04 *

    3.6090    1.1201    0.3573    0.0958    0.0298    0.0187    0.0287    0.0473    0.0538    0.0546


test_500_10_mse_std_avg =

   1.0e+05 *

    2.2264    0.4769    0.0727    0.0105    0.0019    0.0010    0.0014    0.0020    0.0022    0.0023
```

***Figure* 18:** *Average test set MSE and standard deviation computed by performing Ridge Regression on the full 10-sample training set, for all values of gamma.*

Observations

7a

The training set error gives no indication as to what regularisation parameter we should select to achieve optimal test set MSE for the 100-sample training set. ***Figure* 13** (top-right) maintains a consistent MSE over the first 5 values, and then rapidly increases. There is also no obvious minimum test set error and so we can infer that any choice out of the first 5 gamma values would lead to a low test set MSE. The 10-sample set is quite odd in that there is an inflection in the graph in ***Figure* 14** (bottom-right) each time we run the experiment which always corresponds to the optimum test set MSE. It is hard to notice this as the MSE for smaller values of gamma result in far lower training set MSEs, so it is almost as if we must ignore the first few values and look for this point of inflection in order to determine gamma more effectively.

7c

As seen in ***Figure* 17,** the mean squared error has an obvious dip at the fifth value of the regularisation parameter, but there is no way that we could have inferred this from the cross-validation score where the minimum MSE manifests itself at all of the first 5 values. This would suggest that no matter the choice of regularisation parameter (from this range), we would expect to see a good test set performance, which is true (the standard deviation of the MSEs within this range is very low indeed), although there is no way of inferring the optimal.

As for the 10-sample training set we see far higher MSE values, regardless of the regularisation parameter, but there seems to be a more obvious way in which to select the regularisation parameter. As shown in ***Figure* 18**, the cross-validation score at gamma = 10^{-1}, corresponds to the minimum test set error at gamma = 10^{-1}. To confirm that this is an accurate selection method, we would want to repeat the experiment over different training set sample sizes between 10 and 100. This would enable us to see at what point the selection of gamma becomes a less obvious procedure, and contrastingly, how slowly or rapidly the test set error increases as the training set sample size decreases.

Boston Housing and kernels

*   For exercises 9 and 10, we kindly ask that you run ***coursework_part9.m*,** then ***coursework_part10.m***, and finally, ***coursework_part9_10_table.m*** so that the table of results contains all of the necessary values.

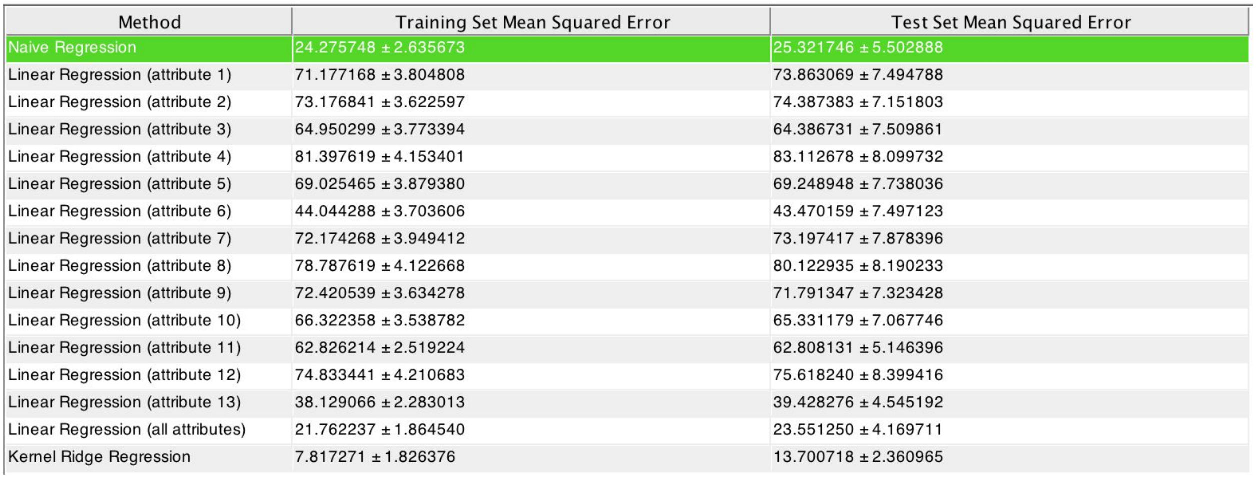
*   Each iteration for our kernel ridge regression algorithm takes approximately 30 seconds with our current efforts - meaning that 20 iterations takes ~10 minutes to perform; we hope that is satisfactory.

Exercise 8 (Load the file 'boston.mat' into Matlab.)

a. See Exercise 8 in ***coursework_part9.m.***

Exercise 9 (Baseline versus full linear regression)

a. See Part 9(a) in ***coursework_part9.m*,** and ***Figure* 19.**

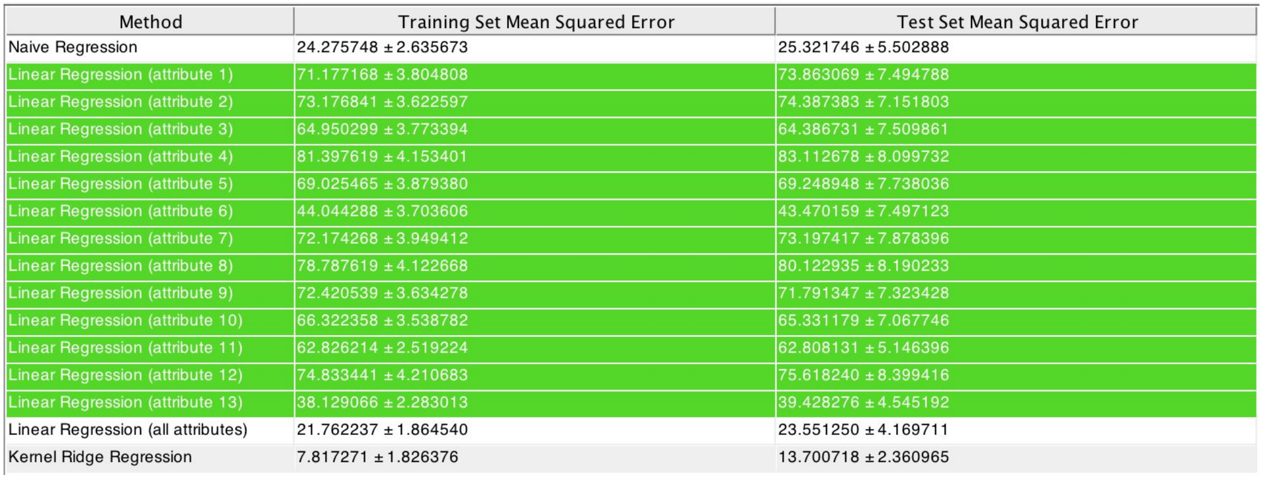


Method	Training Set Mean Squared Error	Test Set Mean Squared Error
Naive Regression	24.275748 ± 2.635673	25.321746 ± 5.502888
Linear Regression (attribute 1)	71.177168 ± 3.804808	73.863069 ± 7.494788
Linear Regression (attribute 2)	73.176841 ± 3.622597	74.387383 ± 7.151803
Linear Regression (attribute 3)	64.950299 ± 3.773394	64.386731 ± 7.509861
Linear Regression (attribute 4)	81.397619 ± 4.153401	83.112678 ± 8.099732
Linear Regression (attribute 5)	69.025465 ± 3.879380	69.248948 ± 7.738036
Linear Regression (attribute 6)	44.044288 ± 3.703606	43.470159 ± 7.497123
Linear Regression (attribute 7)	72.174268 ± 3.949412	73.197417 ± 7.878396
Linear Regression (attribute 8)	78.787619 ± 4.122668	80.122935 ± 8.190233
Linear Regression (attribute 9)	72.420539 ± 3.634278	71.791347 ± 7.323428
Linear Regression (attribute 10)	66.322358 ± 3.538782	65.331179 ± 7.067746
Linear Regression (attribute 11)	62.826214 ± 2.519224	62.808131 ± 5.146396
Linear Regression (attribute 12)	74.833441 ± 4.210683	75.618240 ± 8.399416
Linear Regression (attribute 13)	38.129066 ± 2.283013	39.428276 ± 4.545192
Linear Regression (all attributes)	21.762237 ± 1.864540	23.551250 ± 4.169711
Kernel Ridge Regression	7.817271 ± 1.826376	13.700718 ± 2.360965

Figure 19: *Average training and test set MSE plus or minus the standard deviation, for naive regression (highlighted), over 20 iterations.*

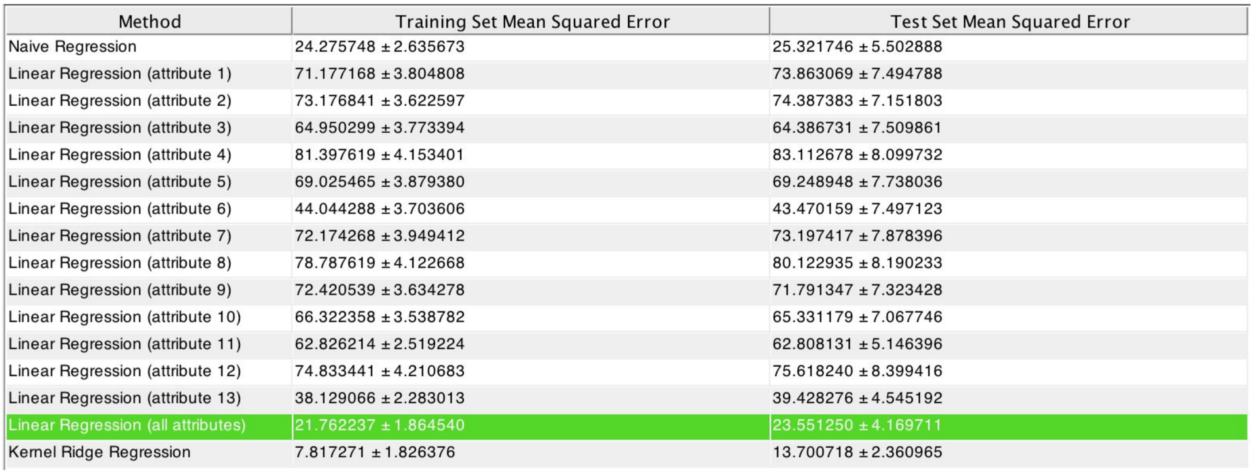
b. See Part 9(b) in ***coursework_part9.m***, and **Figure 20.**

c. See ***courswork_part9.m***, and **Figure 21.**



Method	Training Set Mean Squared Error	Test Set Mean Squared Error
Naive Regression	24.275748 ± 2.635673	25.321746 ± 5.502888
Linear Regression (attribute 1)	71.177168 ± 3.804808	73.863069 ± 7.494788
Linear Regression (attribute 2)	73.176841 ± 3.622597	74.387383 ± 7.151803
Linear Regression (attribute 3)	64.950299 ± 3.773394	64.386731 ± 7.509861
Linear Regression (attribute 4)	81.397619 ± 4.153401	83.112678 ± 8.099732
Linear Regression (attribute 5)	69.025465 ± 3.879380	69.248948 ± 7.738036
Linear Regression (attribute 6)	44.044288 ± 3.703606	43.470159 ± 7.497123
Linear Regression (attribute 7)	72.174268 ± 3.949412	73.197417 ± 7.878396
Linear Regression (attribute 8)	78.787619 ± 4.122668	80.122935 ± 8.190233
Linear Regression (attribute 9)	72.420539 ± 3.634278	71.791347 ± 7.323428
Linear Regression (attribute 10)	66.322358 ± 3.538782	65.331179 ± 7.067746
Linear Regression (attribute 11)	62.826214 ± 2.519224	62.808131 ± 5.146396
Linear Regression (attribute 12)	74.833441 ± 4.210683	75.618240 ± 8.399416
Linear Regression (attribute 13)	38.129066 ± 2.283013	39.428276 ± 4.545192
Linear Regression (all attributes)	21.762237 ± 1.864540	23.551250 ± 4.169711
Kernel Ridge Regression	7.817271 ± 1.826376	13.700718 ± 2.360965

Figure 20: *Average training and test set MSE plus or minus the standard deviation for individual attribute Linear Regression (highlighted), over 20 iterations.*



Method	Training Set Mean Squared Error	Test Set Mean Squared Error
Naive Regression	24.275748 ± 2.635673	25.321746 ± 5.502888
Linear Regression (attribute 1)	71.177168 ± 3.804808	73.863069 ± 7.494788
Linear Regression (attribute 2)	73.176841 ± 3.622597	74.387383 ± 7.151803
Linear Regression (attribute 3)	64.950299 ± 3.773394	64.386731 ± 7.509861
Linear Regression (attribute 4)	81.397619 ± 4.153401	83.112678 ± 8.099732
Linear Regression (attribute 5)	69.025465 ± 3.879380	69.248948 ± 7.738036
Linear Regression (attribute 6)	44.044288 ± 3.703606	43.470159 ± 7.497123
Linear Regression (attribute 7)	72.174268 ± 3.949412	73.197417 ± 7.878396
Linear Regression (attribute 8)	78.787619 ± 4.122668	80.122935 ± 8.190233
Linear Regression (attribute 9)	72.420539 ± 3.634278	71.791347 ± 7.323428
Linear Regression (attribute 10)	66.322358 ± 3.538782	65.331179 ± 7.067746
Linear Regression (attribute 11)	62.826214 ± 2.519224	62.808131 ± 5.146396
Linear Regression (attribute 12)	74.833441 ± 4.210683	75.618240 ± 8.399416
Linear Regression (attribute 13)	38.129066 ± 2.283013	39.428276 ± 4.545192
Linear Regression (all attributes)	21.762237 ± 1.864540	23.551250 ± 4.169711
Kernel Ridge Regression	7.817271 ± 1.826376	13.700718 ± 2.360965

Figure 21: *Average training and test set MSE plus or minus the standard deviation for all attributes combined (highlighted), over 20 iterations.*

Observations

We notice that this method outperforms any of the other individual regressors. This is expected because it takes into account more features of the data set and is thus more representative of its samples, leading to a more accurate prediction.

Exercise 10 (Kernel Ridge Regression)

a.

i. See ***kridgereg.m.***

ii. **Why does our implementation correctly compute** α **?**

We are talking about primal and dual optimisation, and so we can show why alpha is computed correctly through an example of regularised least squares.

Given a matrix $X \in \Re^{n\times d}$, representing n, d-dimensional samples, and a sample label vector $y \in \Re^{n}$, the regularised least squares can be written as: $argmin_w \, \lambda w^T w + ||Xw - y||^2$, where $w \in \Re^{d}$ and λ is a regularisation parameter.

Minimise over $w = (X^T X + \lambda I)^{-1} X^T y$, and its minimum is: $y^T y - y^T X (X^T X + \lambda I)^{-1} X^T y$.

We introduce the slack variable $\xi = Xw - y$, yielding the dual optimisation problem:

$$argmax_\alpha \, 2\alpha^T y - \tfrac{1}{\lambda}\, \alpha^T (XX^T + \lambda I)\alpha \text{, where } \alpha \in \Re^{n}$$

Maximise over $\alpha = \lambda (XX^T + \lambda I)^{-1} y$, and its maximum is: $\lambda y^T (XX^T + \lambda I)^{-1} y$.

The primal solution is then given by the KKT condition: $w = \frac{1}{\lambda} X^T \alpha$

We relate the inverses of $(XX^T + \lambda I)$ and $(X^T X + \lambda I)$ and arrive at the equation:

$$\lambda(XX^T + \lambda I)^{-1} = I - X(\lambda I + X^T X)^{-1} X^T$$

We have recovered that the optimal solutions to the primal and dual problems are equivalent, thus we know that the duality gap is zero. We can therefore conclude that the optimisation of the primal and dual problems are equivalent, both in terms of the solution and time complexity, although we have not shown that they are the same for time complexity. This demonstrates that our implementation correctly computes alpha, as we have shown that our algorithm optimises the same problem as that which computes alpha in the exercises.

□

- **b.** See ***dualcost.m.***
- **c.** See ***coursework_part10.m.***
  - i. See ***Figure*** **22.**
  - ii. Run ***coursework_part10.m.***
- **d.** Run ***coursework_part9_10.m*** and ***Figure*** **23.**

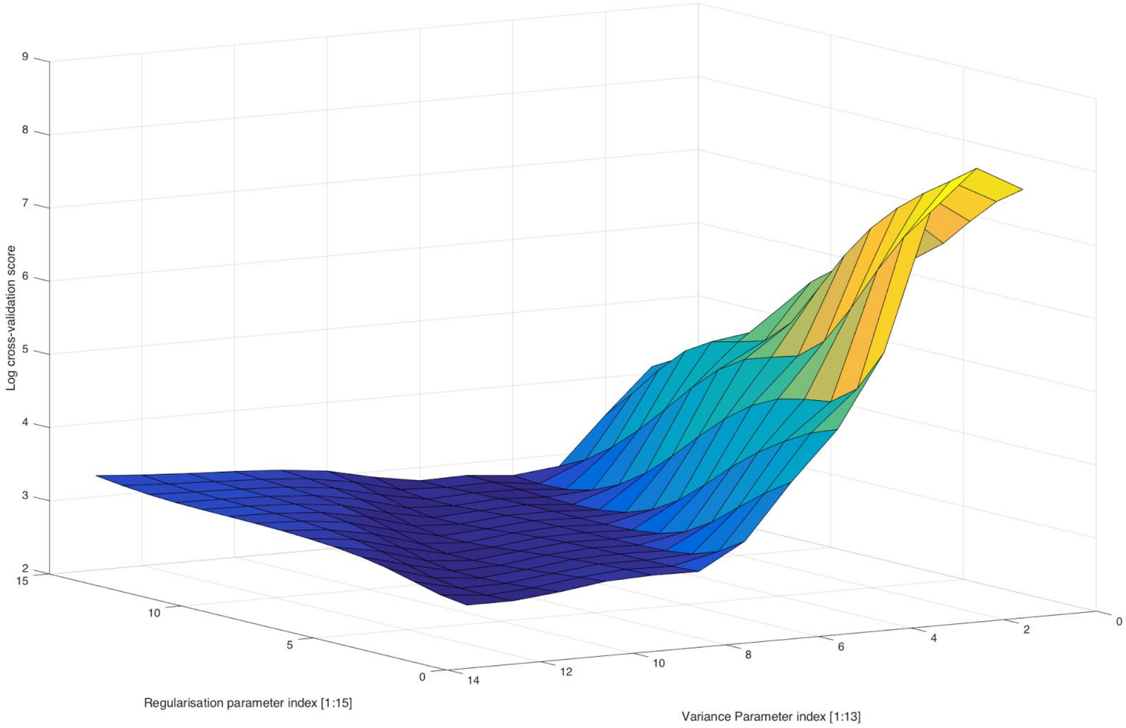
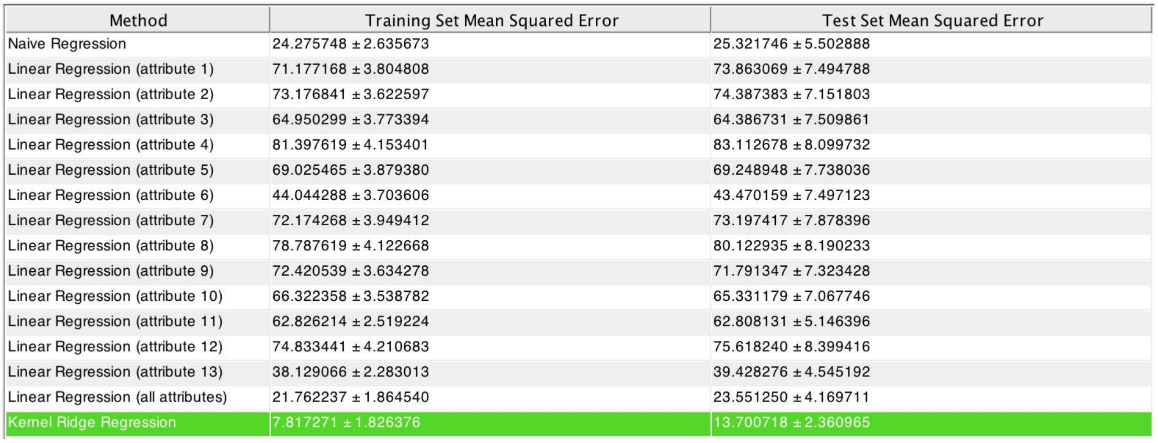


Figure 22: *Average cross-validation error as a function of gamma and sigma, calculated over 20 iterations.*



Method	Training Set Mean Squared Error	Test Set Mean Squared Error
Naive Regression	24.275748 ± 2.635673	25.321746 ± 5.502888
Linear Regression (attribute 1)	71.177168 ± 3.804808	73.863069 ± 7.494788
Linear Regression (attribute 2)	73.176841 ± 3.622597	74.387383 ± 7.151803
Linear Regression (attribute 3)	64.950299 ± 3.773394	64.386731 ± 7.509861
Linear Regression (attribute 4)	81.397619 ± 4.153401	83.112678 ± 8.099732
Linear Regression (attribute 5)	69.025465 ± 3.879380	69.248948 ± 7.738036
Linear Regression (attribute 6)	44.044288 ± 3.703606	43.470159 ± 7.497123
Linear Regression (attribute 7)	72.174268 ± 3.949412	73.197417 ± 7.878396
Linear Regression (attribute 8)	78.787619 ± 4.122668	80.122935 ± 8.190233
Linear Regression (attribute 9)	72.420539 ± 3.634278	71.791347 ± 7.323428
Linear Regression (attribute 10)	66.322358 ± 3.538782	65.331179 ± 7.067746
Linear Regression (attribute 11)	62.826214 ± 2.519224	62.808131 ± 5.146396
Linear Regression (attribute 12)	74.833441 ± 4.210683	75.618240 ± 8.399416
Linear Regression (attribute 13)	38.129066 ± 2.283013	39.428276 ± 4.545192
Linear Regression (all attributes)	21.762237 ± 1.864540	23.551250 ± 4.169711
Kernel Ridge Regression	7.817271 ± 1.826376	13.700718 ± 2.360965

Figure 23: *Test and training set MSEs for each method of regression - results from kernel Ridge Regression are highlighted.*

Observations

As expected, we see from ***Figure 23*** that Kernel Ridge Regression considerably outperformed any of the other forms of linear regression that we tried.