we have the given objective function.

minimize. $\frac{1}{2} w^T S w - \nu \rho + \sum_t c^t \xi^t$

subject to

$\Rightarrow r^t (w^T x^t + w_0) \geq \rho - \xi^t$

$\Rightarrow \xi^t \geq 0$

$\rho \geq 0.$

$\Rightarrow S$ is a positive definite

$\Rightarrow c^t > 0 \quad \forall t$

$\Rightarrow \nu \in [0, 1]$

we have the L_p which includes the lagrange multiplier

$$L_p = \frac{1}{2} w^T S w - \nu \rho + \sum_t c^t \xi^t$$

$$- \sum \alpha^t \left[r^t \left(w^T x^t + w_0 \right) - \rho + \xi^t \right]$$

$$- \sum_t \mu^t \xi^t - \eta \rho \quad \text{------} (1)$$

The above equation is a quadratic convex optimization problem we will now solve the dual problem using Karush - Kuhn Tucker conditions, subject to the gradient L_p w.r.t w, w_0, ρ, ξ^t should be $= 0$.

$$\frac{\partial L_p}{\partial w} = S w - \sum_t \alpha^t r^t x^t$$

since S is a positive semidefinite matrix

$$\Rightarrow S w = \sum_t \alpha^t r^t x^t \quad \text{------} (II)$$

$$\frac{\partial L_p}{\partial w_0} = - \sum_t \alpha^t r^t = 0.$$

$$\sum_t \alpha^t r^t = 0 \quad \text{------} (III)$$

$$\frac{\partial L_p}{\partial \rho} = -\vartheta + \sum_t \alpha^t - \eta = 0$$

$$\vartheta = \sum_t \alpha^t - \eta \quad \text{——} \ (iv)$$

$$\frac{\partial L_p}{\partial \xi} = c^t - \alpha^t - \mu^t$$

$$\mu^t = c^t - \alpha^t \quad \text{——} \ (v)$$

we now substitute (ii), (iii), (iv), (v) into (i) to get

$$Ld = \frac{1}{2} \omega^T \sum \alpha^t r^t x^t - \omega^T \sum \alpha^t r^t x^t$$

$$- \rho \sum \alpha^t + \eta \rho - \eta \rho + \rho$$

$$+ \sum_t c^t \xi^t - \sum_t \alpha^t \xi^t - \sum (c^t -$$

$$Ld = -\frac{1}{2} \omega^T \sum \alpha^t r^t x^t \quad \text{——} \ (vi)$$

we use the value of ω^T with (ii)

$$\omega^T = \sum_s \alpha^s r^s (x^s)^T$$

adding this value to (vi)

$$Ld = -\frac{1}{2} \sum_s \alpha^s r^s (x^s)^T S^{-1} \cdot \sum_t \alpha^t r^t$$

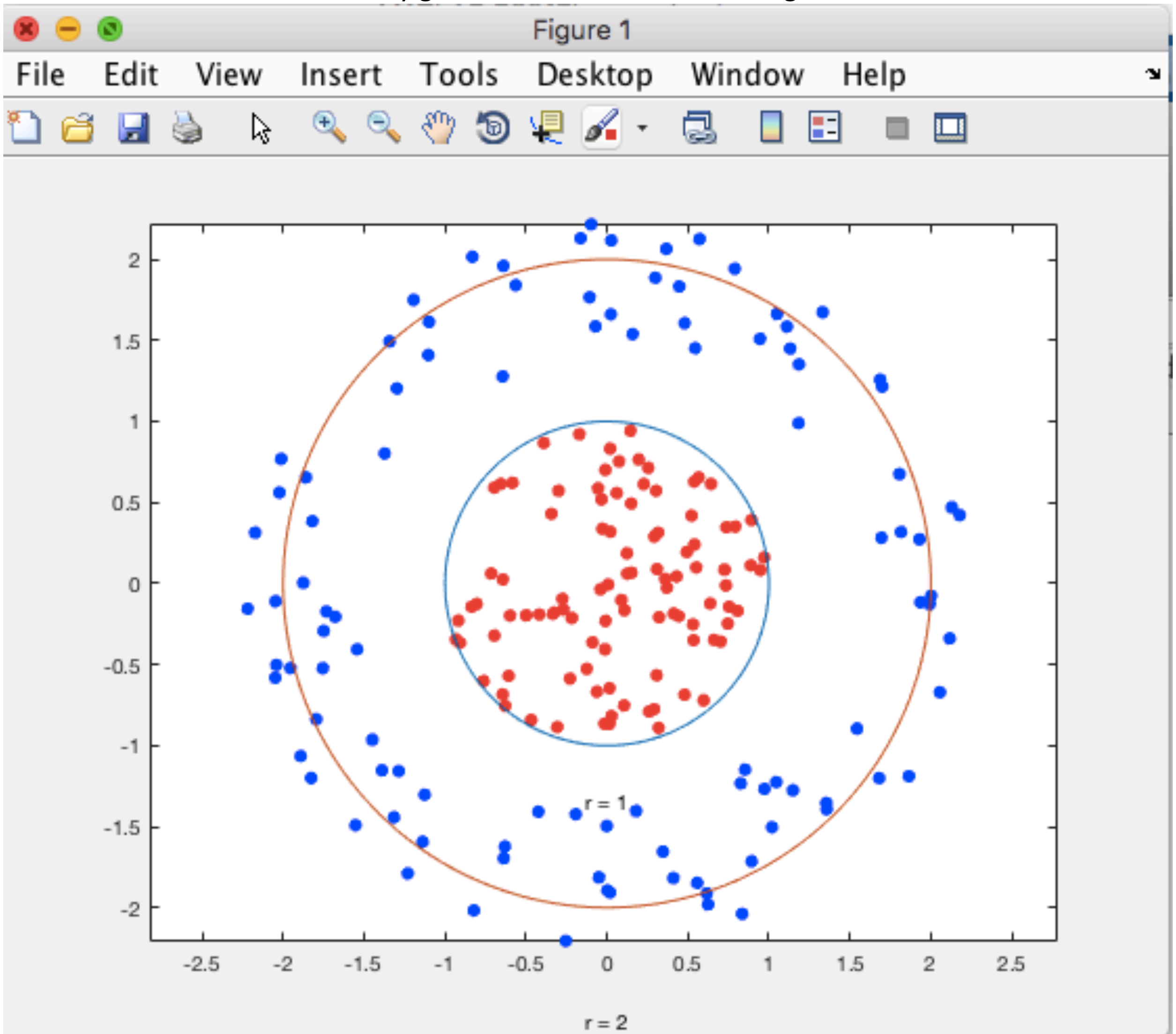
$$= -\frac{1}{2} \sum_s \sum_t \alpha^s \alpha^t r^s r^t (x^s)^T S^{-1}$$

subject to

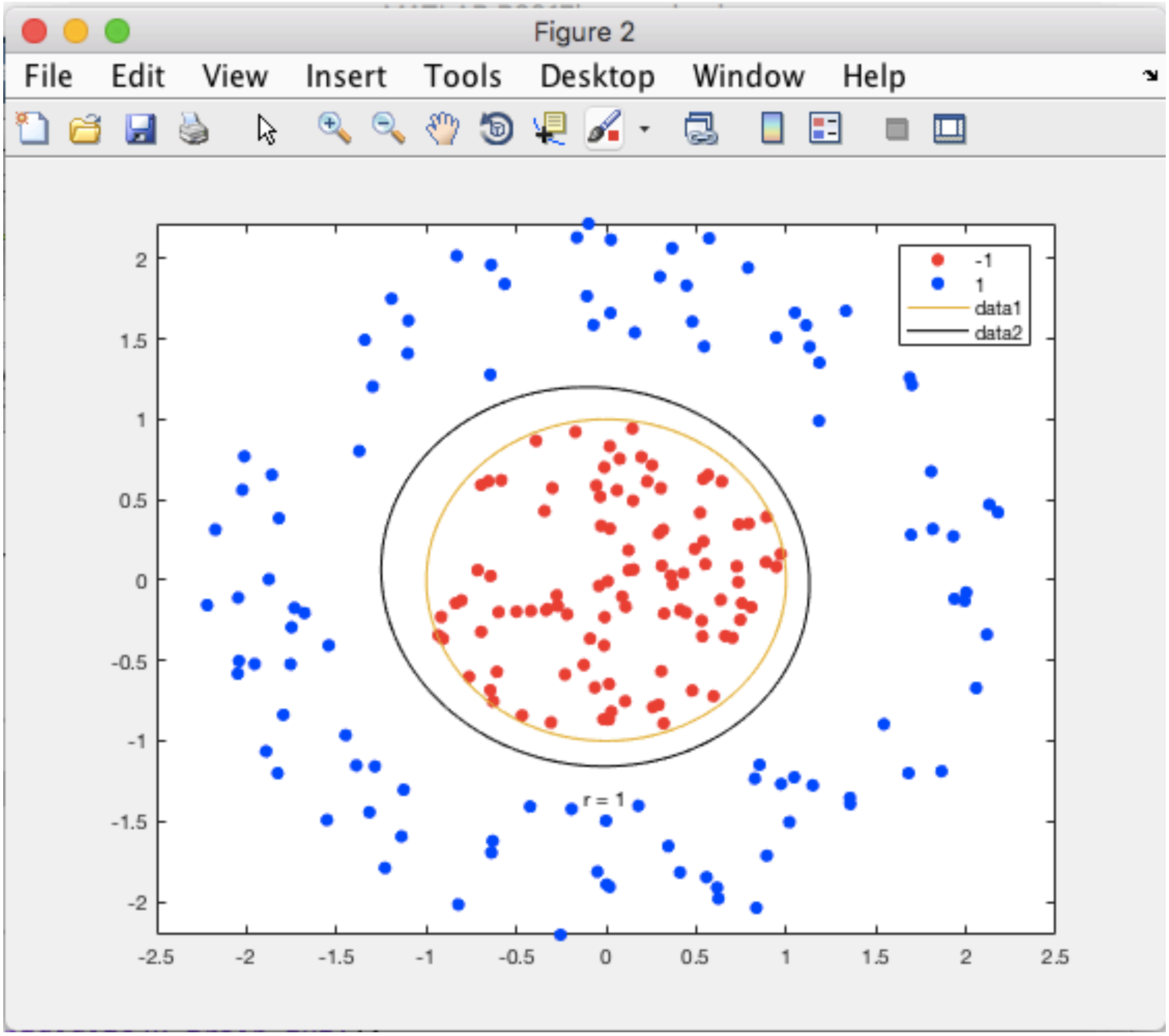
$$\sum_t \alpha^t r^t = 0 \quad \& \quad 0 \leq \alpha^t \leq c^t.$$

Q2

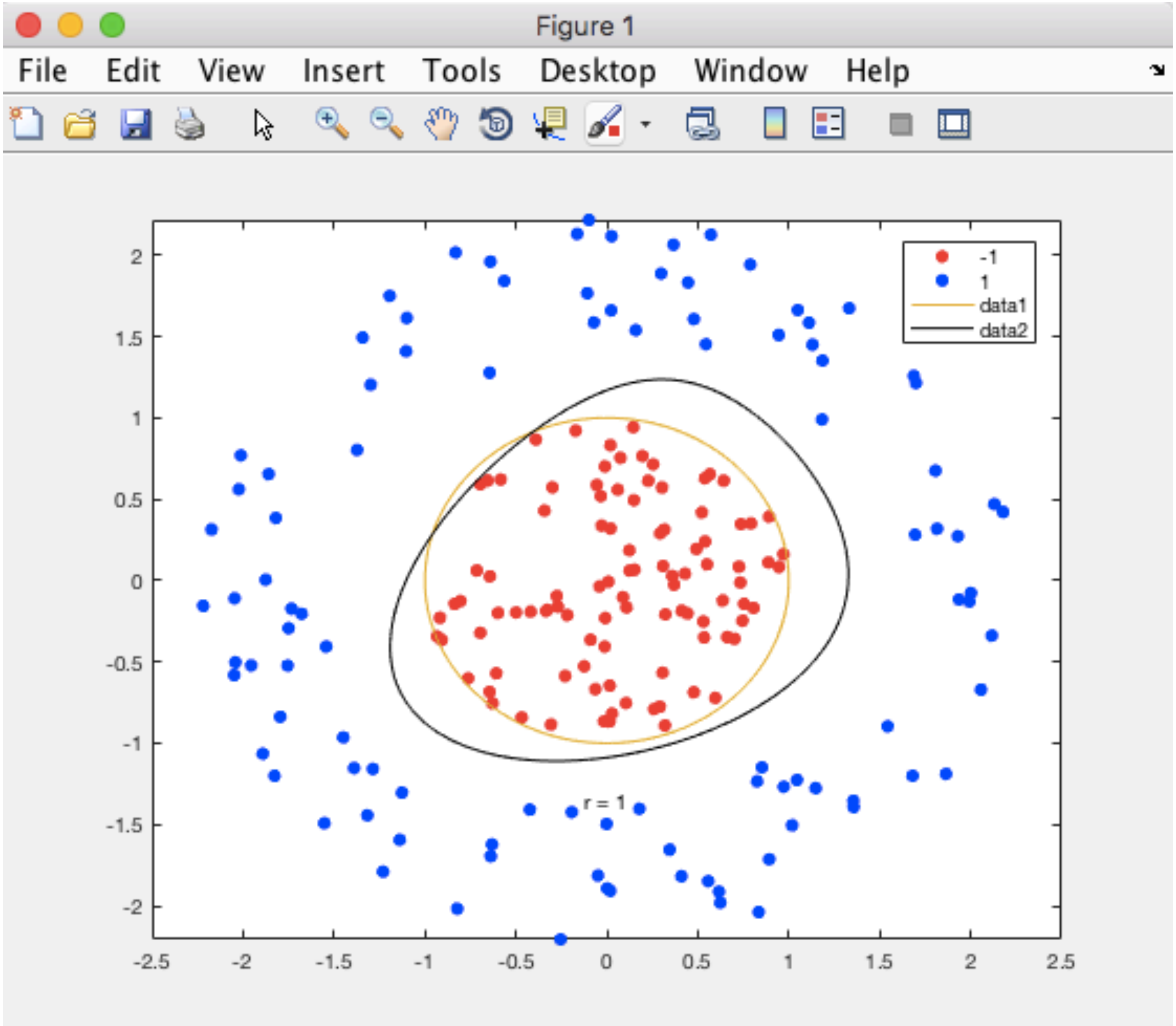
The data cluster for the randomly generated data is the following



Using a q value of 2, we get the following decision boundary for the polynomial kernel perceptron



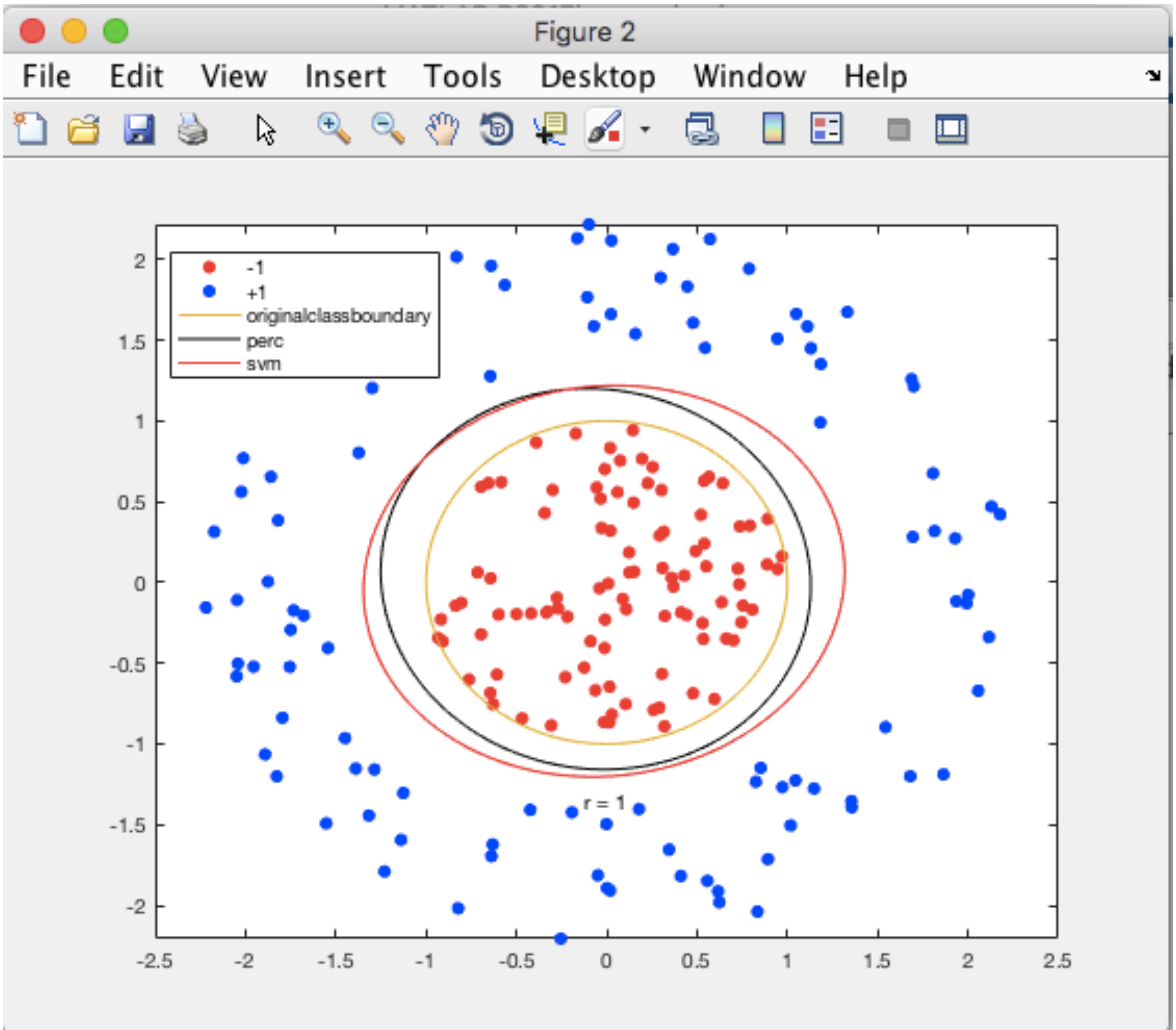
The yellow line approximates the circle around which the data was generated which we use to understand which polynomial order best mimics this decision boundary.
If we use q=3 we get the following,



This gets worse with increasing q, so we stick to q=2.

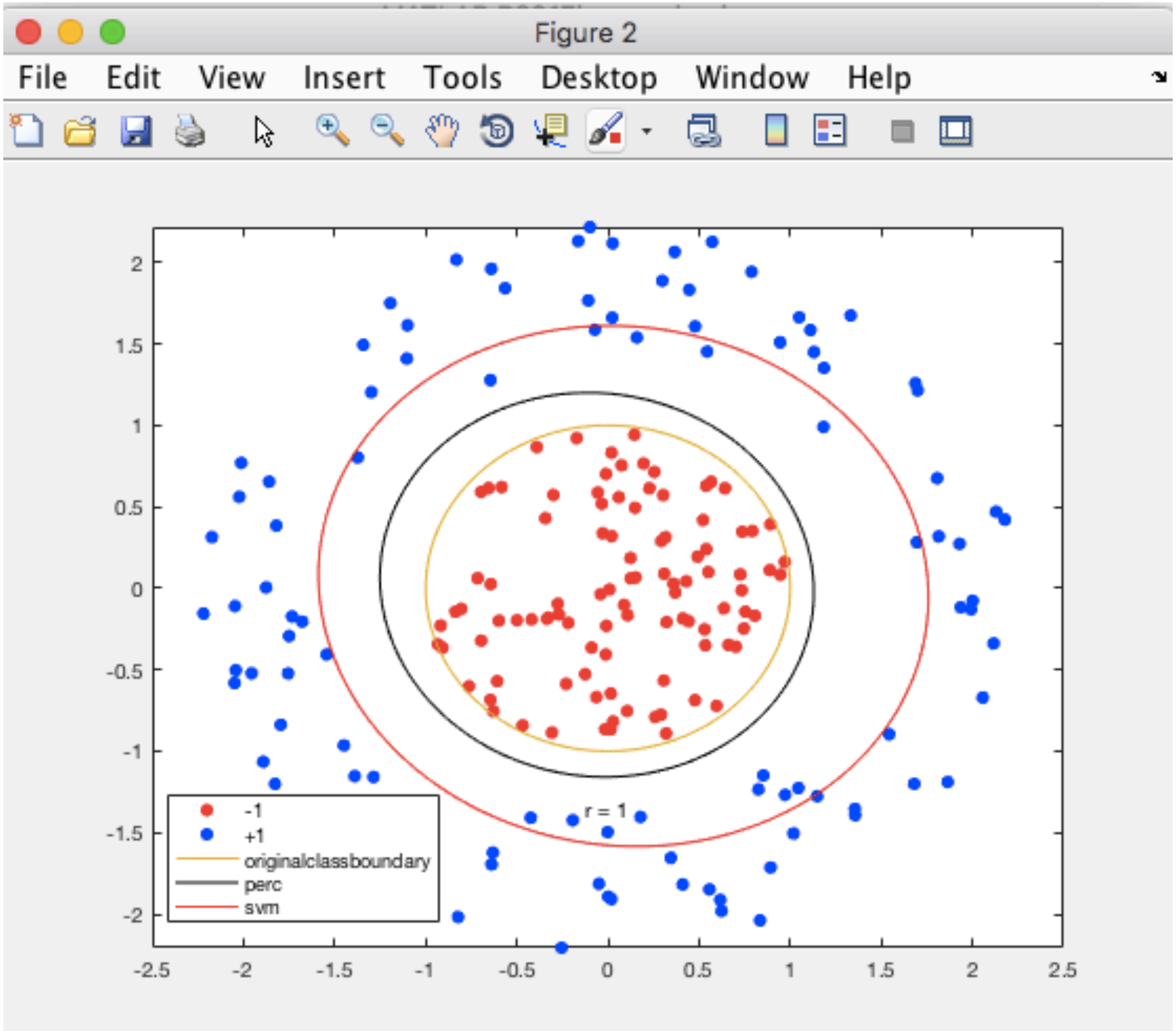
Q2b.

We now train the data with the svm model using the same polynomial kernel function and plot the decision boundary. On doing so we get the following.



Here, the red line signifies the svm model's decision boundary and the black line signifies the kernel perceptron model's decision boundary. The svm model aims to maximize the margin of the decision boundary while reducing error unlike the kernel perceptron that only works to minimize the loss function(i.e hinge loss). This causes the svm's boundary to be larger than the kernel perceptron's.

When we reduce the boxconstraint value, which affects the penalty induced when misclassifying points, the algorithm allows for more mistakes in classification while ensuring an increased margin in defining the decision boundary. When we get the box constraint value to 0.0001 we get the following plot.



As we can see, the model is okay with misclassifying some of the blue points in lieu of maximizing the margin between the observed data and the decision boundary.

Q2c

When running the kernel perceptron algorithm on the two datasets(train and test) we get the following error rates.

```
>> Q2
train error rate for 79 data is 0.354610
The test error rate for 79 is 1.773050
The error rate for 49 data training is 49.645390
The error rate for 49 data testing is 3.169014
```