Part1

1. Linear SVM

How to **run**: python svm.py linear

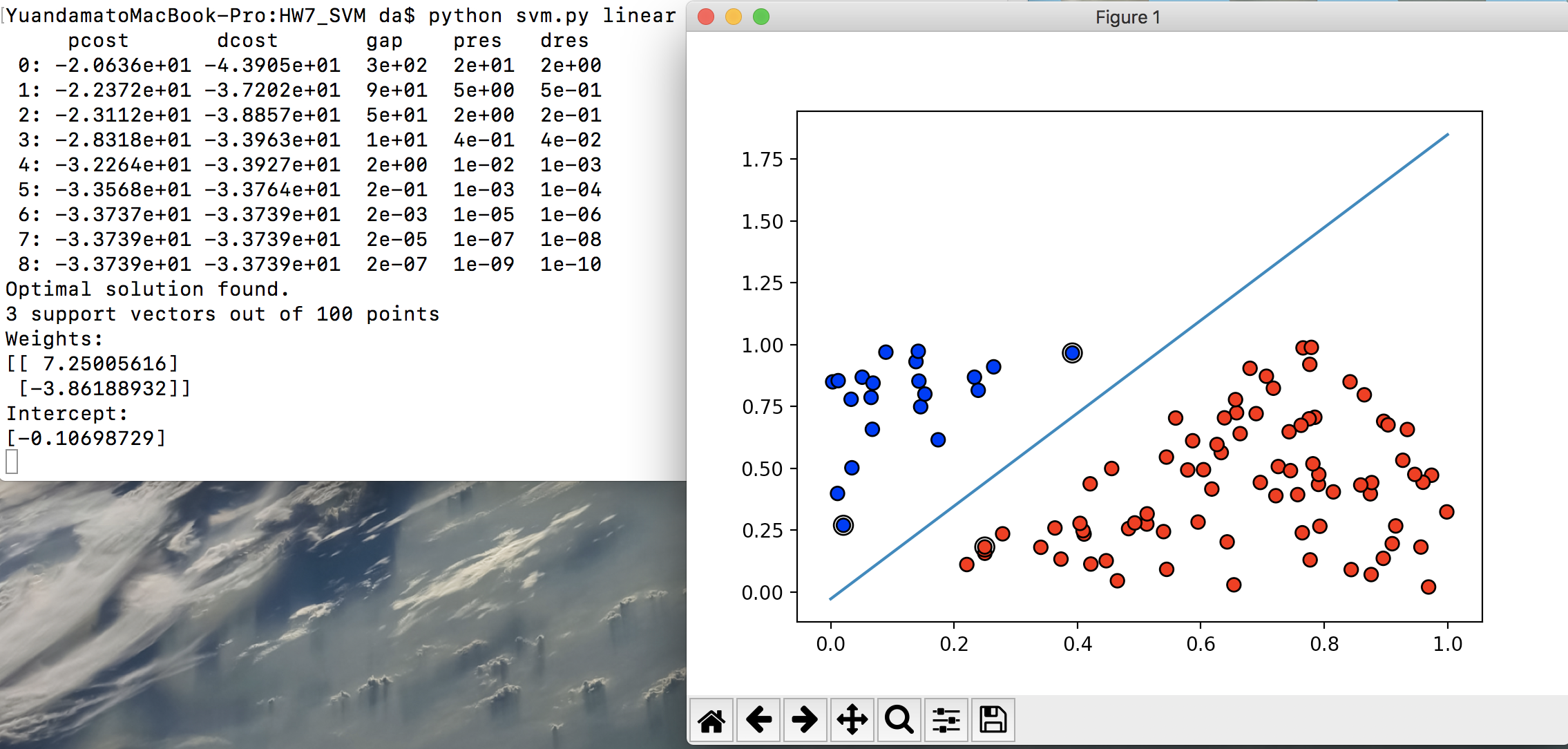
the python script will accept a parameter (linear | nonlinear) and will also use linsep.txt to train the data.

The equation is

Y = 7.25005616 \* X0 + -3.86188932 \* X1 + -0.10698729

1. Output

It will output the weights, intercept and the picture of the line



1. Data structures

In this programming, we used array, numpy.arrray, cvxopt.matrix

1. Code-level optimizations

* We used cvxopt to resolve the Quadratic Programming and used np.diag to represent G easily
* Substituted for loop with np.multiply and np.dot
* Used class to combine linear and nonlinear coding together

1. Challenges

* Learned how to use the cvxopt and convert the svm objective function and constraints
* Learned how to draw the picture with matplotlib

1. Nonlinear

How to **run**: python svm.py nonlinear

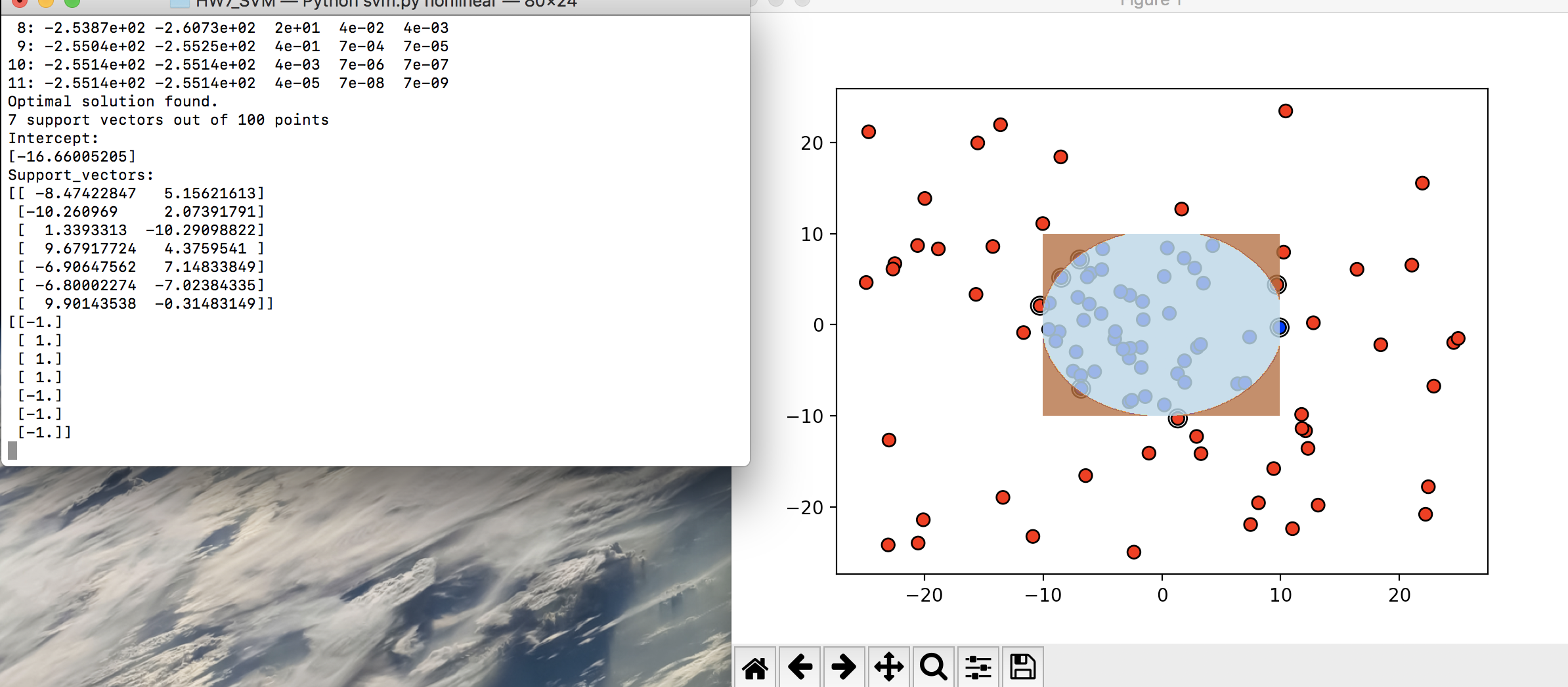
This will use the nonlinsep.txt

We used the **polynomial kernel**:

(1 + self.omiga \* np.dot(self.X[i], self.X[j])) \*\* 2

1. Output

It will output the intercept, the support vectors and the picture.



1. Data structure

array, numpy.arrray, cvxopt.matrix

1. Code-level optimizations

* Used kernel matrix to combine the linear and nonlinear, linear is just a special case with kernel function = np.dot(xi, xj)
* Changed the Q, Weights and Intercept
* Draw the picture with using np.meshgrid

1. Challenges

Draw the circle picture is not as easy as linear one