## PROGRAM DESCRIPTION

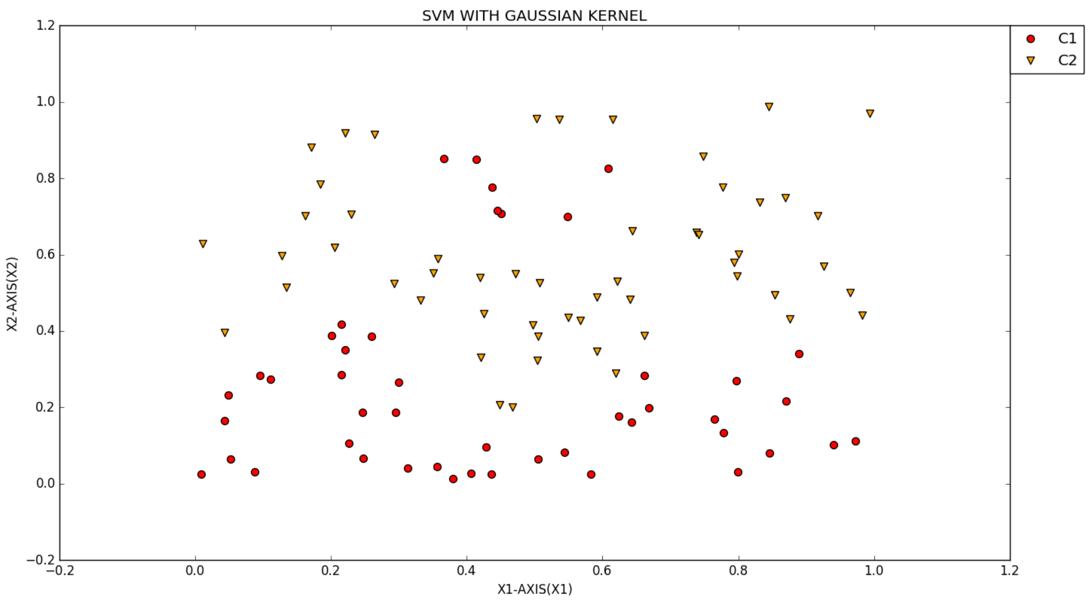
In this computer experiment we will apply SVM on a non_linearly separable dataset using the kernel trick.

We start creating 100 points x1….x100 independently and uniformly at random on $[0,1]^2$ . These 100 point,represented by 2 components x1 and x2 will be our input pattern.

We can now calculate the labels for each point xi using the following formula:

$$d_i = \begin{cases} 1, & x_{i2} < \frac{1}{5}\sin(10x_{i1}) + 0.3 \text{ or } (x_{i2} - 0.8)^2 + (x_{i1} - 0.5)^2 < 0.15^2 \\ -1, & \text{otherwise} \end{cases}$$

In this way we obtain two classes C1, that contains the points whose output is 1 and C2 that contains the points whose output -1.
If we plot the points we can see the distribution of the 2 classes:




Now we pick an appropriate kernel, in particular a **gaussian kernel**, that is characterized by the following formula:

$$k(x, y) = \exp\left( -\frac{\|x - y\|^2}{2\sigma^2} \right)$$

We apply the gaussian kernel to our dataset and in this way we obtain the matrix K, that is our program has the following shape: K(100x100).

Now we need to design an SVM that can separate the 2 classes C1 and C2. In order to accomplish this task what we will solve the quadratic function using an external library called **CVXOPT.** The primal SVM objective that is the following:

$$\min \tfrac{1}{2} ||w||2$$

$$\text{s.t. } y_i(w^T x_i+b)\geq 1 \ \forall i$$

And the corresponding dual problem:

$$\min \tfrac{1}{2} \alpha^T K\alpha - 1^T \alpha$$

$$\text{s.t. } \alpha_i \geq 0 \ \forall i$$

$$\text{and } y^T\alpha = 0$$

The CVXOPT library solve the following problem :

$$\min \tfrac{1}{2} x^T Px - q^Tx$$

$$\text{s.t. } Gx \preceq h$$

$$\text{and } Ax=b$$

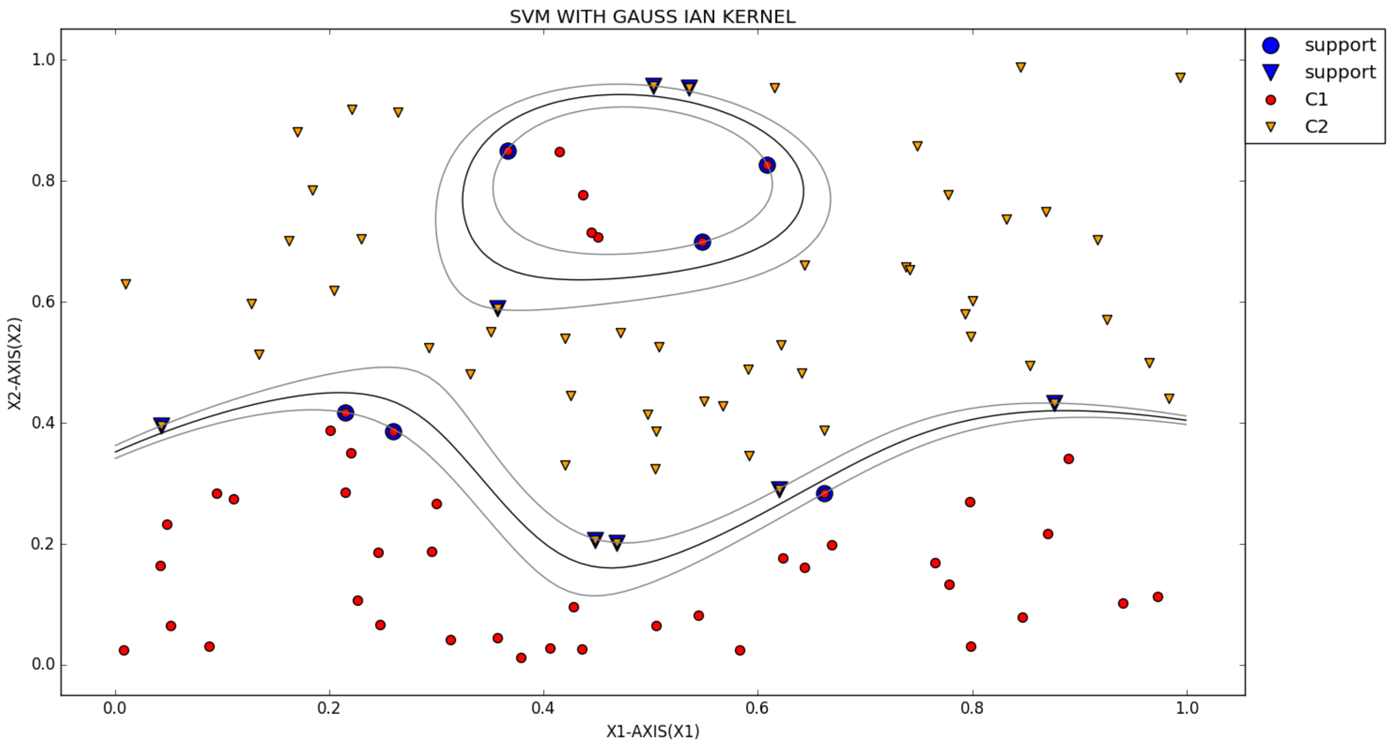
Now we will map the SVM formulation to the **CVXOPT.** In particular P is our K matrix, q will be a vector of ones, G will be a matrix with -1 on its diagonal, h is a vector of 0 , A is the label y transpose, and b is 0.
After writing the formulation CVXOPT will solve this function and will return an optimal solution.
The optimal solution returned will allow us to find the support vectors that is this case (using a random seed 2 ) will be 14.

Using the values of alpha we can now apply the discriminant function that has the following form:

$$g(\mathbf{x}) = \sum_{i=1}^{I_s} \alpha_i d_i K(\mathbf{x}_i, \mathbf{x}) + \theta$$

Where theta is the optimal bias.

Using this discriminant function we can now plot the svm decision boundaries using the contour plot provided by the "pylab" library of python.
The result is the following:




In this picture we can see that the points with a blue background represent our support vectors.
The final output of the code is the following:

/usr/bin/python2.7 "/home/marco/Scrivania/Homework/Neural Networks/HW8/HW8.py"

**pcost dcost gap pres dres**
**0: -6.2059e+01 -1.8593e+02 1e+02 7e-15 3e+00**

```
 1: -1.1430e+02 -1.4309e+02  3e+01  5e-15  1e+00
 2: -3.0131e+02 -3.5169e+02  5e+01  1e-14  1e+00
 3: -1.2357e+03 -1.3606e+03  1e+02  5e-14  1e+00
 4: -5.4064e+03 -5.7884e+03  4e+02  4e-13  1e+00
 5: -1.9544e+04 -2.0535e+04  1e+03  7e-13  1e+00
 6: -7.3117e+04 -7.7228e+04  4e+03  5e-12  1e+00
 7: -1.1285e+05 -1.1957e+05  7e+03  1e-11  1e+00
 8: -2.6624e+05 -2.8853e+05  2e+04  7e-11  1e+00
 9: -5.8552e+05 -6.6362e+05  8e+04  5e-11  1e+00
10: -1.2048e+06 -1.4561e+06  3e+05  2e-10  8e-01
11: -1.9010e+06 -2.3154e+06  4e+05  2e-10  5e-01
12: -2.1182e+06 -2.2329e+06  1e+05  3e-10  7e-02
13: -2.1238e+06 -2.1264e+06  3e+03  4e-10  1e-03
14: -2.1240e+06 -2.1240e+06  3e+01  7e-10  2e-05
15: -2.1240e+06 -2.1240e+06  3e-01  1e-09  2e-07
16: -2.1240e+06 -2.1240e+06  3e-03  5e-10  2e-09
Optimal solution found.

14 support vectors out of 100

The number of correct prediction is 100 over 100
Accuracy : 100.0%

Process finished with exit code 0
```

As we can see the training accuracy is 100% so there are no missclassified points as requested by the assignment and the number of support vectors is 14.