Homework 2

Problem 1.

1. I implement a Matlab function that minimizes the logistic regression objective, but with L2 regularization on the weight vector, using the function fminunc to compute the minimum. The regularization term is an L1 norm on w and doesn't include the offset w_0.
There is no closed expression for the optimal values of w and w_0, but we can compute them using the gradient descent method (or a built in matlab function in this case).

2. The datasets contain 400 2D data points each, with labels $y = +1$ or $y = -1$. We set $\lambda = 0$ and run the previous logistic regression classifier on each of the 4 datasets (train) and test it on the validate dataset.
We obtain the following table (table 1) summarizing the offset, weight vector and error rate on the train and validate set:

Dataset	w_0	w	error rate in train set	error rate in validate set
stdev1	-22.4496	(95.3841,101.2140)	0	0
stdev2	-0.0470	(0.7638,1.1144)	0.0925	0.08
stdev4	-0.0093	(0.2363,0.2034)	0.26	0.2475
nonsep	0.0006	(-0.0248,-0.0237)	0.485	0.5075

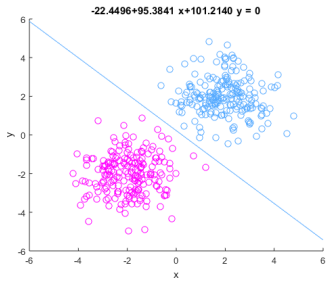
Table 1: table of weight and error values for the 4 datasets using the logistic regression classifier with L1 regularization, $\lambda = 0$.

We see that the error rate on stdev1 is 0 for both train and validate set.
For both stdev2 and stdev4, the train error rate is slightly higher than the validate error rate.
The error is high for the nonsep dataset. That's because we can't use a linear separator to classify the data, so the error is around 0.5.
Using the table above, we can compute the decision boundary equation, we then obtain the following plots (fig.1-4 in page 2) for all 4 datasets (train and validate) for $\lambda = 0$:

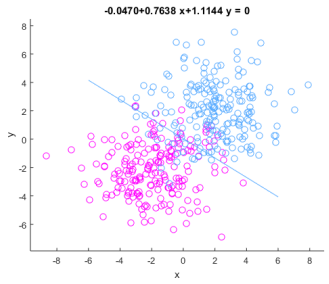
3. The following table gives the weight and error values for the 4 datasets using the logistic regression classifier with L1 regularization, $\lambda = 1000$.

Dataset	w_0	w	error rate in train set	error rate in validate set
stdev1	0.008	(0.1370,0.1409)	0	0
stdev2	0.0013	(0.1208,0.1338)	0.095	0.0625
stdev4	-0.0029	(0.0936,0.0832)	0.2625	0.2375
nonsep	0.0001	(-0.0048,-0.0047)	0.4825	0.4975

Table 2: table of weight and error values for the 4 datasets using the logistic regression classifier with L1 regularization, $\lambda = 1000$.

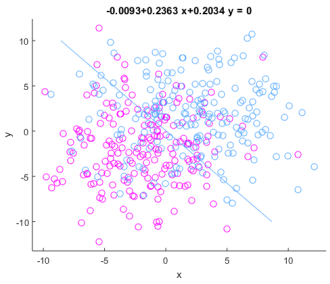


(a) stdev1 train

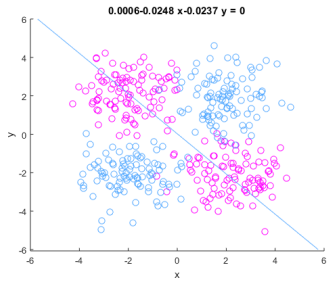


(b) stdev2 train

Figure 1: data points and decision boundary with $\lambda = 0$

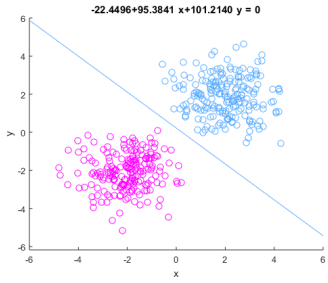


(a) stdev4 train

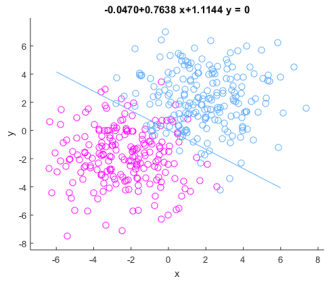


(b) nonsep train

Figure 2: data points and decision boundary with $\lambda = 0$

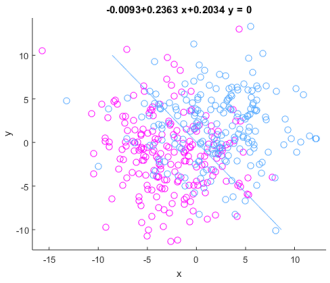


(a) stdev1 validate

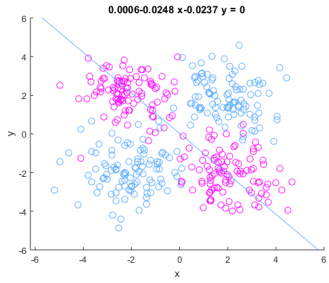


(b) stdev2 validate

Figure 3: data points and decision boundary with $\lambda = 0$



(a) stdev4 validate



(b) nonsep validate

Figure 4: data points and decision boundary with $\lambda = 0$

In table 2, we see that we still have 0 error rate on stdev1 (train and validate).
We obtain the same error rate on stdev2 and stdev4 train sets but the validate error rate slightly improves when we increase λ.
The error rate is slightly improved for the nonseparate dataset (train and validate). Looking at the data, we expect it to stay around 0.5 when using a linear separator.
In this case, when λ is very large, the offset tends to 0 and the weight vector tends to $(0.5, -0.5)$.
We can look at the errors on the train and validate set when we increase $\lambda = 10^6$ in table 3:

Dataset	error rate in train set	error rate in validate set
stdev1	0	0
stdev2	0.95	0.0625
stdev4	0.2625	0.2375
nonsep	0.49	0.4925

Table 3: table of weight and error values for the 4 datasets using the logistic regression classifier with L1 regularization, $\lambda = 10^6$.

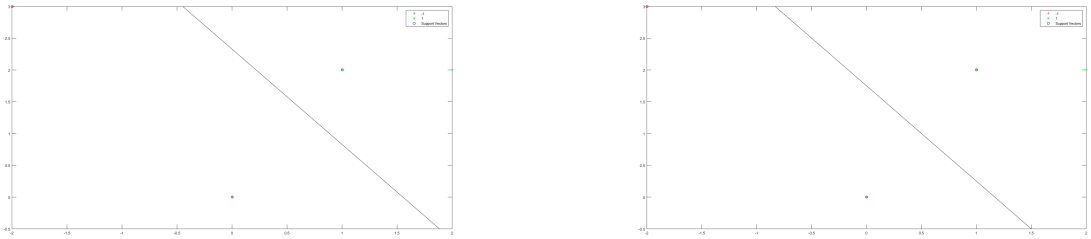
Problem 2. 1. I implement the dual form of linear SVM with slack variables. The function takes the data as input and returns the alpha vector and the support vectors, allowing to compute the bias, the weight vector and the misclassification error.
For the 2D example provided, we write the problem as follows:

$$min \sum_{i=1}^{4} \alpha_i - \frac{1}{2}(5\alpha_1^2 + 12\alpha_1\alpha_2 - 8\alpha_1\alpha_4 + 8\alpha_2^2 - 4\alpha_2\alpha_4 + 13\alpha_4^2) \tag{1}$$

$$st. \alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 = 0 \tag{2}$$

$$\alpha_i \in [0, C] \tag{3}$$

We solve using the classifier implemented in question 1 and obtain the following plots (fig. 5):

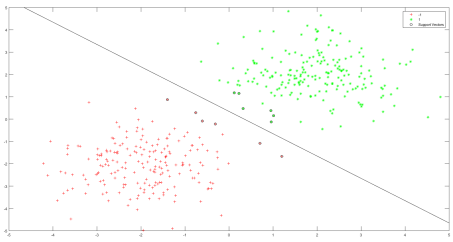


(a) C=1 (b) C=100

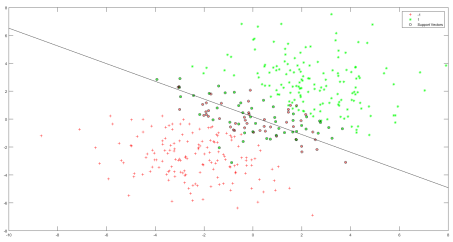
Figure 5: SVM for the provided example with $C = 1$ and $C = 100$

2. We test implementation on the same 2D datasets from problem 1 and obtain the following plots (fig. 6-7):

$C = 1$
datasets: stdev1, stdev2, stdev4 and nonsep.

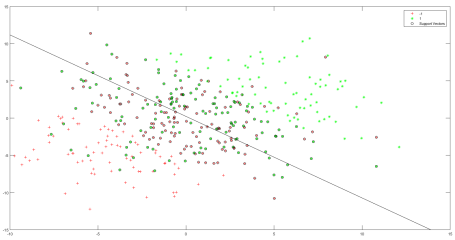


(a) stdev1

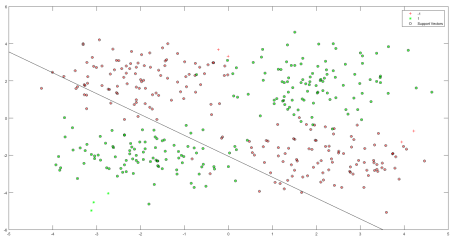


(b) stdev2

Figure 6: SVM for stdev1 and stdev 2 with $C = 1$



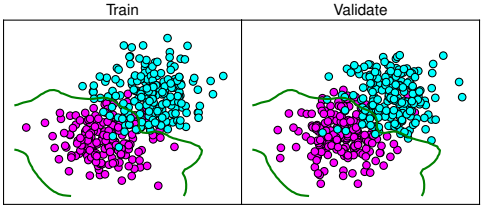
(a) stdev4



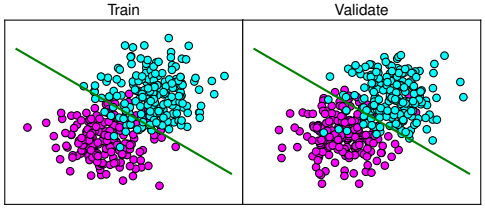
(b) nonsep

Figure 7: SVM for stdev4 and nonsep with $C = 1$

The following figures summarize the decision boundaries and error rates on train and validate set (fig: 8-10):

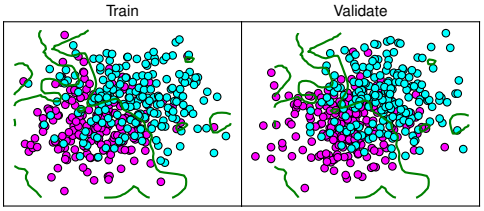


(a) stdev2 using a Gaussian Kernel, $C = 0.1$, $\beta = 1$, Error = .08

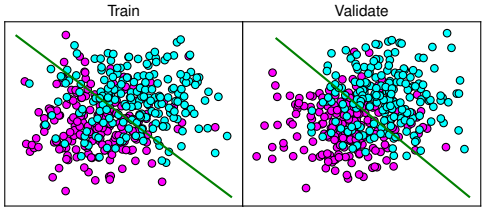


(b) stdev2 using a Linear Kernel, $c = 0.1$, Error = .20

Figure 8

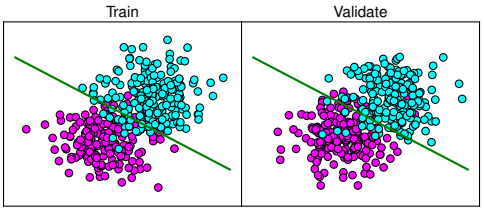


(a) stdev4 using a Gaussian Kernel with $\beta = 1$, $C = 0.01$, Error = .22

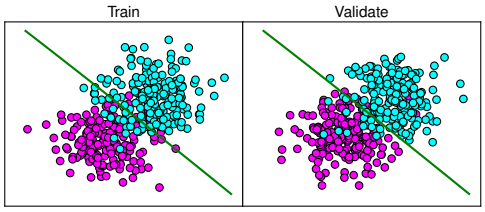


(b) stdev4 using a Linear Kernel, $c = 0.1$, Error = .45

Figure 9



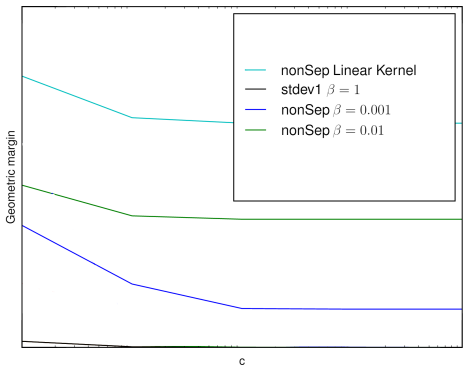
(a) Linear kernel SVM on stdev2, $C = 1$. Error of .18.



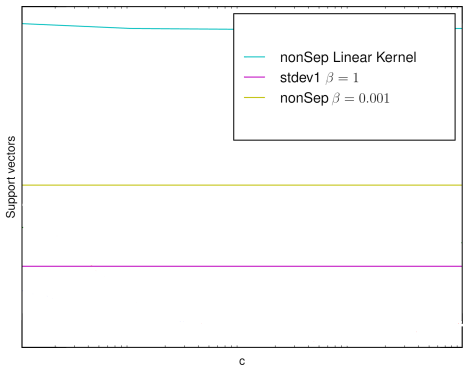
(b) Gaussian kernel SVM with $\beta = 0.001$ and $C = 1$ on stdev2. Error of .18.

Figure 10

3. a)- As C increases, the geometric margin decreases, as shown for various values of C, various kernels, and various datasets in figure 11.(a). This always happens as C increases, because this means there can be more slack in the final SVM, which means the SVM will be more tolerable to incorrect classifications for inseparable data.
b)-The number of support vectors first decreases, then increases as C increases, as shown in figure 11.(b). This means that the classifier is less overfit on the training data and will have smaller errors on the testing data.
c)- Choosing C to maximize the margin will yield a value of C equal to zero, which is the same as having a hard-margin SVM that does not perform well on non-separable data. An alternate criteria for choosing C could be the minimum number of support vectors (since the number of support vectors eventually increases with a higher value of C as the classifier gets over-fit to the training data). Changing C has almost no effect on the training error unless the data is highly non-separable.



(a) Geometric margin as a function of C



(b) Number of support vectors as a function of C

Figure 11

Problem 3.

1. First, we scale the data by substracting the mean and dividing by the standard deviation of the train set. We scale the validate data set using the same mean and standard deviation.
Then we run the logistic regression classifier and obtain the following (table 4):

Dataset	error rate in train set	error rate in validate set	error rate in test set
Titanic	0.17	0.202	0.222

Table 4: table of train, validate and test error for the titanic data, using the logistic regression classifier.

The weight vector is: (0.3680 0.2720 -0.5651 1.3660 -0.3941 -0.1740 0.1653 0.0950 -0.0728 0.1218 -0.0547) with a bias of -0.7705.

2. After scaling the data, we run the SVM classifier and obtain the following results for different values of C:

C	bias	weight vector
0.01	0.4864	(-0.0236,-0.0494,0.0639,-0.5264,0.05271,0.02700,-0.0301,-0.0958,0.0303,-0.0346,1.25E-16)
1	0.2802	(2.15E-05,-7.05E-05,4.14E-05,-0.9622,0.0001,0.0003,-0.0001,-0.0004,-7.09E-05,1.09E-05,9.92E-05)
100	0.2805	(-0.0001,-6.9E-05,0.0001,-0.9622,3.1E-05,0.0001,-0.0003,-2.3E-06,-7.5E-07,-4.00671E-05,5.7E-05)

Table 5: table of bias and weight vector using different C on titanic data.

The following table summarizes the error rates:

C	train error rate	validate error rate
0.01	0.39	0.48
1	0.28	0.38
100	0.19	0.24

Table 6: table of error rates using different C on titanic data.

It seems that higher C yield lower error rates.

3. The logistic regression classifier seems to give lower error rates.
According to this classifier, the most significant feature is data.4 : sex (sex 1 is more likely to survive).
According to the SVM classifier (for $C = 100$), the most significant feature is also data.4 :sex.
It is worth mentioning that logistic regression is computationally more expensive than SVM, although the classification problem (using linear separator) is roughly the same.