# programming project 2

Duc Nguyen

October 23, 2017

## 1 Regularization

The following 5 figures show the results of doing regularized linear regression on 5 different data sets. For reference, we know that the MSE for the true function 3.78 for 100-10, 3.78 for 100-100 and 4.015 for 1000-100.

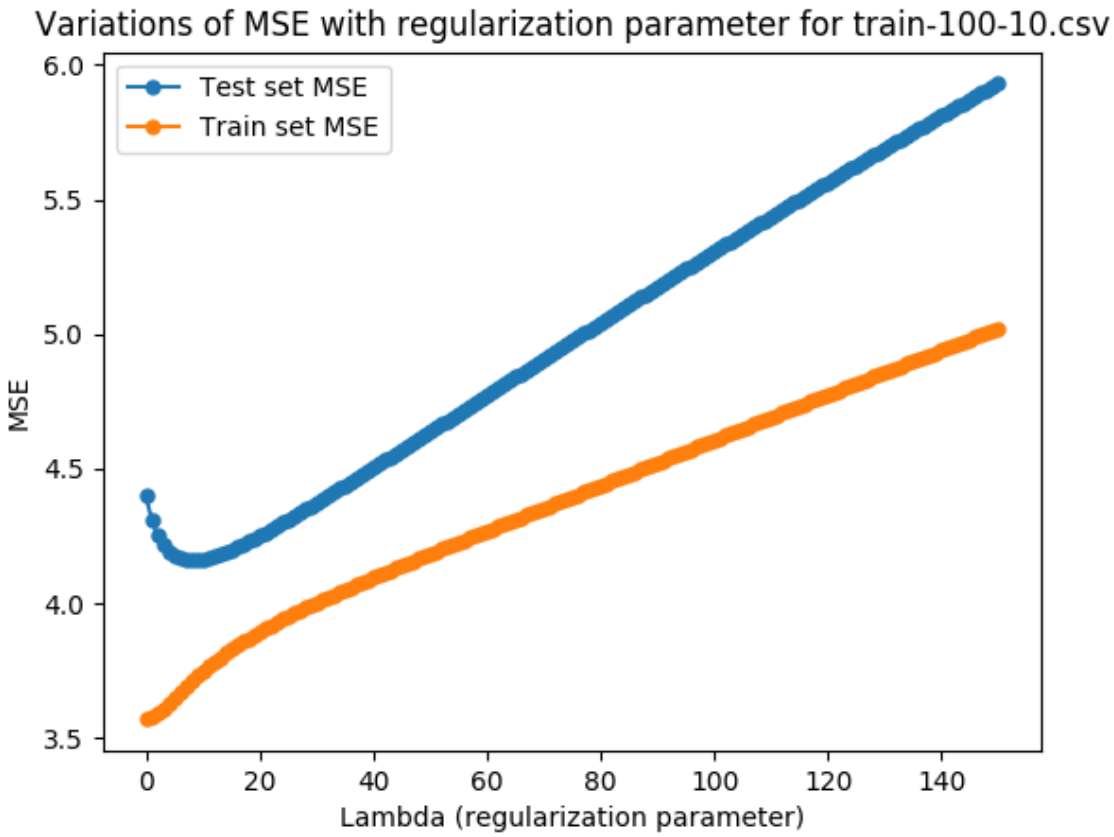


Figure 1: Dataset: train-100-10.csv

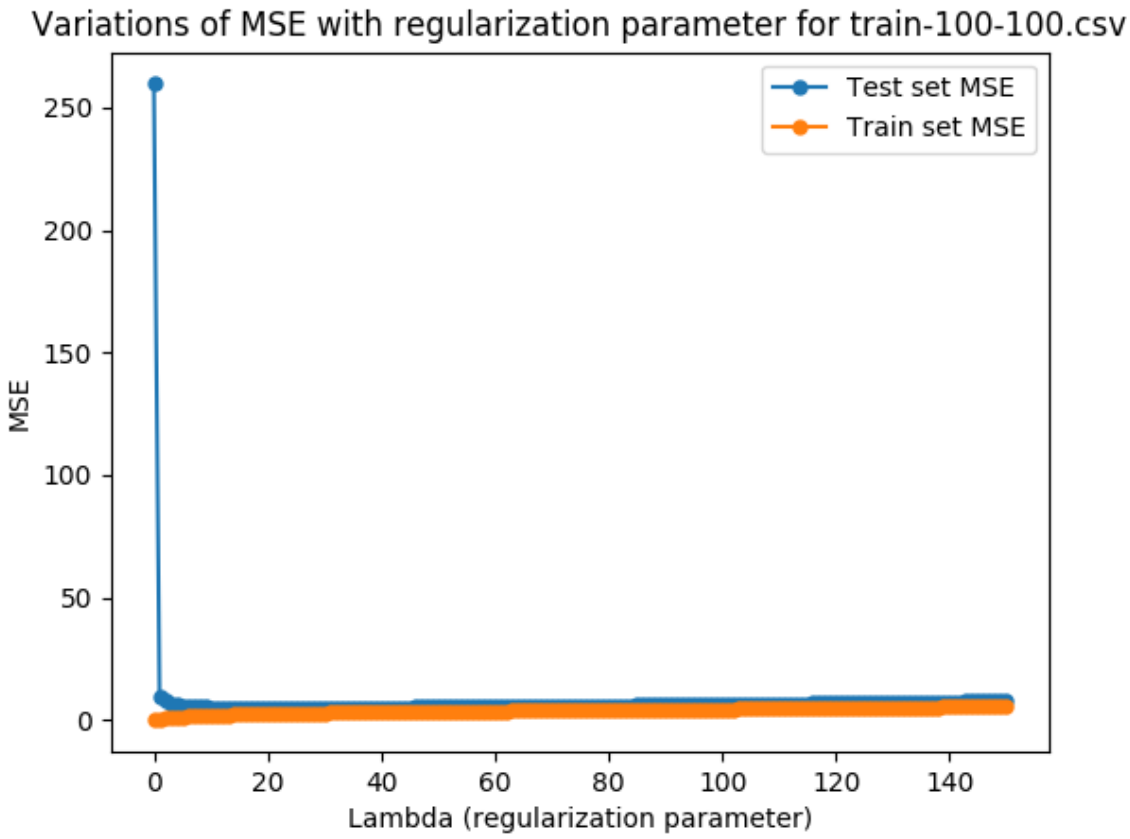


Figure 2: Dataset: train-100-100.csv

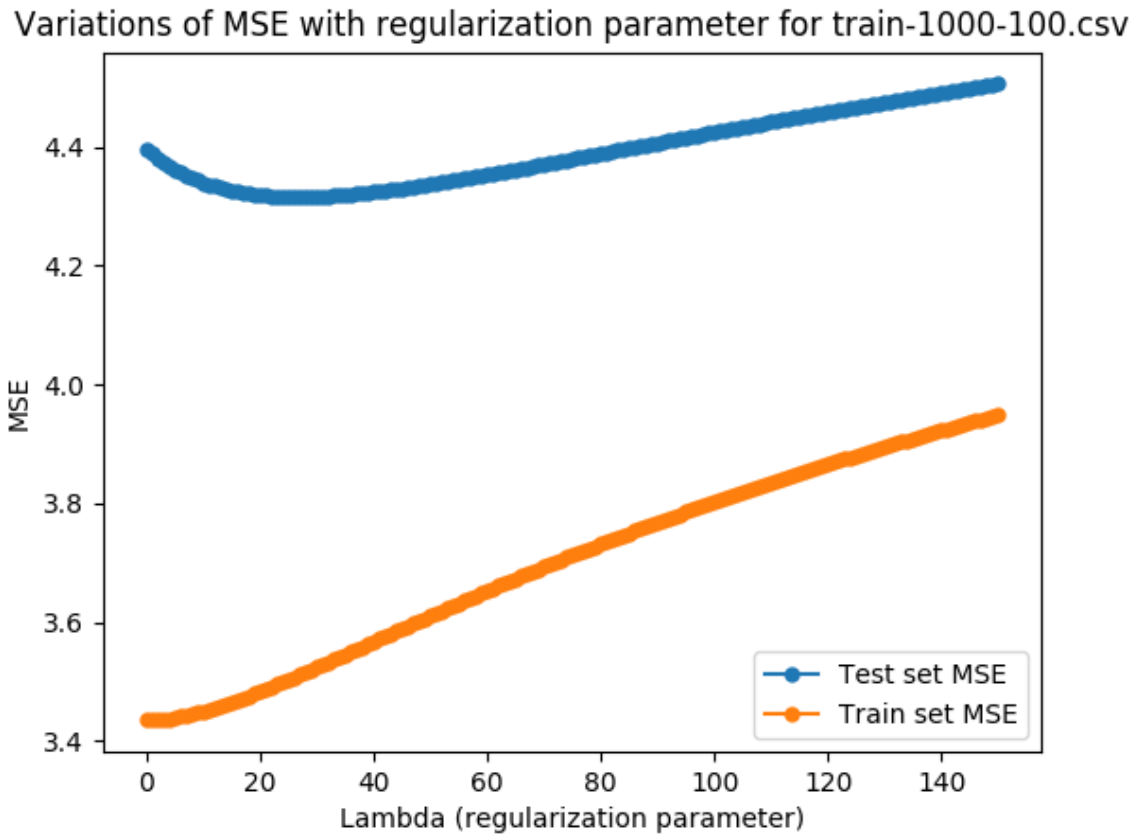


Figure 3: Dataset: train-1000-100.csv

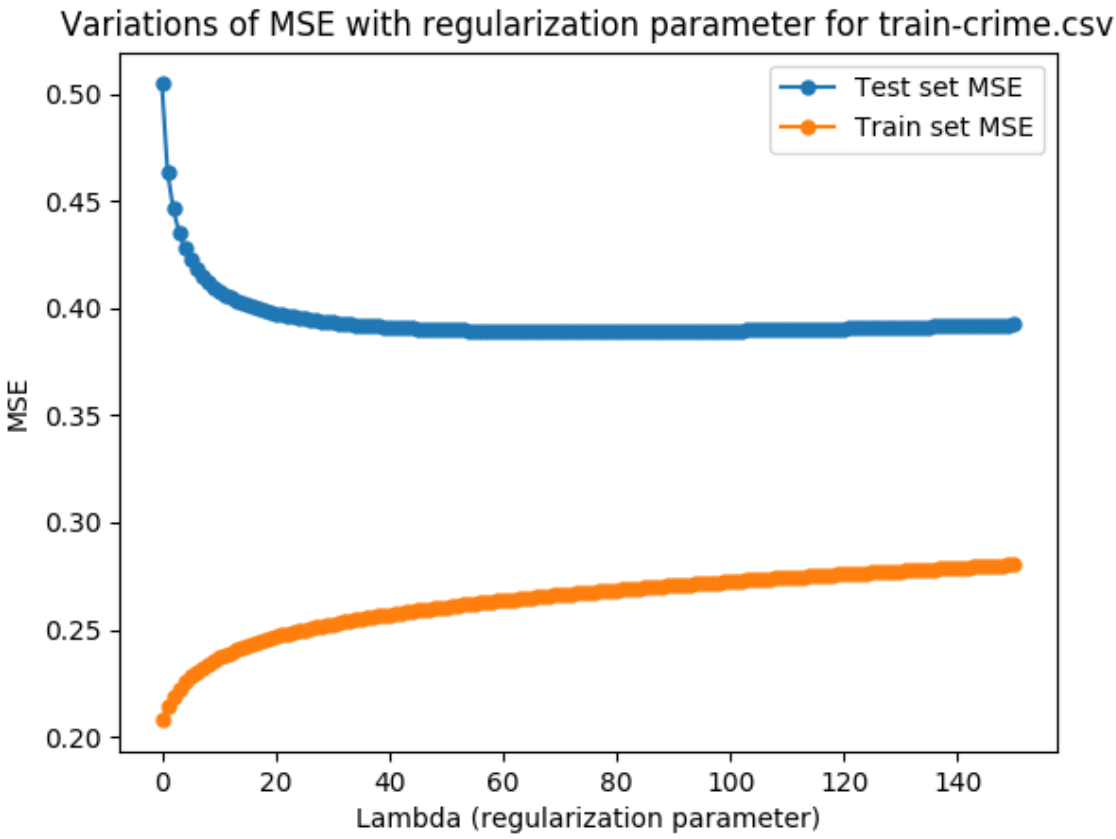


Figure 4: Dataset: train-crime.csv

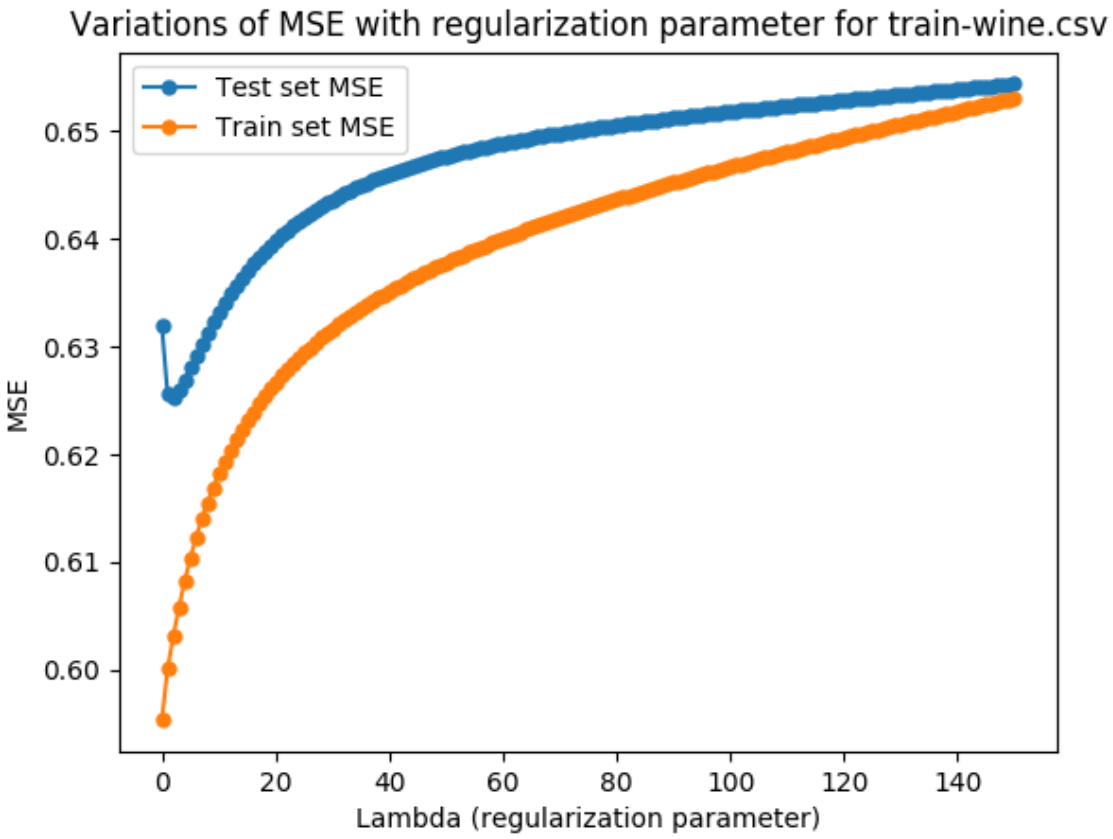


Figure 5: Dataset: train-wine.csv

Question 1: Can we use training performance to choose $\lambda$?

From the 5 figures above we can see that the mean squared error increases with high regularization parameter. The algorithm achieves lowest training MSE with $\lambda = 0$. However, with $\lambda = 0$, this is unregularized linear regression and often performs poorly in terms of test set MSE. This is because regularization is a term that 'perturbs' the matrix $\beta\Phi^T\Phi$. However, this also means that the resulting least square solution is not the same as $(\Phi^T\Phi)^{-1}\Phi^T t$. However, unregularized linear regression introduces the problem of overfitting where the mean squared error is low in the training set but high in the test set. Regularization addresses this problem.

Question 2: How does $\lambda$ affect error on the test set?

From the 5 figures we could see that there is an optimal choice of $\lambda$ for each data set. In fact, for these data sets, we can often find a 'too small', 'just right' and 'too large' values for $\lambda$. The error usually varies with $\lambda$ in a V-shape or upward parabola for this reason. Too small values of $\lambda$ doesn't do enough to correct overfitting problem. However, too large value of $\lambda$ causes the algorithm to produce an inaccurate estimate of the underlying data distribution.

Question 3: How $\lambda$ varies with number of features and examples?

We can answer this question by looking at the first 3 data sets (100-10, 100-100 and 1000-100) and output from task 1 below. We can see that increasing the number of features results in a higher optimal $\lambda$. We also see that unregularized linear regression performs very poorly in the 100-100 data set. This is due to over-fitting problem. High number of features also means that

some of those features might be irrelevant or unimportant. Fitting those features in the model therefore introduces the over-fitting problem. Higher value of $\lambda$ aims to correct this problem. Note that because of the high MSE value of unregularized regression in 100-100 datasets, this gives the impression that mse varies more with $\lambda$ in 100-10 but this is just a graphical scaling illusion.

On the other hand, between 100-100 and 1000-100, increasing the number of examples gives a higher optimal $\lambda$, however, the gap between the two $\lambda$ values is not as big as between 100-10 and 100-100. Increasing the number of examples while keeping the same number of features also raises the chances of overfitting problem. However, the performance of the optimal $\lambda$ is better. This is because we have more training examples, fitting over more training examples allows to the algorithm to learn better estimate of the data distribution.

```
optimal lambda:  8
MSE:  4.15967850948
optimal lambda:  22
MSE:  5.07829980059
optimal lambda:  27
MSE:  4.31557063032
optimal lambda:  75
MSE:  0.389023387713
optimal lambda:  2
MSE:  0.625308842305
```

# 2 Learning curves

In figure 6 below, we can see the effect of training size on the performance of linear regression (test MSE). For this task, I chose 3 different values of $\lambda$ from task 1, namely 5, 27 and 145 (too small, just right and too high respectively). First note that when the training size is 'sufficiently large' (about 300-400 samples), the different choices of $\lambda$ give similar performances on the test set. The differences are in how the different $\lambda$ affects performance when training size is low, from 10 to 200. At very small training size, the problem of overfitting is not significant, a result, 'too small' values of $\lambda$ performs better while 'too large' values cause significant deviation from the best fit estimate.

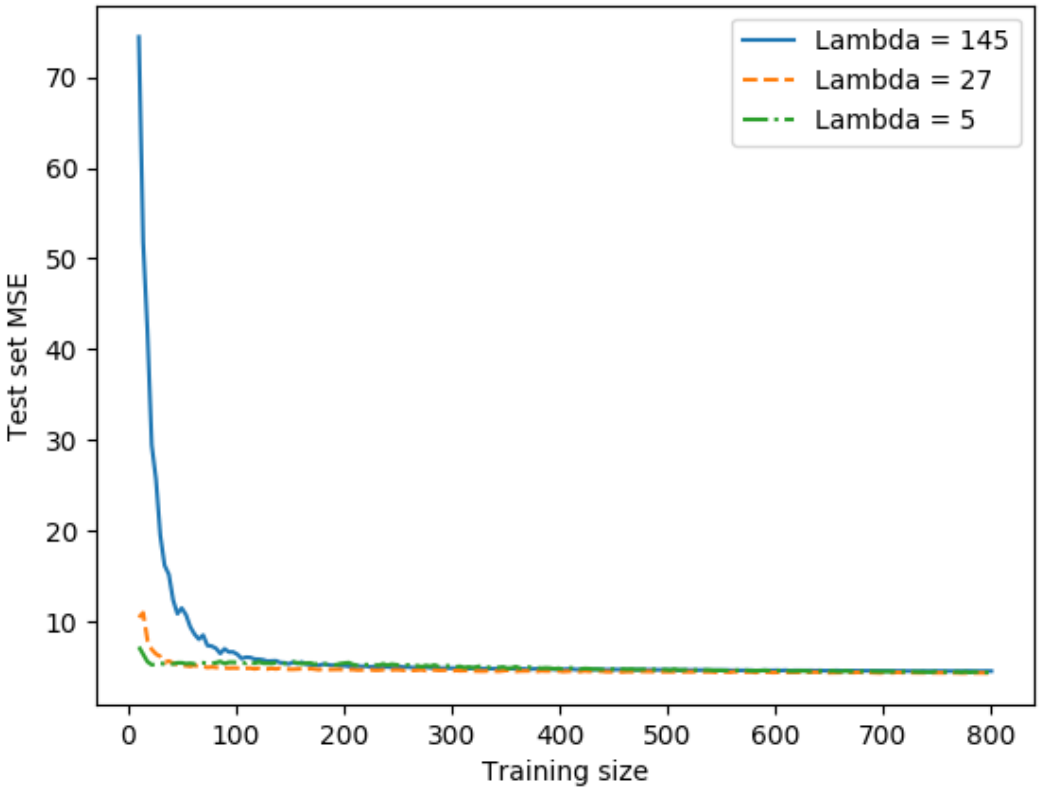


Figure 6: Dataset: train-1000-100.csv

# 3 Bayesian Model Selection

Figure 7 below shows how Bayesian Model selection performs on the test set MSE after finding the prior parameter $\alpha$ and $\beta$ that maximizes the evidence function on the training samples.

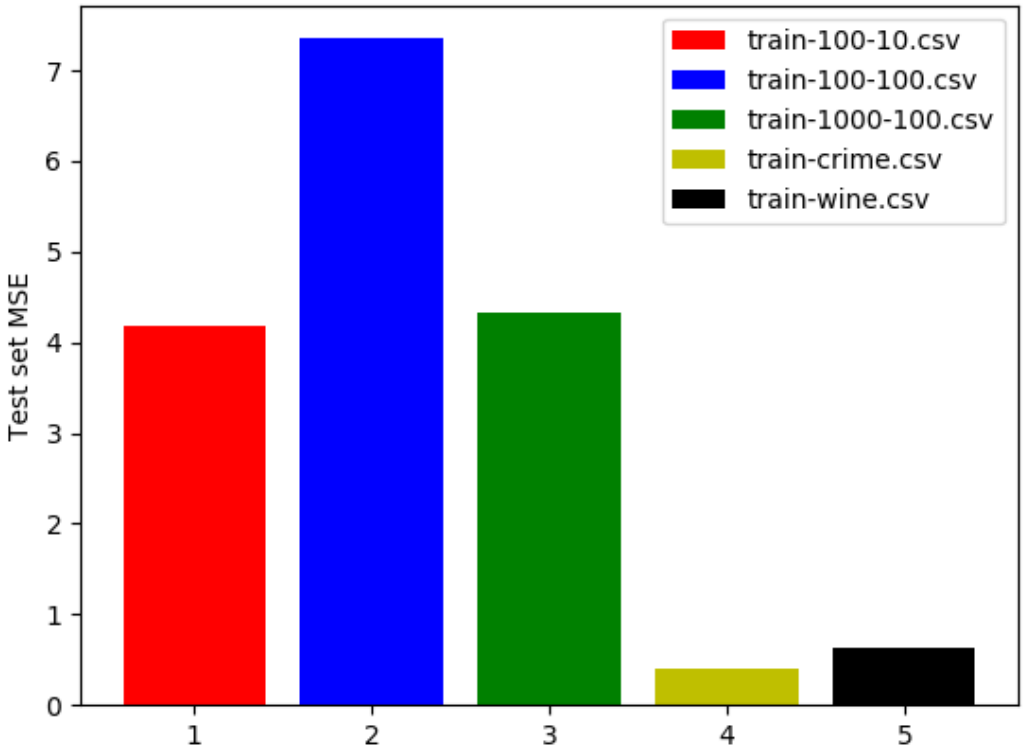


Figure 7: Performance of bayesian model selection on 5 datasets

Furthermore, below is the direct output of task 3.

```
Task 3
train-100-10.csv   :  4.18010141645
train-100-100.csv   :  7.35245622363
train-1000-100.csv   :  4.33835146361
train-crime.csv   :  0.391102307473
train-wine.csv   :  0.626746238571
```

By comparing to the result from task 1, we can see that in terms of test set MSE, Bayesian model selection performs almost as well as the optimal $\lambda$ in task 1 for all the data sets. Note that in task 1, we can choose the optimal $\lambda$ based on cross-validation while Bayesian model selection only relies on training set to choose the optimal parameter $\alpha$ and $\beta$. Hence, Bayesian model selection allows us to make full use of the training data.

Question: how does the performance depend on the number of examples and features?

We can answer this question by considering the 3 data sets 100-10, 100-100 and 1000-100. Between 100-10 and 100-100, increasing the number of features causes the performance to worsen. This makes sense because more features where some of the features might be unimportant or irrelevant also worsens the problem of over-fitting. On the other hand, between 100-100 and 1000-100, the performance improves. This is because we have more training examples, and thus the estimate produced by the algorithm better explains the data distribution.

# 4 Model selection for parameters and model order

In the following two figures, we see how $d$, the number of dimensions in the data affects the performance of Bayesian model selection and unregularized linear regression. I've also attached the direction output from task 4. Note that the figure is badly scaled because of the high MSE value for $d = 0$, it looks like there is little difference in MSE for different values of $d > 3$, however, looking at the direct output from task 4 can show that there is significant difference in test set mse between different $d$ values.

From the results, we can see that the log evidence can be a good indicator for the choice of $\alpha$ and $\beta$ in Bayesian Model selection as well as a good indication of the polynomial degree d in polynomial regression. When the log evidence is the highest, the test set MSE is also the lowest for both MAP (Bayesian model selection) and unregularized linear regression.

From the results, it seems that the optimal choice of degree is 3 as it produces the highest evidence and also lowest MSE on the test set. Obviously, when d is too low, this is underfitting because linear regression cannot captures underlying data distribution. On the other hand, if d is too high, this introduces overfitting problem. As a result, test set MSE worsens.

Furthermore, at higher dimensions $d = 8, 9, 10$, the log evidence turns to negative infinity. This is perhaps due to numerical instability of the matrix $\Phi$ at higher dimension. Furthermore, we are dealing with the eigenvalues for $\Phi^T\Phi$. This numerical instability explains why the algorithm calculates the determinant of the matrix $A = (\alpha I + \beta\Phi^T\Phi)$ to be infinity since some of the eigenvalues are highly positive. However, this might not pose an issue in our example because the log evidence correctly points out the 'optimal' choice of polynomial dimension.

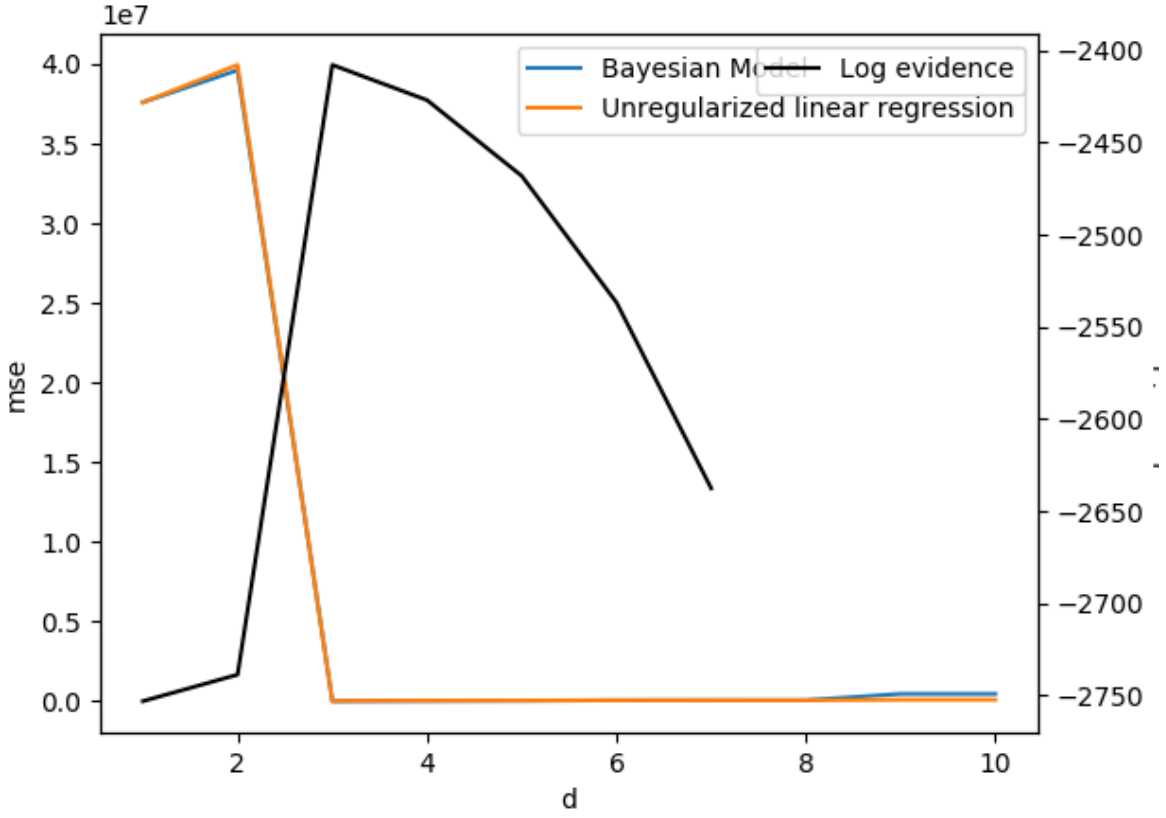


Figure 8: Data set: f3. How dimension of data affects performance and log evidence

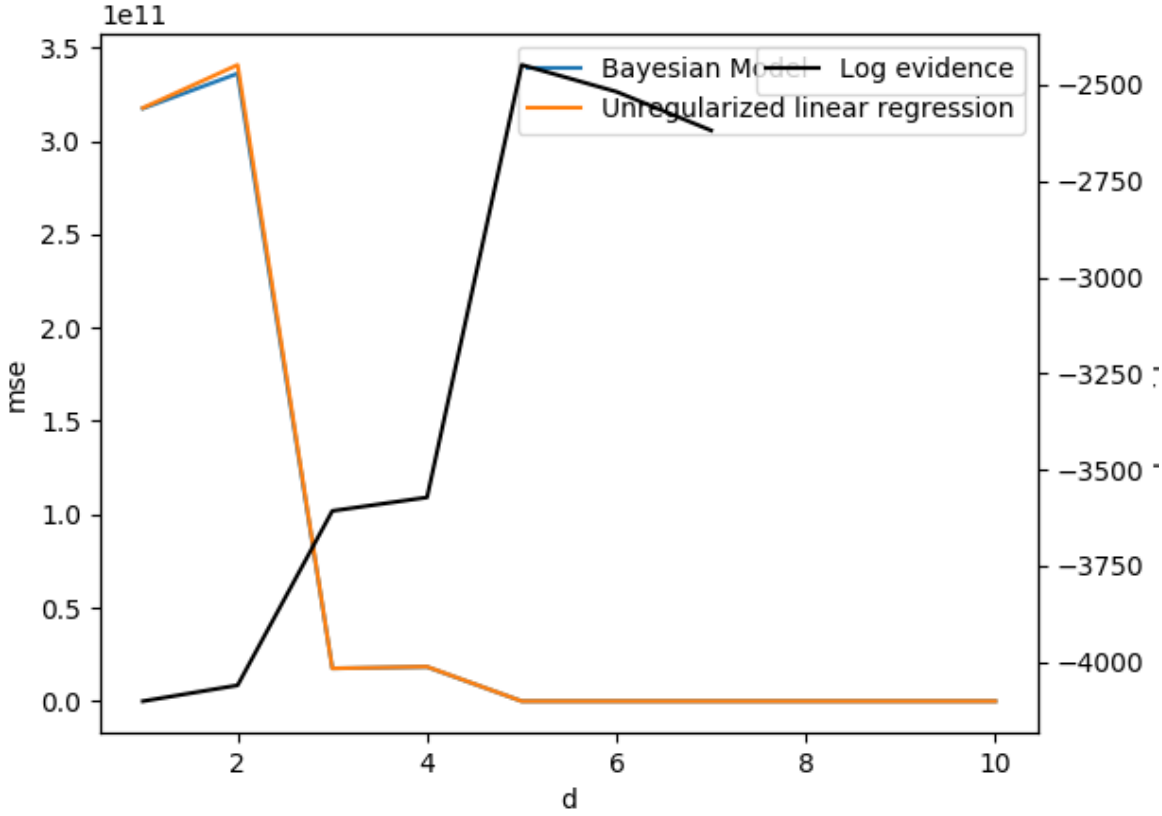


Figure 9: Data set: f5. How dimension of data affects performance and log evidence

```
Output from task 4

Data:  /home/duc/Documents/homework/COMP136/PP2/data/train-f3.csv
dimension:  1
mse unregularized linear regression:  37588010.7113
mse model selection:  37599015.3449
log evidence:  -2753.08342955
dimension:  2
mse unregularized linear regression:  39945001.3538
mse model selection:  39610963.6647
log evidence:  -2738.69164755
dimension:  3
mse unregularized linear regression:  26408.8941494
mse model selection:  4899.15907643
log evidence:  -2407.96039634
dimension:  4
mse unregularized linear regression:  49126.6999109
mse model selection:  19526.3179527
log evidence:  -2427.06885081
dimension:  5
mse unregularized linear regression:  51392.0042222
```

```
mse model selection:  33603.4963643
log evidence:  -2468.18612376
dimension:  6
mse unregularized linear regression:  61993.4489737
mse model selection:  62540.4791028
log evidence:  -2536.78272088
dimension:  7
mse unregularized linear regression:  65763.5728629
mse model selection:  67885.8683404
log evidence:  -2637.64044354
dimension:  8
mse unregularized linear regression:  66081.530431
mse model selection:  66308.0147203
log evidence:  -inf
dimension:  9
mse unregularized linear regression:  94291.0767605
mse model selection:  463398.771823
log evidence:  -inf
dimension:  10
mse unregularized linear regression:  97098.9771536
mse model selection:  461238.499968
log evidence:  -inf


Data:  /home/duc/Documents/homework/COMP136/PP2/data/train-f5.csv
dimension:  1
mse unregularized linear regression:  317641564451.0
mse model selection:  317348996658.0
log evidence:  -4101.46165667
dimension:  2
mse unregularized linear regression:  340688646775.0
mse model selection:  336061630847.0
log evidence:  -4060.21895646
dimension:  3
mse unregularized linear regression:  17579791671.8
mse model selection:  17575052256.5
log evidence:  -3607.0514682
dimension:  4
mse unregularized linear regression:  18461024800.3
mse model selection:  18334476951.6
log evidence:  -3572.00380929
dimension:  5
mse unregularized linear regression:  32501.7561196
mse model selection:  27748.5178424
log evidence:  -2448.27730183
dimension:  6
mse unregularized linear regression:  37814.8934114
```

```
mse model selection:  33009.6596951
log evidence:  -2518.00489351
dimension:  7
mse unregularized linear regression:  34901.827664
mse model selection:  30718.9554499
log evidence:  -2618.68826279
dimension:  8
mse unregularized linear regression:  66599.4256865
mse model selection:  33322.1135668
log evidence:  -inf
dimension:  9
mse unregularized linear regression:  91250.1566689
mse model selection:  49213.1313672
log evidence:  -inf
dimension:  10
mse unregularized linear regression:  132543.054734
mse model selection:  50114.5830345
log evidence:  -inf
```