Question 1:

We aim to minimize:

$$\frac{1}{2}w^T Sw - vp + \sum_t C^t \varepsilon^t$$

With respect to w, w0, p, and epsilon^t. We also have the following constraints:

$$r^t(w^T x^t + w_0) \geq \rho - \varepsilon^t, \varepsilon^t \geq 0, \rho \geq 0$$

Which means we must add the following Lagrange terms, yielding the primal:

$$L_p = \frac{1}{2}w^T Sw - vp + \sum_t C^t \varepsilon^t - \sum_t \alpha^t (r^t(w^T x^t + w_0) - \rho + \varepsilon^t) - \sum_t \mu^t \varepsilon^t - \beta\rho$$

Where:

$$\alpha^t \geq 0, \mu^t \geq 0, \beta \geq 0$$

To make the rest of the derivation simpler, we expand out the terms in the primal to yield:

$$L_p = \frac{1}{2}w^T Sw - vp + \sum_t C^t \varepsilon^t - w^T \sum_t \alpha^t r^t x^t - w_0 \sum_t \alpha^t r^t + \rho \sum_t \alpha^t - \sum_t \alpha^t \varepsilon^t - \sum_t \mu^t \varepsilon^t - \beta\rho$$

Now we take the derivative with respect to all of the parameters and see what information we can obtain. First we take the derivative with respect to w:

$$\frac{\delta L_p}{\delta w} = Sw - \sum_t \alpha^t r^t x^t = 0$$

Which implies:

$$w = S^{-1} \sum_t \alpha^t r^t x^t$$

Next we take the derivative with respect to w0:

$$\frac{\delta L_p}{\delta w_0} = \sum_t \alpha^t r^t = 0$$

Next we take the derivative with respect to the epsilon terms:

$$\frac{\delta L_p}{\delta \varepsilon^t} = C^t - \alpha^t - \mu^t = 0$$

Which implies:

$$C^t = \alpha^t + \mu^t$$

However, since all mu terms are >= 0, we can further say:

$$0 \leq \alpha^t \leq C^t$$

Finally, taking the derivate with respect to rho, we obtain:

$$L_p = -v + \sum_t \alpha^t - \beta = 0$$

Which implies:

$$v = \sum_t \alpha^t - \beta$$

But since beta >= 0, we can say:

$$v \leq \sum_t \alpha^t$$

Now we derive the dual by substituting these values into the primal:

$$L_d = \frac{1}{2}\left(S^{-1}\sum_t \alpha^t r^t x^t\right)^T S(S^{-1}\sum_t \alpha^t r^t x^t) - \left(\sum_t \alpha^t - \beta\right)\rho + \sum_t (\alpha^t + \mu^t)\varepsilon^t$$
$$- \left(S^{-1}\sum_t \alpha^t r^t x^t\right)^T (\sum_t \alpha^t r^t x^t) - w_0(0) + \rho \sum_t \alpha^t - \sum_t \alpha^t \varepsilon^t - \sum_t \mu^t \varepsilon^t - \beta\rho$$

And simplifying this yields:

$$L_d = -\frac{1}{2}\left(\sum_t \alpha^t r^t x^t\right)^T (S^{-1})^T (\sum_t \alpha^t r^t x^t)$$

Which we maximize with respect to alpha^t subject to:

$$\sum_t \alpha^t r^t = 0, 0 \leq \alpha^t \leq C^t, v \leq \sum_t \alpha^t$$

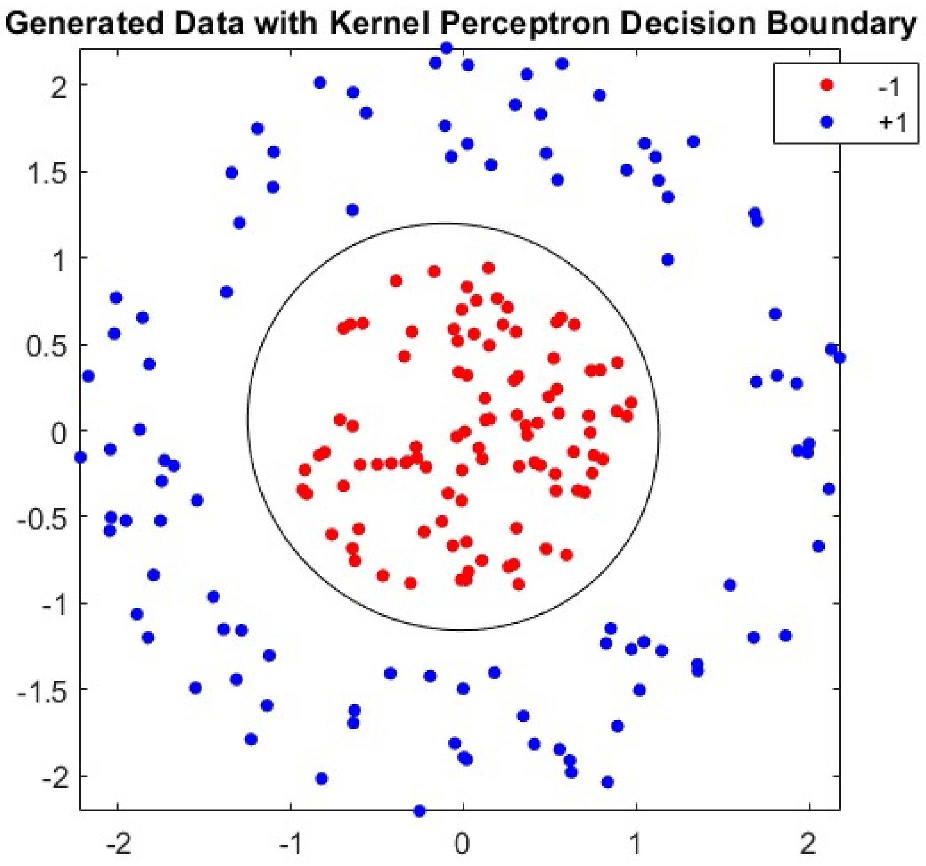
Thus we've obtained the desired result.

Question 2:

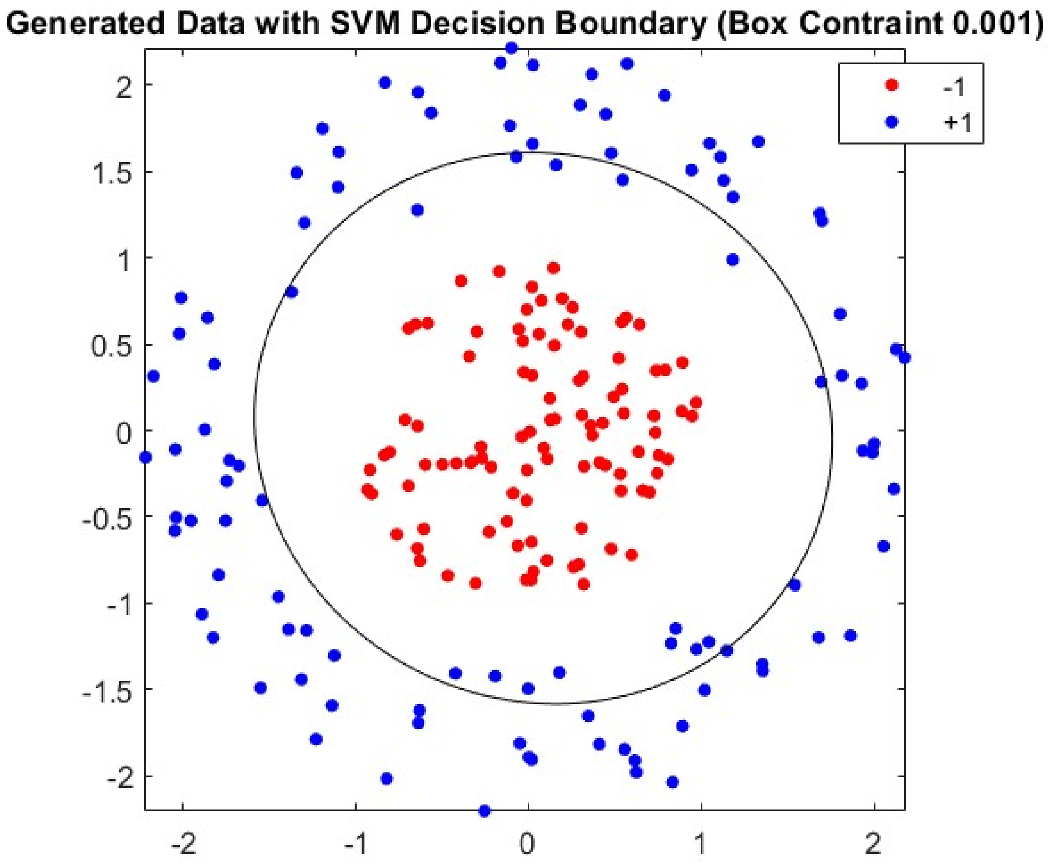
a. After playing around with different degrees for the kernel, I eventually settled on 2 since this classified the data nicely (0 error rate) and was a simpler model than choosing a higher degree polynomial. Below is the output from the MATLAB terminal related to the generated data:

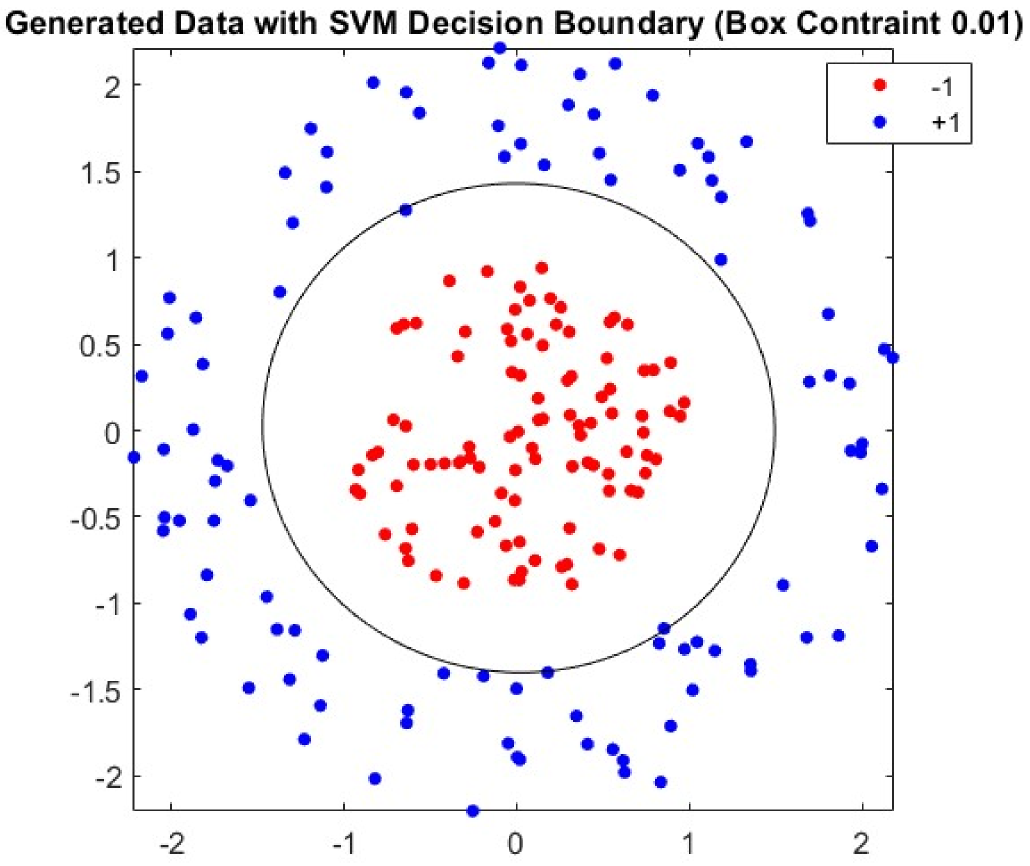
```
Generated Data Error Rate: 0
```

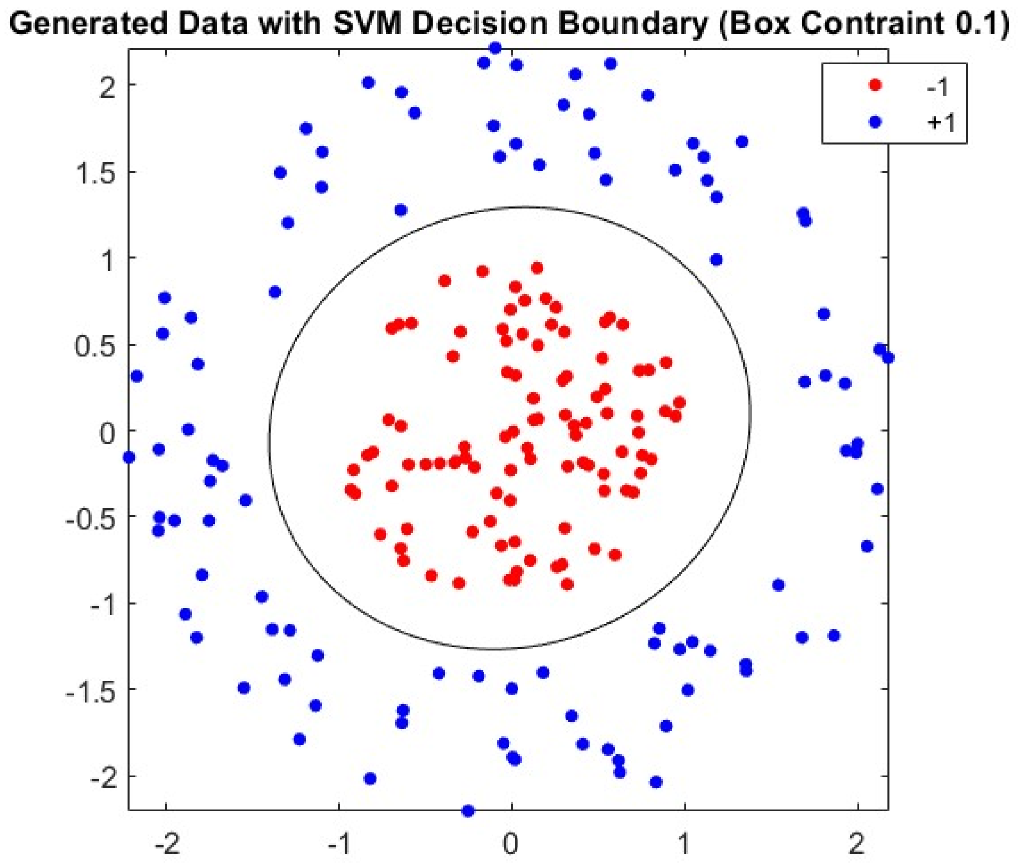
We can see from looking at the graph generated by my script why the error rate is 0. Namely, the boundary perfectly separates the blue points from the red points:

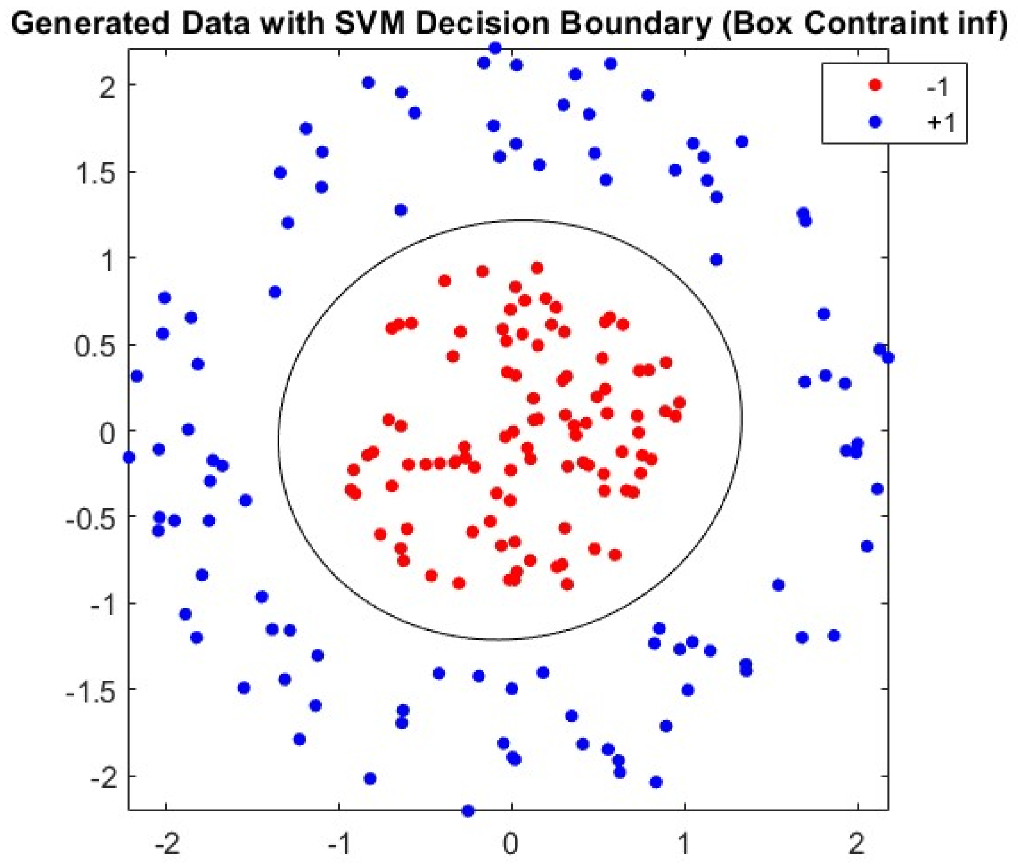


b. For the SVM classifier, I used fitcsvm with a polynomial kernel of degree 2 (same as the kernel in the kernel perceptron), and made plots for various box constraint values. When lowering the box constraint, we can see that the boundary moves outwards and points begin to get misclassified. This is because lowering the box constraint lowers the penalty for misclassification, and thus makes misclassification more likely. Below are graphs of the boundary curves for the SVM classifier with various box constraint values:


Generated Data with SVM Decision Boundary (Box Contraint 0.001)
-1
+1
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2


Generated Data with SVM Decision Boundary (Box Contraint 0.01)
-1
+1
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2


Generated Data with SVM Decision Boundary (Box Contraint 0.1)
-1
+1
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2



NOTE: For parts A and B, I did not put the SVM curve and Kernel Perceptron curves in the same graph. This is because I thought it was clearer to separate them (partly because I was unable to figure out how to label the curves in a meaningful way). This was approved by Zhuliu (who I assume will be grading this), but I just wanted to mention it since technically the HW instructions say to put these on one plot.

c. For both the optdigits49 dataset and the optdigits79 dataset, the kernel perceptron was trained on the training set, and then was used to classify both the training and test set. Below is the terminal output generated by my MATLAB script related to the optdigits data sets:

```
Optdigits49 Training Error Rate: 0.0047281
Optdigits49 Test Error Rate: 0.03169
Optdigits79 Training Error Rate: 0.0035461
Optdigits79 Test Error Rate: 0.01773
```

We can see that even with a polynomial kernel of only degree 2, we are able to achieve very high classification accuracy on the test sets. We also see that the algorithm performs better on the 79 set than on the 49 set, which is not surprising, since 4 and 9 can look pretty similar depending on how they are drawn.