Homework 2

1 Comparing Different Loss Functions

Question A: For classification, the squared error would excessively penalize outliers and hypothesis that yield high $\mathbf{w}^T$x values will be severely penalized even if the $\mathbf{w}^T$x values have the same sign as their respective y and thus classified correctly.

Question B:

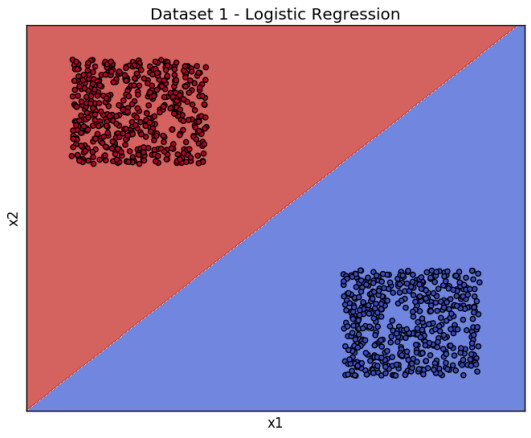
$$\nabla_w L_{\text{hinge}} = \begin{cases} 0 & y\mathbf{w}^T\mathbf{x} \geq 1 \\ -y\mathbf{x} & y\mathbf{w}^T\mathbf{x} < 1 \end{cases}$$

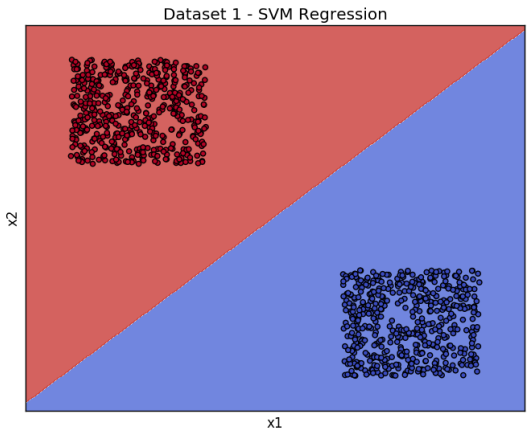
$$\nabla_w L_{\log} = \frac{-\mathbf{x}y}{e^{w^T xy} + 1}$$

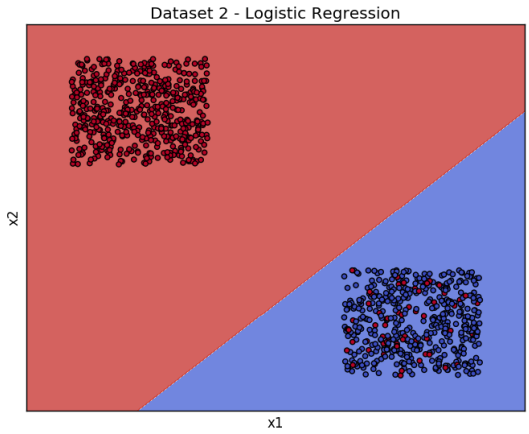
x_0	x_1	x_2	y	$\nabla_w L_{\text{hinge}}$	$\nabla_w L_{\log}$
1	0.5	3	1	(-1,-0.5,-3)	(-0.378, -0.189, -1.133)
1	2	-2	1	(0,0,0)	(-0.119, -0.238, 0.238)
1	-3	1	-1	(0,0,0)	(0.047, -0.142, 0.047)

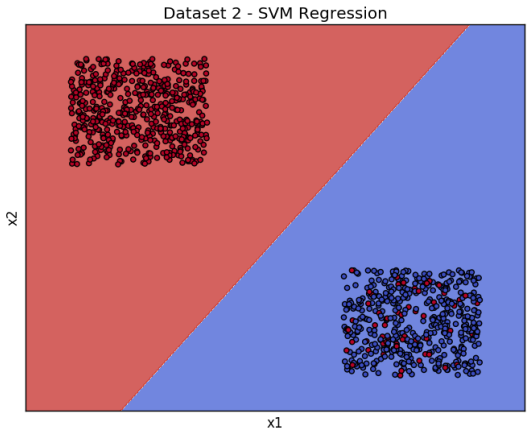
The gradient of log loss will go to zero as $\mathbf{w}^T\mathbf{x}$y gets infinitely big meaning each prediction is a large positive number when y = 1 or a large negative number when y = -1, which makes sense given that minimizing log loss is the equivalent of maximizing likelihood. The gradient of hinge loss will converge to 0 when all the points are correctly classified because in this case, y$\mathbf{w}^T\mathbf{x} \geq 1$ for all points. Thus, simply minimizing L_{hinge} is not enough for "maximum margin" classification, because minimizing L_{hinge} only ensures that points are classified correctly, but says nothing about margin. In order to maximize the margin, we must add the $\lambda\|w\|^2$ term and minimize $L_{\text{hinge}} + \lambda\|w\|^2$ for accurate classification AND maximum margin.

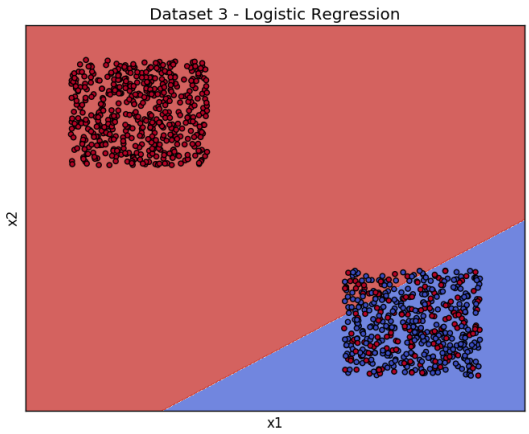
Question C:

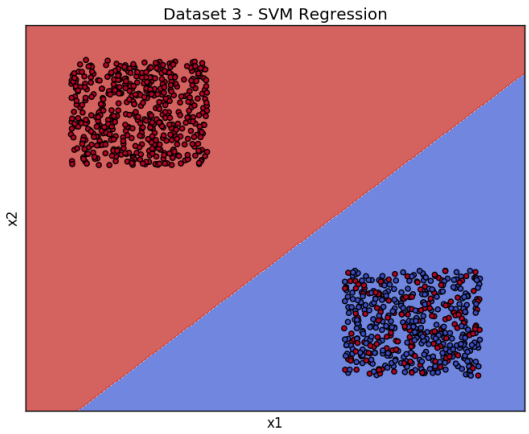






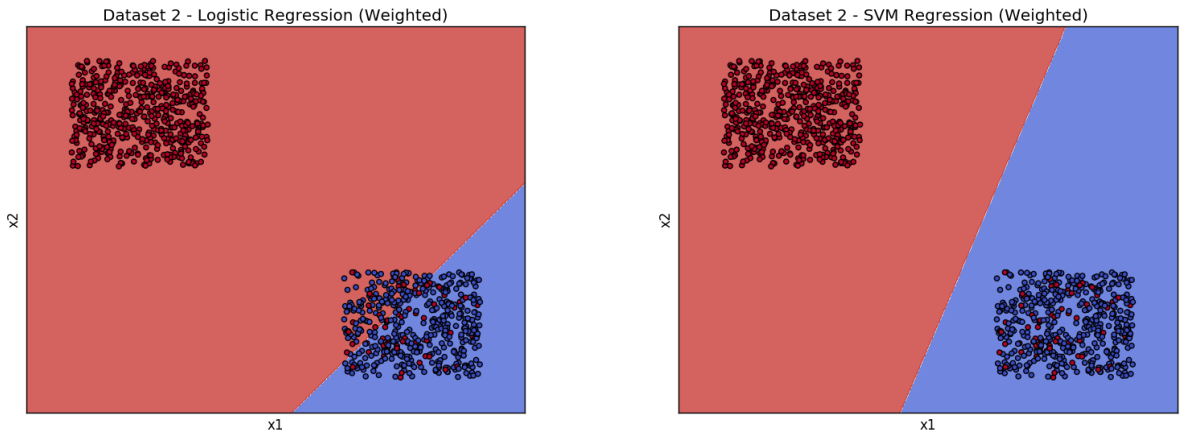






It appears that with more outliers, the logistic regression model has the boundary closer to the negatively labeled points. This makes sense because for the logistic regression model, the more noise, the more the boundary will be shifted to towards the errors to minimize the log error and thereby maximize likelihood. For the SVM however, we see a much more balanced or "even" divide between the two clusters and with increasing noise, the boundary shifts towards the negative cluster much less than the boundary shift of logistic regression with increasing noise. This makes sense given that because logistic regression is trying to maximize likelihood, it will shift the boundary farther towards the noisy cluster (even cutting through it in the case of 15% noise) to maximize the probabilities of correct classification. However, the SVM does not shift its boundary nearly as much because it is worried about maximum margin and harsh penalties for misclassification. In the case of 15% noise, unlike the logistic regression, the SVM does not cut through the data because in doing so would put it at risk of misclassifying more negative points than correctly classifying the positive points which are noise.

Question D:



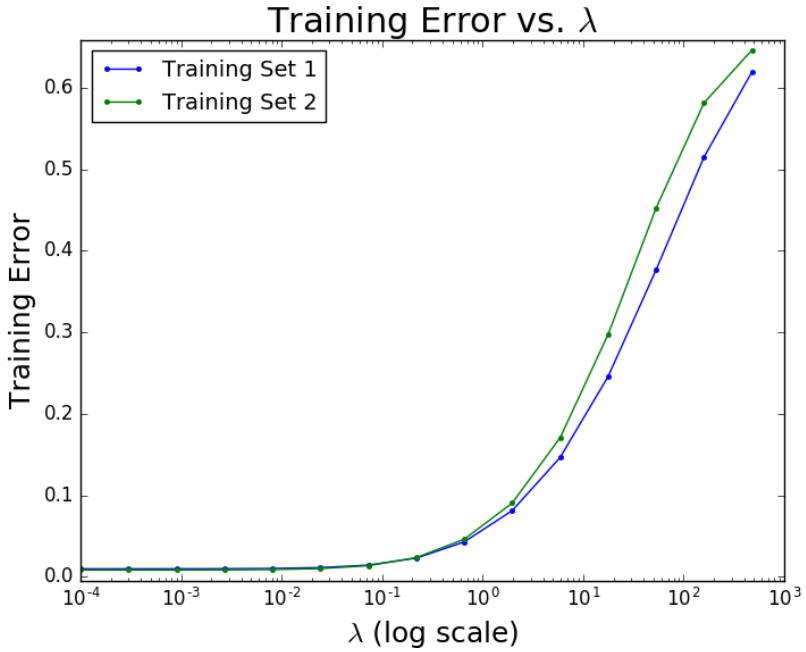
Both the logistic regression and SVM models shift towards the negative cluster which contains some noise meaning that there are some positive points in this cluster. This makes sense because now that positive points are weighed 5x that of negative points, the models now put more emphasis on correctly classifying the positive points in the negative cluster because with 5x the weight, the loss of not correctly classifying a positive point is much greater than that of misclassifying a negative point. Again, we see that the SVM regression's boundary is much more neutral than that of the logistic regression which can be attributed again to margin maximization and fear of misclassification.

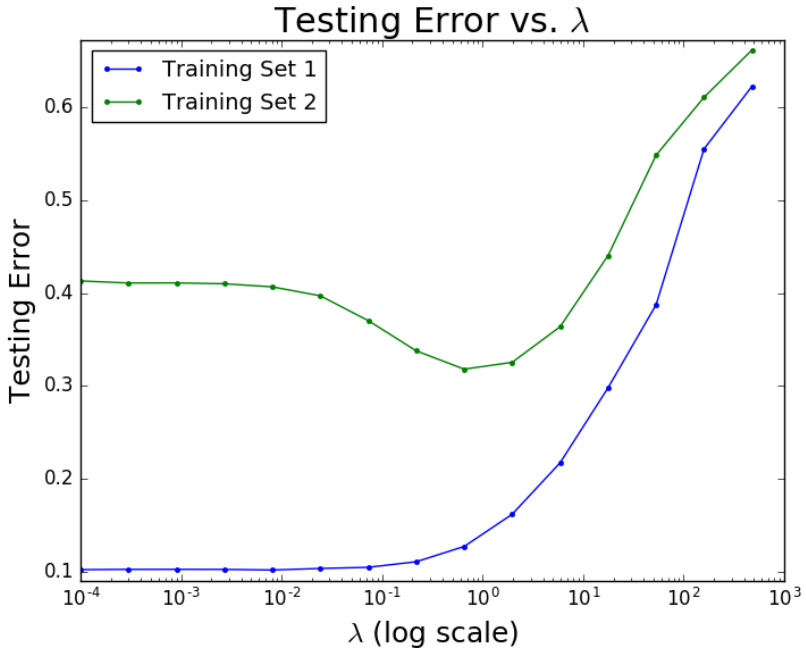
2 Effects of Regularization

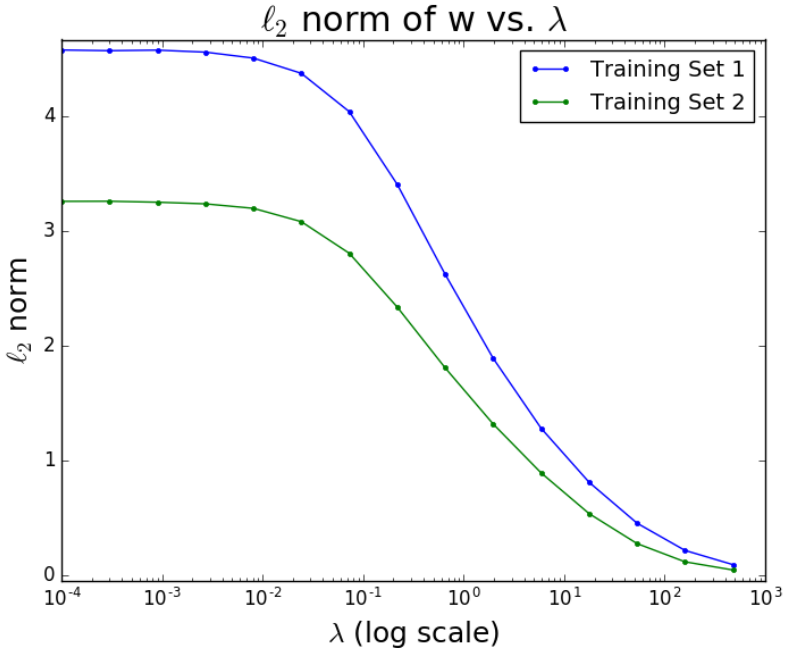
Question A: No, adding the term cannot decrease the training error. For least-squares linear regression, the unregularized, closed-form solution, is the optimal solution for minimizing in-sample error. Adding a penalty form cannot further decrease the training error. Adding a penalty term will not always decrease the out-of-sample error either. When there is overfitting, regularization can improve generalization by reducing variance. However, if regularization is overused and we reduce variance by too much, bias will increase and we may be at risk for underfitting. So, although regularization can decrease out-of-sample errors, it may not always do so as it depends on the parameters of regularization.

Question B: Unlike the ℓ_1 norm, the ℓ_0 norm is nonconvex and noncontinuous. This is problematic for minimization as derivatives/ gradients are necessary for the minimization of a regularized error. Thus, ℓ_0 regularization is rarely used.

Question C:


Training Error vs. λ
Training Set 1
Training Set 2
Training Error
λ (log scale)


Testing Error vs. λ
Training Set 1
Training Set 2
Testing Error
λ (log scale)



Question D: Given that second training set is a subset of the first training set and thereby contains fewer training points, we expect and observe the testing error resulting from training set 1 to be lower than that from training set 2 (more training examples decreases variance, bias about the same with constant λ, so thus testing error decreases). We see clear overfitting for set 2 with increasing testing error as λ decreases for the first half of the graph while overfitting is not as clear for set 1, again confirming that variance decreases with increased training size.

Regarding training error, training set 1 has a higher(ever so slightly) E_{in} at lower λ and lower E_{in} at high λ, when compared to the training errors for training set 2. This can be attributed to the fact that because training set 2 is a subset of training set 1, at lower λ, complex models will be able to fit the dataset containing fewer points better, thus explaining why training set 1 has the higher E_{in} for small λ. For large λ and thus simpler models, points with larger error for the smaller dataset (training set 2) will negatively affect the E_{in} of the smaller dataset more than the E_{in} of the larger data set, thus why the training error for training set 1 is lower for large λ.

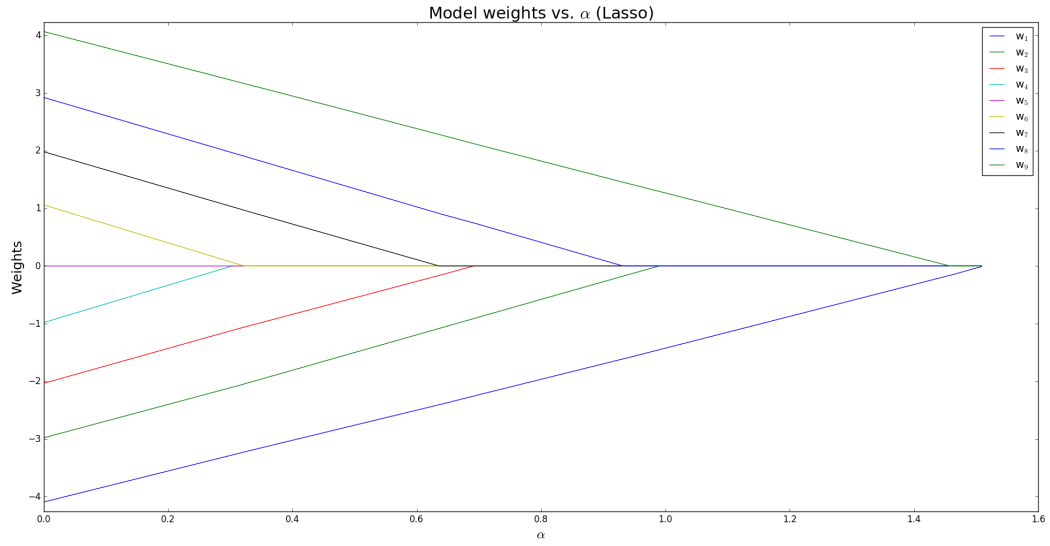
Question E: There seems to be little overfitting. Using Python, I found that out of the 15 points plotted, the minimum test error came on point 3, when $\lambda = 9.00 \text{ x } 10^{-3}$, but for the λ values less than this, the testing error is only slightly bigger (hard to qualitatively tell from the graph as it appears that the testing error starts at the minimum from the first λ.) This contrasts with the testing error from fitting to set 2 where for the first half of the graph, we have increasing testing error with decreasing λ, a true sign of overfitting. However, in both cases, we do see underfitting. With increasing λ, especially after $\lambda = 1$, the training and testing errors vastly increase, signaling that for these large λ, we are regularizing so much that the constrained model is too simple to capture the dataset accurately.

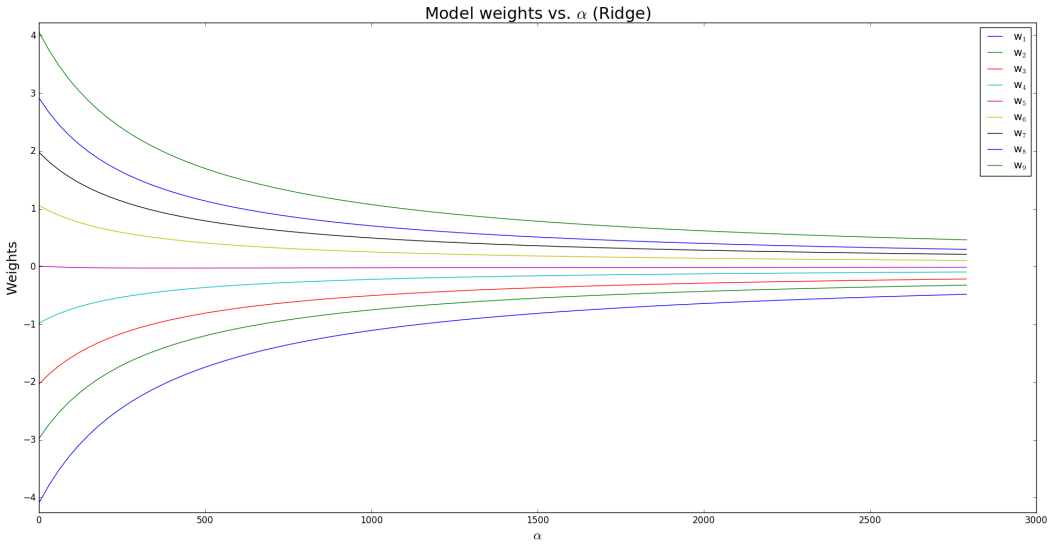
Question F: We see that the ℓ_2 norm of $\mathbf{w}$ decreases with increasing λ. This makes sense because with increasing λ, there is a greater emphasis on the regularization term of the ℓ_2-regularized logistic error.

Question G: Using Python, I found that the minimum testing error came from $\lambda = 0.6561$. At this λ, we have the lowest E_{out}, meaning the best chance of generalizing out of sample.

3 Lasso vs. Ridge Regularization

Question A:





As the regularization parameter increases, the number of model weights that are exactly zero with lasso regression increase. For Ridge regression, the weights get closer and closer to zero, but none of the model weights actually become zero (asymptotic).

Question B:

i. We take the subgradient.

$$\begin{aligned} &\nabla_{w_1}\left[\left\|\mathbf{y}-\mathbf{X}^T\mathbf{w}\right\|^2+\lambda\|\mathbf{w}\|_1\right] \\ &=-2\mathbf{X}^T(\mathbf{y}-\mathbf{X}^T\mathbf{w})+\lambda=0 \\ &\rightarrow \mathbf{w}=-(\lambda-2\mathbf{X}^T\mathbf{y})(2\mathbf{X}^T\mathbf{X})^{-1} \end{aligned}$$

ii. Yes, when $\lambda=\left\|2\mathbf{X}^T\mathbf{y}\right\|$, w_1 goes to 0.

iii.

$$\begin{aligned} &\partial_w\left[\left\|\mathbf{y}-\mathbf{X}^T\mathbf{w}\right\|^2+\lambda\|\mathbf{w}\|_1^2\right] \\ &=-2\mathbf{X}^T(\mathbf{y}-\mathbf{X}^T\mathbf{w})+2\lambda\mathbf{w}=0 \\ &\rightarrow \mathbf{w}=\mathbf{X}^T\mathbf{y}(\lambda I+\mathbf{X}^T\mathbf{X})^{-1} \end{aligned}$$

iv. No, there does not exist a $\lambda>0$ such that $\mathrm{w}_i=0$.

Let's say, for the sake of argument, that $\mathbf{w}=0$. Now, we multiply both sides by $\lambda I+\mathbf{X}^T\mathbf{X}$ (we know $\lambda>0$) and get that $0=\mathbf{X}^T\mathbf{y}$ which clearly does not depend on λ. This is consistent with our previous findings of Ridge regression, where weights get closer to 0, but never actually equal 0.