# Problem 1

## Question A

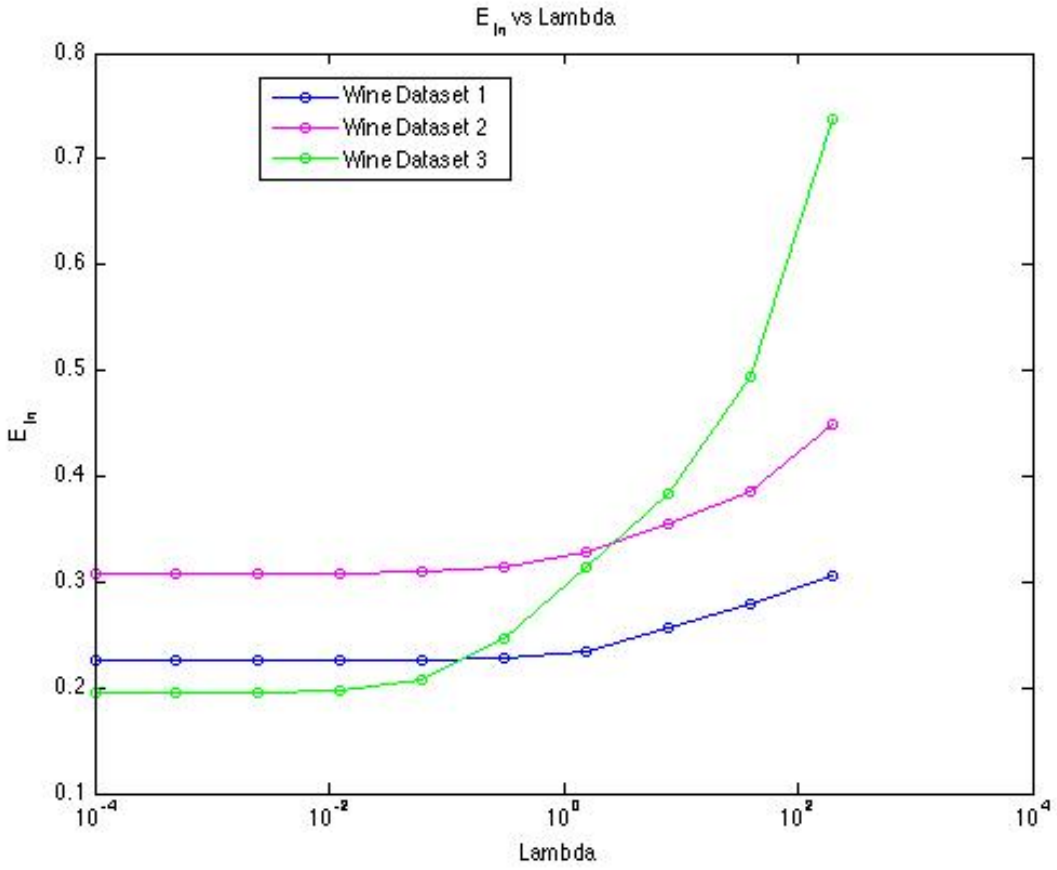
Adding the penalty term cannot decrease the training (in-sample) error. This is because adding the penalty term is equivalent to constraining the model complexity. And constraining the model complexity must either harm the in-sample error or keep it the same; we cannot bring the in sample error *down* by constraining which models we use.

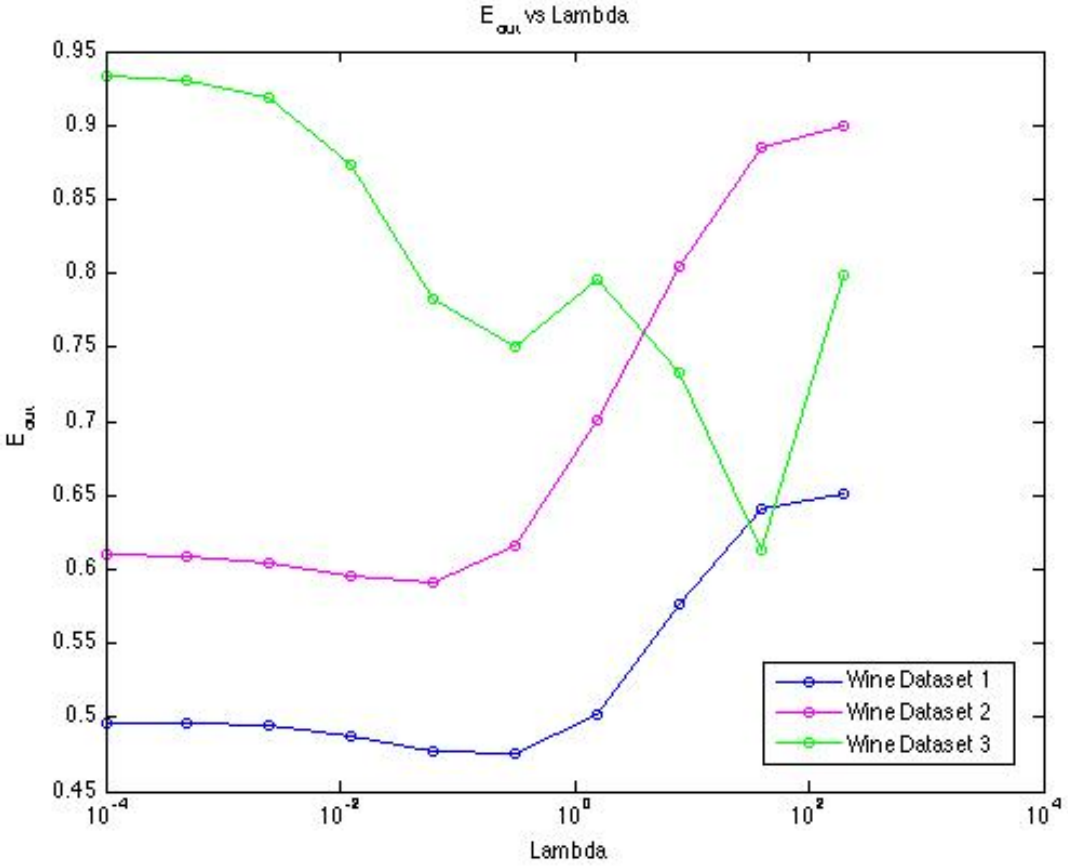
Adding the penalty term does not always decrease the out-of-sample errors. We can find that it does not decrease the out-of-sample errors in cases when adding the penalty term causes underfitting; that is, when it causes a simpler model to be used to model a more complex one. Basically, adding the penalty term can be harmful or unhelpful when a more complex model generalizes better than a simpler model.

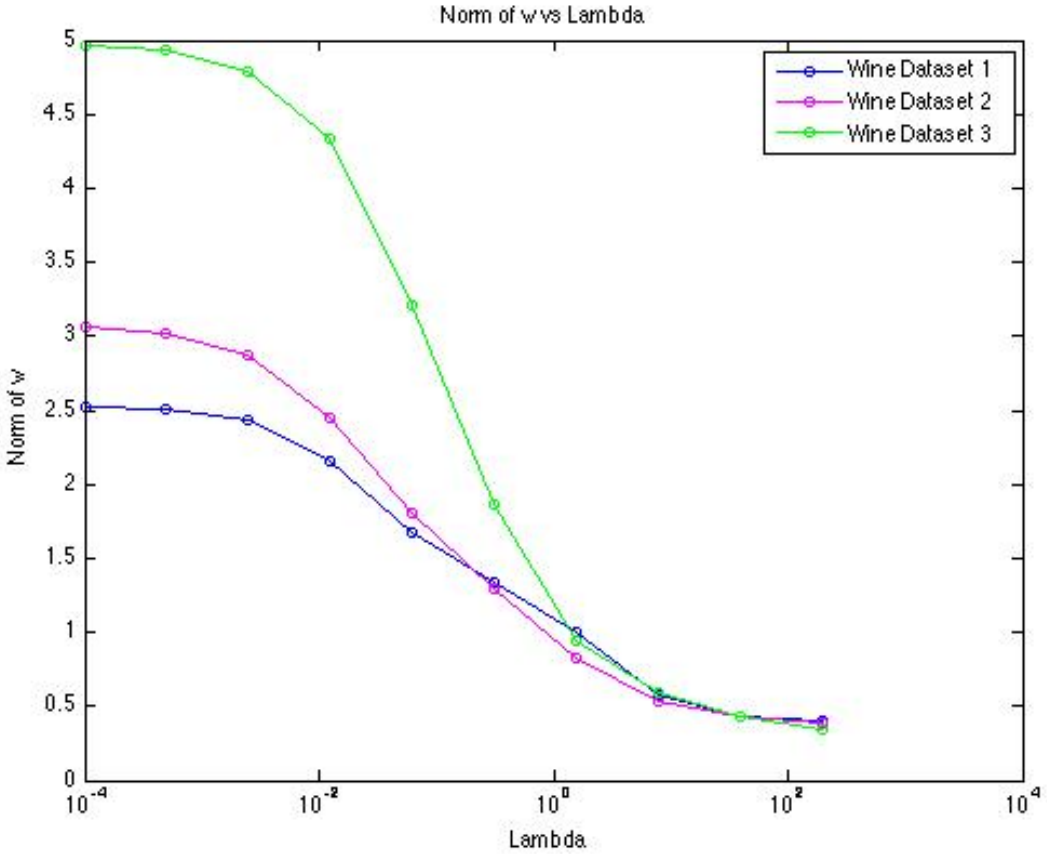
## Question B

We rarely use $\ell_0$ regularization because it involves the $\ell_0$ norm of $\mathbf{w}$. And the $\ell_0$ norm of $\mathbf{w}$ is not continuous (by its definition), which means we cannot perform gradient descent (because we can't find the gradient or sub gradient of the regularization penalty term).

## Question C




E_out vs Lambda
E_out
Lambda
Wine Dataset 1
Wine Dataset 2
Wine Dataset 3


Norm of w vs Lambda
Norm of w
Lambda
Wine Dataset 1
Wine Dataset 2
Wine Dataset 3

## Question D

**training** For lower lambdas, training with Wine Dataset 3 gives us slightly lower $E_{in}$, but for higher lambdas, training with Wine Dataset 1 gives us much lower $E_{in}$ as training with Wine Dataset 3 gets much worse. This means more complex models are better at fitting Dataset 3 and simpler models are worse at fitting Dataset 3. The former can be explained by the following reason. Since Dataset 3 is a smaller subset of Dataset 1, it is easier to more closely fit all the points with a more complex model. This makes sense; it is easier to fit less points than more points with more complex models. The latter can be explained by the following reason. Since Dataset 3 is a smaller subset of Dataset 1, when the models get simpler and fit the data less closely, points with large error negatively affect the overall $E_{in}$ much more for Dataset 3 than Dataset 1 (again, because Dataset 3 is a smaller subset of Dataset 1).

**validation** For basically all lambdas, training with Wine Dataset 1 gives us lower $E_{out}$, except for at $\lambda = 39.0625$, where training with Wine Dataset 3 gives us a slightly lower $E_{out}$. We can explain this trend with the fact that variance should be lower when training with Wine Dataset 1 and with equivalent lambdas, because at equivalent lambdas model complexity is the same, but Wine Dataset 1 has more data points which reduces variance. Also, bias should be about the same whether training with Dataset 1 or 3 (keeping lambda constant), because bias only depends on model complexity and is independent of training data size. Thus, since the biases are around the same and the variance is lower when training with Dataset 1, by the Bias-Variance Decomposition, $E_{out}$ should be lower when training with Dataset 1.

## Question E

Around the midway point and going left, we start to see overfitting. We see this as $E_{out}$ rises from its local minimum as lambda decreases (going left). This is overfitting because we are increasing model complexity at the cost of hurting generalization ($\lambda$ is decreasing, $E_{out}$ is increasing). Around the midway point and going right, we also start to see underfitting. We see this as $E_{out}$ rises from its local minimum as lambda increases (going right). This is underfitting because in this area, we are decreasing model complexity and using models that are too constrained/simple and thus cannot capture the data accurately enough, which is reflected in the rising $E_{out}$ and $E_{in}$.

## Question F

We can see that the norm of $\mathbf{w}$ decreases as $\lambda$ increases when we train with Wine Dataset 1. This is because setting $\lambda$ to some value effectively constrains the norm of $\mathbf{w}$ to be less than or equal to some constant $c$. And as $\lambda$ increases, the $c$ we are constraining the norm of $\mathbf{w}$ to be less than or equal to decreases. This intuitively makes sense, as we are trying to minimize $E$, which contains the term $\frac{\lambda}{N}\mathbf{w}^\intercal\mathbf{w}$, so as $\lambda$ increases bigger $\mathbf{w}$s get more and more harmful. So basically, the norm of $\mathbf{w}$ decreases as $\lambda$ increases because a bigger $\lambda$ corresponds to a stricter constraint on the norm of $\mathbf{w}$ (a smaller $c$ in $\mathbf{w}^\intercal\mathbf{w} \leq c$).

## Question G

I would choose $\lambda = 39.0625$ to train my final model, because it has the lowest $E_{out}$ and thus has the best chance of generalizing out of sample.

# Problem 2

## Question A

### (1)

Using Bayes rule and the given prior, we have that

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w}|\lambda)}{p(\mathcal{D})}$$

We start with

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, p(\mathbf{w}|\mathcal{D})$$

We can now plug in the numerator of the first equation we wrote, excluding the denominator as it does not have $\mathbf{w}$ in it and we are maximizing over $\mathbf{w}$.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, p(\mathcal{D}|\mathbf{w})p(\mathbf{w}|\lambda)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, p(\mathcal{D}|\mathbf{w}) \prod_{j=1}^{D} \frac{\lambda}{2} e^{-\lambda|w_j|}$$

We can now do the following:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, \log\left(p(\mathcal{D}|\mathbf{w}) \prod_{j=1}^{D} \frac{\lambda}{2} e^{-\lambda|w_j|}\right)$$

because the quantity inside is nonnegative and log is monotonically increasing so if you maximize it you maximize its argument. Then clearly we can just minimize the negative to get the following.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\log\left(p(\mathcal{D}|\mathbf{w}) \prod_{j=1}^{D} \frac{\lambda}{2} e^{-\lambda|w_j|}\right)$$

Now we can just apply log rules to get the following.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \log \prod_{j=1}^{D} \frac{\lambda}{2} e^{-\lambda|w_j|}\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} \log \frac{\lambda}{2} e^{-\lambda|w_j|}\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} \log \frac{\lambda}{2} + \log e^{-\lambda|w_j|}\right)$$

We can get rid of all the $\log \frac{\lambda}{2}$ terms, as these are constant with respect to $\mathbf{w}$. This leaves us with

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} \log e^{-\lambda|w_j|}\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} -\lambda|w_j|\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\log p(\mathcal{D}|\mathbf{w}) + \lambda||\mathbf{w}||_1$$

**(2)**

Using Bayes rule and the given prior, we have that

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w}|\lambda)}{p(\mathcal{D})}$$

We start with

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, p(\mathbf{w}|\mathcal{D})$$

We can now plug in the numerator of the first equation we wrote, excluding the denominator as it does not have $\mathbf{w}$ in it and we are maximizing over $\mathbf{w}$.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, p(\mathcal{D}|\mathbf{w})p(\mathbf{w}|\lambda)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, p(\mathcal{D}|\mathbf{w}) \prod_{j=1}^{D} \sqrt{\frac{\lambda}{\pi}} e^{-\lambda w_j^2}$$

We can now do the following:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, \log\left(p(\mathcal{D}|\mathbf{w}) \prod_{j=1}^{D} \sqrt{\frac{\lambda}{\pi}} e^{-\lambda w_j^2}\right)$$

because the quantity inside is nonnegative and log is monotonically increasing so if you maximize it you maximize its argument. Then clearly we can just minimize the negative to get the following.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\log\left(p(\mathcal{D}|\mathbf{w}) \prod_{j=1}^{D} \sqrt{\frac{\lambda}{\pi}} e^{-\lambda w_j^2}\right)$$

Now we can just apply log rules to get the following.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \log \prod_{j=1}^{D} \sqrt{\frac{\lambda}{\pi}} e^{-\lambda w_j^2}\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} \log \sqrt{\frac{\lambda}{\pi}} e^{-\lambda w_j^2}\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} \log \sqrt{\frac{\lambda}{\pi}} + \log e^{-\lambda w_j^2}\right)$$

We can get rid of all the $\log \sqrt{\frac{\lambda}{\pi}}$ terms, as these are constant with respect to $\mathbf{w}$. This leaves us with

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} \log e^{-\lambda w_j^2}\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\left(\log p(\mathcal{D}|\mathbf{w}) + \sum_{j=1}^{D} -\lambda w_j^2\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, -\log p(\mathcal{D}|\mathbf{w}) + \lambda ||\mathbf{w}||_2^2$$

**(3)**

We have that

$$\mathbf{y} \sim \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}),\ \mathcal{D} \text{ contains } \mathbf{X} \text{ and } \mathbf{y}$$

We start with

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, p(\mathcal{D}|\mathbf{w})$$

The multivariate normal distribution has a density given by the following when the second parameter ($\sigma^2\mathbf{I}$ in our case) is positive definite:

$$\frac{1}{\sqrt{(2\pi)^n|\sigma^2\mathbf{I}|}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\intercal(\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\mathbf{X}\mathbf{w})}$$

where $n$ is the number of training points. Since $\sigma^2\mathbf{I}$ is positive definite, we can use this density function. This density function describes the relative likelihood for generating the dataset $\mathcal{D}$ given a certain weight vector $\mathbf{w}$. In other words, it is equivalent to $p(\mathcal{D}|\mathbf{w})$. So we can plug it into the above equation to get the following:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}} \frac{1}{\sqrt{(2\pi)^n|\sigma^2\mathbf{I}|}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\intercal(\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\mathbf{X}\mathbf{w})}$$

We can drop constants (terms that do not involve $\mathbf{w}$) to get the following:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, e^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\intercal(\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\mathbf{X}\mathbf{w})}$$

We can now do the following:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, \log\left(e^{-\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\intercal(\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\mathbf{X}\mathbf{w})}\right)$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}}\, -\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\intercal(\sigma^2\mathbf{I})^{-1}(\mathbf{y}-\mathbf{X}\mathbf{w})$$

because the quantity inside is nonnegative and log is monotonically increasing so if you maximize it you maximize its argument. Then clearly we can just minimize the negative (and get rid of constants) to get the following.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, (\mathbf{y}-\mathbf{X}\mathbf{w})^\intercal(\mathbf{y}-\mathbf{X}\mathbf{w})$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}}\, ||\mathbf{y}-\mathbf{X}\mathbf{w}||_2^2$$

## Question B

Figure 1: A plot of the each of the weights as a function of $\lambda$, where $\mathbf{w}$ was estimated using linear regression with Lasso regularization

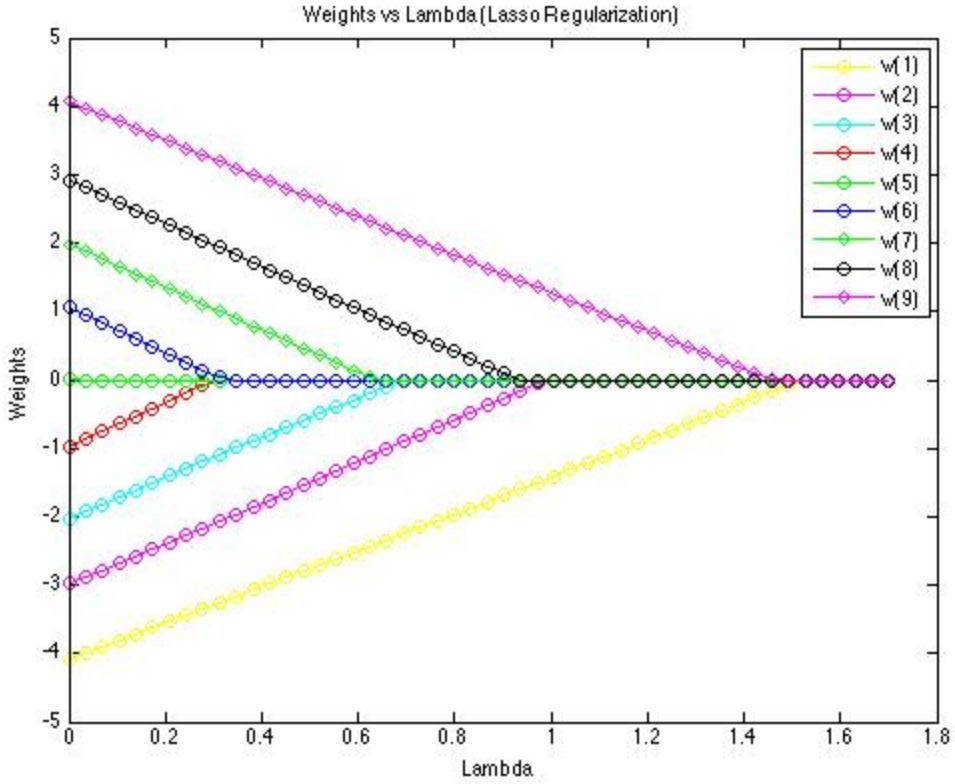
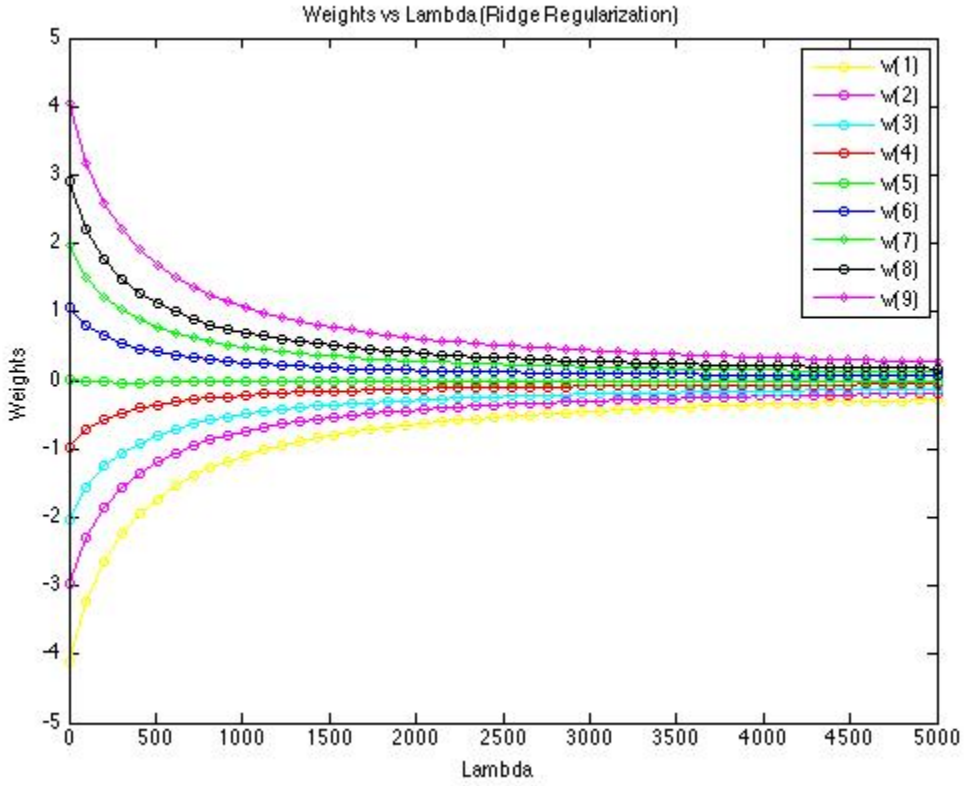
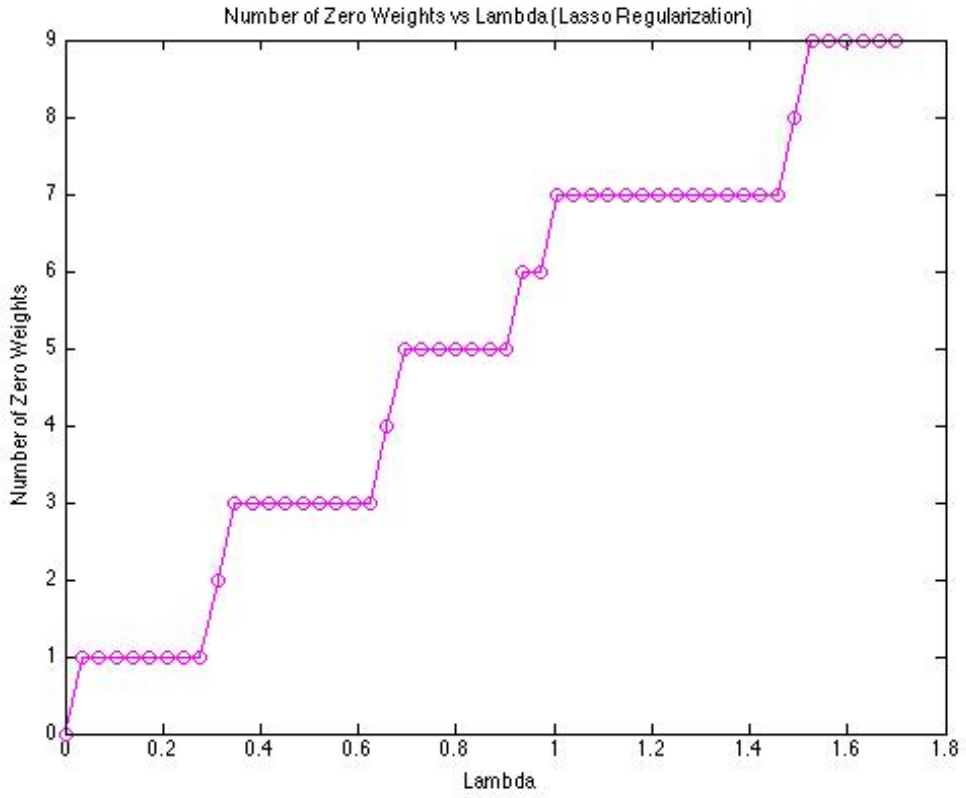



Figure 2: A plot of the each of the weights as a function of $\lambda$, where $\mathbf{w}$ was estimated using linear regression with Ridge regularization



**(3)**

As the regularization parameter varies with Lasso regression, all the estimated weights go to zero (before $\lambda = 1.8$). As the regularization parameter varies with Ridge regression, none of the estimated weights go to zero (at least up to $\lambda = 50000$). We can see the former behavior in the plot below:




## Question C

Note: $\mathbf{X}$ is a matrix where each row is a single training point.

**(1)**

We want to minimize over all the points, so we will minimize the following:

$$f(\mathbf{w}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \lambda||\mathbf{w}||_1$$

To minimize this, we can find the gradient

$$\nabla f(\mathbf{w}) = -2\mathbf{X}^\intercal(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{\lambda \mathbf{w}}{|\mathbf{w}|}$$

$$\nabla f(\mathbf{w}) = -2\mathbf{X}^\intercal\mathbf{y} + 2\mathbf{X}^\intercal\mathbf{X}\mathbf{w} \pm \lambda I$$

and set the gradient to 0 (because $f$ is convex)

$$-2\mathbf{X}^\intercal\mathbf{y} + 2\mathbf{X}^\intercal\mathbf{X}\mathbf{w} \pm \lambda I = 0$$

$$2\mathbf{X}^\intercal\mathbf{X}\mathbf{w} = 2\mathbf{X}^\intercal\mathbf{y} \mp \lambda I$$

$$\mathbf{w} = (2\mathbf{X}^\intercal\mathbf{X})^{-1}(2\mathbf{X}^\intercal\mathbf{y} \mp \lambda I)$$

**(2)**

We want to see if there exists a value for $\lambda$ such that $\mathbf{w} = 0$, and find the smallest such value. From the above, we have that

$$\mathbf{w} = (2\mathbf{X}^\intercal\mathbf{X})^{-1}(2\mathbf{X}^\intercal\mathbf{y} \mp \lambda I)$$

We can see that we need the second term (involving $\lambda$ to equal 0. So we have that

$$0 = (2\mathbf{X}^\intercal\mathbf{y} \mp \lambda I)$$

Now we get two cases.

$$\mathbf{w} > 0 \implies 0 = 2\mathbf{X}^\intercal\mathbf{y} - \lambda I \implies \lambda = 2\mathbf{X}^\intercal\mathbf{y}$$

$$\mathbf{w} < 0 \implies 0 = 2\mathbf{X}^\intercal\mathbf{y} + \lambda I \implies \lambda = -2\mathbf{X}^\intercal\mathbf{y}$$

Notice that when $\mathbf{w} < 0$, $\mathbf{X}^\intercal\mathbf{y} < 0$ as well. We can see this with the following reasoning.

$$\mathbf{w} = (2\mathbf{X}^\intercal\mathbf{X})^{-1}(2\mathbf{X}^\intercal\mathbf{y} + \lambda I) < 0$$

$$(2\mathbf{X}^\intercal\mathbf{y} + \lambda I) < 0 \implies \mathbf{X}^\intercal\mathbf{y} < 0$$

Therefore, the two cases above collapse into the one case below:

$$\lambda = ||2\mathbf{X}^\intercal\mathbf{y}||$$

This is the smallest such value.

**(3)**

We want to minimize over all the points, so we will minimize the following:

$$f(\mathbf{w}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \lambda||\mathbf{w}||_1$$

$$f(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||^2 + \lambda||\mathbf{w}||_1$$

To minimize this, we can find the gradient

$$\nabla f(\mathbf{w}) = 2\mathbf{X}^\intercal(\mathbf{X}\mathbf{w} - \mathbf{y}) + 2\lambda\mathbf{w}$$

$$\nabla f(\mathbf{w}) = 2\mathbf{X}^\intercal\mathbf{X}\mathbf{w} - 2\mathbf{X}^\intercal\mathbf{y} + 2\lambda\mathbf{w}$$

and set the gradient to 0 (because $f$ is convex)

$$2\mathbf{X}^\intercal\mathbf{X}\mathbf{w} - 2\mathbf{X}^\intercal\mathbf{y} + 2\lambda\mathbf{w} = 0$$

$$\mathbf{X}^\intercal\mathbf{X}\mathbf{w} - \mathbf{X}^\intercal\mathbf{y} + \lambda\mathbf{w} = 0$$

$$(\mathbf{X}^\intercal\mathbf{X} + \lambda I)\mathbf{w} = \mathbf{X}^\intercal\mathbf{y}$$

$$\mathbf{w} = (\mathbf{X}^\intercal\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\intercal\mathbf{y}$$

**(4)**

We want to see if there exists a value for $\lambda$ such that $\mathbf{w} = 0$. So basically, since

$$\mathbf{w} = (\mathbf{X}^\intercal\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\intercal\mathbf{y}$$

we want

$$(\mathbf{X}^\intercal\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\intercal\mathbf{y} = 0$$

$$(\mathbf{X}^\intercal\mathbf{X} + \lambda I)^{-1} = 0$$

But for this equality to hold, the inverse of some matrix would have to equal 0. But this is not possible (because $\mathbf{CD} = \mathbf{DC} = I \neq 0$). So there does not exist a value for $\lambda$ such that $\mathbf{w} = 0$.