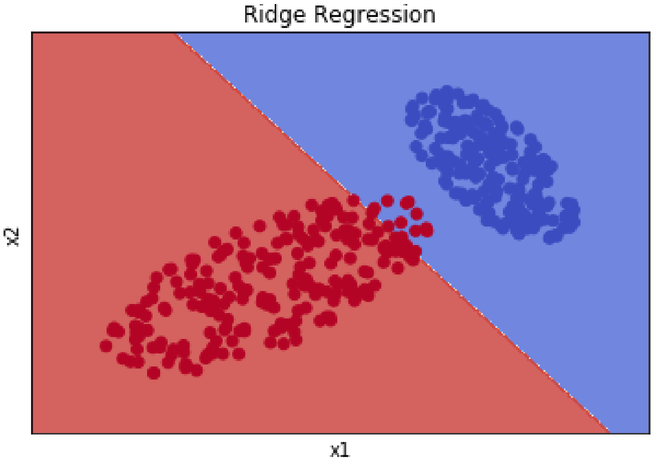
Jessica Wang
Machine Learning & Data Mining
Set 2

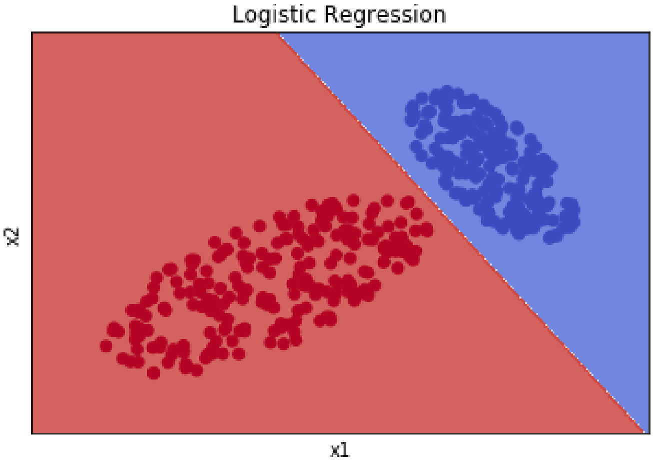
I. Comparing Different Loss Functions

Problem A.

Squared loss is often a terrible choice of loss function to train on for classification problems because points in the data set that lie farther away from the decision boundary are heavily penalized since the squared loss tries to move the distance boundary so that the distance between points is minimalized which is not ideal or necessary for classification whereas classification just cares about the label of the data point.

Problem B.


Ridge Regression


Logistic Regression

The ridge regression model is unable to successfully separate and correctly classify all the points because it uses squared loss which is not optimal for reasons listed in the previous part. On the other hand, the logistic regression model can successfully separate and correctly classify all the points when C, the inverse of the regularization strength, is very large. The decision boundary is better for logistic regression than ridge regression.

Problem C.

$$\nabla_w L_{hinge} = \begin{cases} 0, & yw^T x > 1 \\ -yx, & yw^T x \leq 1 \end{cases}$$

$$\nabla_w L_{log} = -(1 - \frac{1}{1 + e^{-yw^T x}})xy$$

	$\nabla_w L_{hinge}$	$\nabla_w L_{log}$
$({}^{1}/{}_{2}, 3)$	[-1, -0.5, -3]	[-0.3775, -0.1888, -1.1326]
$(2, -2)$	[0, 0, 0]	[-0.1192, -0.2384, 0.2384]
$(-3, 1)$	[0, 0, 0]	[0.0474, -0.1423, 0.0474]

Problem D.

The gradient of the hinge loss will converge to zero when all the points are correctly classified while the gradient of the log loss will approach but never converge to zero. This is because by the gradient of the hinge loss, when all the points are correctly classified and $yw^T x \geq 1$, the gradient of the hinge loss is zero. On the other hand, the gradient of the log loss can never reach zero as this would require $e^{-yw^T x}$ to equal zero. For a linearly separable dataset, there is no way to reduce or altogether eliminate training error without changing the decision boundary. Although, for hinge loss, multiplying the weights by a scalar could affect the training error without changing the decision boundary.

Problem E.

For an SVM to be a "maximum margin" classifier, its learning objective must be to minimize $L_{hinge} + \lambda\left|| w |\right|^2$ and not just L_{hinge} since the margin is related to $\left|| w |\right|$ and as given by the lecture slides, maximizing the margins requires $\arg\underset{w,b}{min} \frac{1}{2} w^T w = \frac{1}{2}\left|| w |\right|^2$. Otherwise, the L_{hinge} only tries to correctly classify the data points.

II. Effects of Regularization

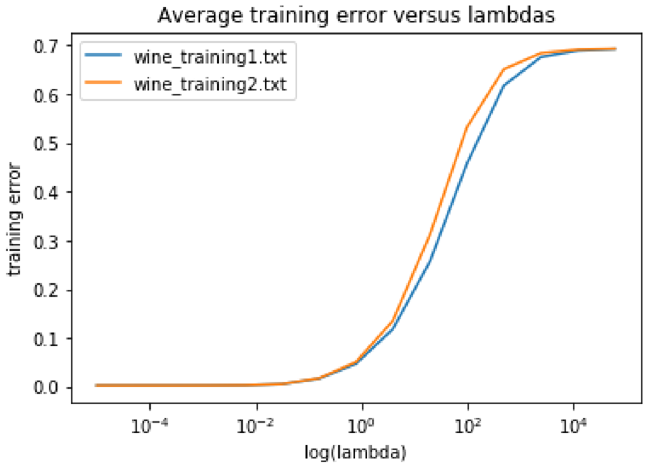
Problem A.

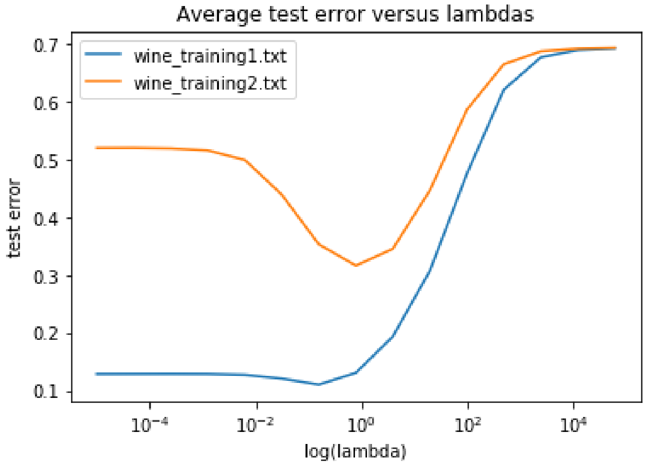
Adding a regularization penalty term will increase the training (in-sample) error since the decision boundary is not allowed to move around as much by constraining how much the weights are allowed to change and thereby reducing overfitting. Adding a penalty term can, but may not always, decreases the out-of-sample error because it regularizes the model and help combat overfitting.

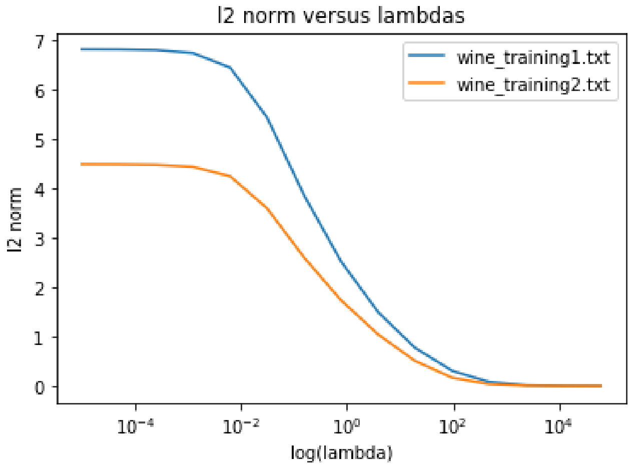
Problem B.

Since the L_0 norm simply counts the nonzero components of a feature vector, it is more difficult to optimize since it is not differentiable. On the other hand, L_1 regularization is convex and continuous which allows for better optimization using stochastic gradient descent.

Problem C.


Average training error versus lambdas
wine_training1.txt
wine_training2.txt
training error
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}
log(lambda)


Average test error versus lambdas
wine_training1.txt
wine_training2.txt
test error
0.1
0.2
0.3
0.4
0.5
0.6
0.7
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}
log(lambda)


l2 norm versus lambdas
wine_training1.txt
wine_training2.txt
l2 norm
0
1
2
3
4
5
6
7
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}
log(lambda)

Problem D.

The training errors from training with wine_training1.txt and wine_training2.txt are similar and follow a similar trajectory as described by the plots. The training error from training with wine_training1.txt is slightly smaller than that from wine_training2 because there are more data points to train on using wine_training1.txt. However, the testing error from wine_training2.txt is consistently higher than that of wine_training1.txt and this is likely due to the fact that wine_training2.txt is a subset of the data in wine_training1.txt so there is less data and the model overfitted.

Problem E.

With very small values of lambda, the regularization term does not prevent over-fitting and with large values of lambda, the regularization term causes under-fitting.

Problem F.

The l_2 norm of w decreases as lambda increases because more emphasis is placed on the regularization term.

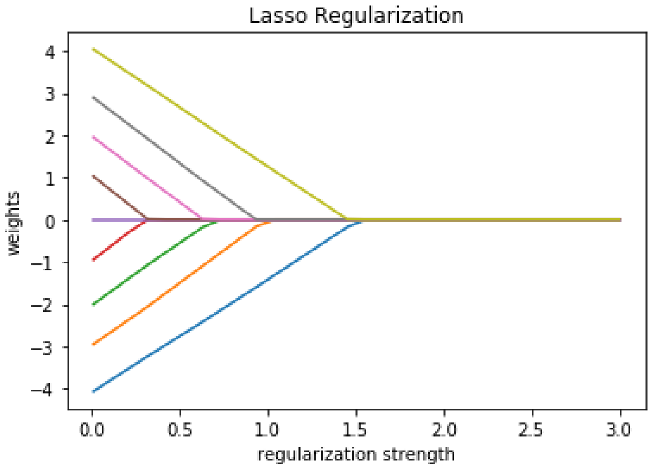
Problem G.

If the model was trained with wine_training2.txt, I would choose the lambda with the value of 0.78125 because it results in the lowest average testing error.

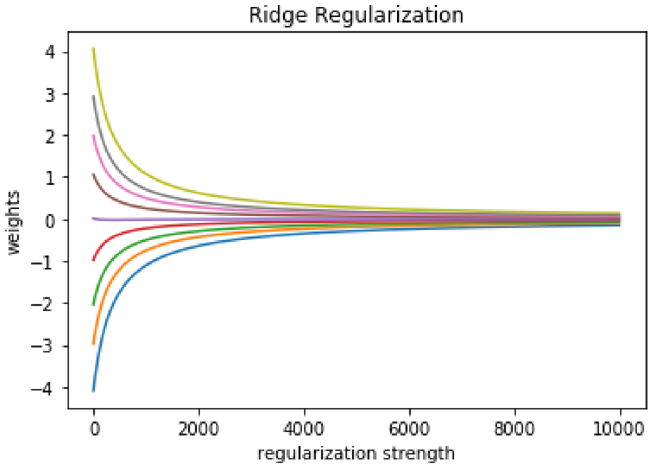
III. Lasso vs. Ridge Regularization

Problem A.

i.



ii.



iii. As the regularization parameter increases, the number of model weights that are exactly zero increases for Lasso regression. With Ridge regression, the model weights can approach but never reach zero so the number of model weights that are zero are not affected by the regularization strength as the regularization parameter increases.

Problem B.

i. We observe that since w is a scalar, $||\, w\, ||$ = | w |. As such, we need to consider three cases: (1) w < 0 (2) w > 0, and (3) w = 0.

<u>(1) For w < 0:</u>

$$\underset{w}{\arg\min}\ \|\,\boldsymbol{y}-\boldsymbol{x}w\,\|^2 + \lambda\|\,w\,\|_1$$

$$(y-xw)(y-xw)^T - \lambda w = 0$$

$$-2x^T(y-xw) - \lambda = 0$$

$$-2x^Ty + 2x^Txw - \lambda = 0$$

$$2x^Txw = \lambda + 2x^Ty$$

$$w = \frac{2x^Ty + \lambda}{2x^Tx}$$

<u>(2) For w > 0:</u>

$$\underset{w}{\arg\min}\ \|\,\boldsymbol{y}-\boldsymbol{x}w\,\|^2 + \lambda\|\,w\,\|_1$$

$$(y-xw)(y-xw)^T + \lambda w = 0$$

$$-2x^T(y-xw) + \lambda = 0$$

$$-2x^Ty + 2x^Txw + \lambda = 0$$

$$2x^Txw = 2x^Ty - \lambda$$

$$w = \frac{2x^Ty - \lambda}{2x^Tx}$$

<u>(3) For w = 0:</u>

$$w = 0$$

ii. We suppose that when $\lambda = 0, w \neq 0$. There does exist a value for λ such that w = 0. The smallest value of $\lambda = 2|x^T y|$. From the previous parts of this problem, we observe:

(1) For w < 0:

$$w = \frac{2x^T y + \lambda}{2x^T x}$$
$$0 = \frac{2x^T y + \lambda}{2x^T x}$$
$$0 = 2x^T y + \lambda$$
$$\lambda = -2x^T y$$

(2) For w > 0:

$$w = \frac{2x^T y - \lambda}{2x^T x}$$
$$0 = \frac{2x^T y - \lambda}{2x^T x}$$
$$0 = 2x^T y - \lambda$$
$$\lambda = 2x^T y$$

Therefore, we observe that the smallest value of $\lambda = 2|x^T y|$

iii. $\arg\min_{w} \left|\left| \boldsymbol{y} - \boldsymbol{x}w \right|\right|^2 + \lambda\left|\left| w \right|\right|_2^2$

$$-x^T y + x^T x w + \lambda w = 0$$
$$-x^T y + x^T x w + \lambda I w = 0$$
$$-x^T y + w(x^T x + \lambda I) = 0$$
$$w(x^T x + \lambda I) = x^T y$$
$$w = x^T y(x^T x + \lambda I)^{-1}$$

iv. We suppose that when $\lambda = 0, w_i \neq 0$. There does not exist a value for $\lambda > 0$ such that $w_i \neq 0$ because if we consider the expression we derived above, $w_i = 0$ would require $x^T y(x^T x + \lambda I)^{-1} = 0$.