

ECE/CS 559 Neural Networks, Fall 2018 - Homework #7

Due: 11/15/2018, the end of class.

Erdem Koyuncu

All the notes in the beginning of Homework #1 apply.

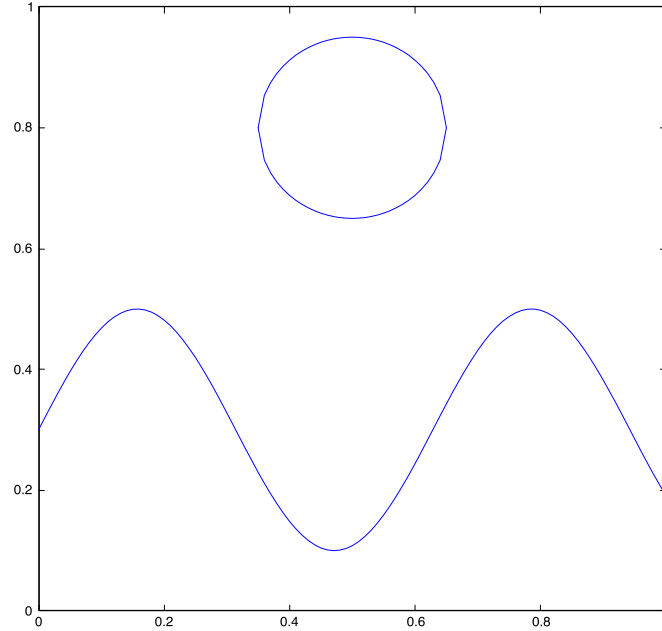
1. **(100pts)** In this computer project, we will design an SVM. You may use an existing library for solving the quadratic optimization problem that is associated with the SVM. For example, the free software Octave has such a command named “quadprog” to solve such problems. Other than that, you cannot use any existing machine learning/SVM library. As usual, please include the computer codes in your report.

(a) Draw 100 points $\mathbf{x}_1, \dots, \mathbf{x}_{100}$ independently and uniformly at random on $[0, 1]^2$. These will be our input patterns.

(b) Given $\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}$, let

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

denote the desired classes corresponding to the input patterns. Let $C_1 = \{\mathbf{x}_i : d_i = 1\}$ and $C_{-1} = \{\mathbf{x}_i : d_i = -1\}$. The region where d_i is 1 is the union of the region that remains below the mountains and the region that remains inside the sun in the figure below. Sketch the points \mathbf{x}_i , indicate (with different colors or markers) their desired class. **Make sure your samples include at least one point inside the sun.**



- (c) Pick an appropriate kernel (write it down in your report) and design an SVM that separates the classes C_1 and C_{-1} . Recall that the result of the SVM will be a discriminant function $g(\mathbf{x}) = \sum_{i=1}^{\mathcal{I}_s} \alpha_i d_i K(\mathbf{x}_i, \mathbf{x}) + \theta$, where α_i are some positive constants, \mathcal{I}_s is the set of the indices of support vectors, and θ is the optimal bias. Provide a rough sketch of the decision boundaries

$$\begin{aligned} \mathcal{H} &\triangleq \{\mathbf{x} : g(\mathbf{x}) = 0\} \\ \mathcal{H}^+ &\triangleq \{\mathbf{x} : g(\mathbf{x}) = 1\} \\ \mathcal{H}^- &\triangleq \{\mathbf{x} : g(\mathbf{x}) = -1\}. \end{aligned}$$

together with the support vectors. Note that the support vectors will be either on \mathcal{H}^+ or \mathcal{H}^- . **Your SVM should be able to perfectly separate the two classes with no exceptions of misclassified input patterns.**

Example solution: What I am expecting is a figure like this. The red crosses are input patterns with desired class 1, and the black diamonds are the input patterns with desired class -1 . The decision boundary \mathcal{H} (the blue line) can perfectly separate the two classes that were otherwise not linearly separable. There are 7 support vectors (marked by black circles) on \mathcal{H}^+ (the black line), and 6 support vectors (marked by red circles) in \mathcal{H}^- (the red line) for a total of 13 support vectors. As expected, there are no patterns in between the “guard lines” \mathcal{H}^+ and \mathcal{H}^- . Of course, the Kernel that I used is a secret.

Note that plotting the decision boundary $\mathcal{H} \triangleq \{\mathbf{x} : g(\mathbf{x}) = 0\}$ (or plotting \mathcal{H}^+ or \mathcal{H}^-) as a nice perfect line is quite a difficult task. What you can do instead is to approximate it as:

```
for  $x_1 = 0, 0.001, 0.002, \dots, 0.999, 1$ 
```

```
  for  $x_2 = 0, 0.001, 0.002, \dots, 0.999, 1$ 
```

```
    if  $g(\mathbf{x})$  is close enough to 0, then put a point in coordinates  $(x_1, x_2)$ .
```

```
  end
```

```
end
```

That is more-or-less what I did in my own figure (That is why the decision boundaries are not perfect and there are some “gaps” in between, you can even see the points that form the lines when you zoom in the PDF). Of course, you can use better techniques for generating the boundary curves if you like.

