

## ASSIGNMENT 7, due Wed, April 25

*Discussions are encouraged but no written records can be taken away from a discussion. Books and notes can be consulted but not copied from. Homeworks are due in class or in my mail box by 4pm on the due date. For experimental problems, debugging help may be obtained but all code should be your own and all results reported should be from running your own code. In order to verify “curious” results, we may request you submit your code so DO NOT delete your code after you hand in a problem set.*

### 1 (200pts) RBF’s and Neural Networks

In class we discussed how a radial basis function network in 1 dimension can be represented as a 2 layer neural network with a certain activation function and with some of the first layer weights being the  $\mu_i$ ’s, the positions of the centers and the second layer weights being the  $w_i$ ’s. Show (by providing a neural network implementation) that a RBF in 2 dimensions with two centers ( $\mu_1$  and  $\mu_2$ ) can also be represented as a neural network. In other words, provide a neural network implementation of the function

$$g(\mathbf{x}) = w_1 e^{-\frac{\|\mathbf{x}-\mu_1\|^2}{2\sigma^2}} + w_2 e^{-\frac{\|\mathbf{x}-\mu_2\|^2}{2\sigma^2}}$$

In general, there is a neural network implementation for an RBF in d-dimensions with M centers. This does not mean that we should therefore forget about RBF’s as that learning model is always implementable with a neural network. Each has their own uses, as well as their own learning algorithms.

### 2 (500pts) Gaussian Processes

We will work with the 1-dimensional data set given by

$$D = \{(1, 1), (1.25, 1.25), (1.5, 2), (2, 1.5)\}$$

- (a) For the Gaussian process that simulates a RBF, the covariance function is given by

$$C(x_i, x_j) = \sigma^2 e^{-\frac{(x_i - x_j)^2}{2r^2}} \quad (1)$$

Using this covariance function and the algorithm for using a Gaussian process to predict the mean, predict the value of  $y(x)$  for  $x$  in the interval  $[0, 3]$ , using the parameters  $\sigma = 0.5$  and  $r = 0.1, 0.5, 1$ . Show these three curves on the same plot along with the data points themselves. Briefly explain the results you observe.

- (b) For the Gaussian process that simulates a linear function, the covariance function is given by

$$C(x_i, x_j) = \sigma_w^2 x_i x_j + \sigma_\epsilon^2 \delta_{ij} \quad (2)$$

where  $\delta_{ij} = 1$  if  $i = j$  and zero otherwise. Using this covariance function and the algorithm for using a Gaussian process to predict the mean, predict the value of  $y(x)$  for  $x$  in the interval  $[0, 3]$  using  $\sigma_w = 0.5$  and  $\sigma_\epsilon = 0.5$ . Plot this curve along with the data points on the same plot. It should be clear why this covariance function is simulating a linear model.

### 3 (300pts) Support Vector Machines

For this problem, we will use the data set consisting of the 2 points  $\mathbf{x}_1 = (1,0)$ ,  $y_1 = +1$  and  $\mathbf{x}_2 = (-1,0)$ ,  $y_2 = -1$ .

- (a) An optimal separating hyperplane maximizes the minimum distance from the separating hyperplane to the data. In other words, an optimal separating hyperplane satisfies 2 conditions. First, it separates the data. Second, the closest data point (+ve or -ve) is as far as it can possibly be. Show that for a data set consisting of 2-data points, this optimal separating hyperplane is just the plane that is the perpendicular bisector of the line segment joining the two points. In our case, what is the equation of the optimal hyperplane (for these 2 data points).
- (b) Now consider a transformation to a more “complicated”  $Z$ -space. The transformation is given by

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1^3 - x_2 \\ x_1 x_2 \end{bmatrix} \quad (3)$$

- i. What are the data points in this space?
- ii. Construct the optimal hyperplane in this space (i.e., what is the equation for the hyperplane in terms of the  $Z$ -coordinates).

To classify a point in  $X$ -space, one first transforms the point into  $Z$ -space. One then classifies the point in  $Z$ -space using the optimal hyperplane in  $Z$ -space.

- (c) Plot (in  $X$ -space) the decision boundary for the optimal hyperplane constructed using the data in  $X$ -space (from part (a)). On the same plot, plot the decision boundary you would observe in  $X$ -space if you classified  $X$ -space points by first transforming to  $Z$ -space, and then classifying according to the optimal hyperplane constructed using the data in  $Z$ -space (this decision boundary will not be a line!).
- (d) A kernel function,  $K(\mathbf{x}, \mathbf{y})$ , is a function of two *vectors* in  $X$ -space defined by  $K(\mathbf{x}, \mathbf{y}) = \mathbf{z}(\mathbf{x}) \cdot \mathbf{z}(\mathbf{y})$ , where  $\mathbf{z}(\mathbf{x})$  and  $\mathbf{z}(\mathbf{y})$  are the transformed  $\mathbf{x}$  and  $\mathbf{y}$  into  $Z$ -space. In other words, the kernel function computes the dot product of the transformed vectors. Give an expression for the kernel function in terms of the components of  $\mathbf{x}$  and  $\mathbf{y}$ .

Support vector machines “magically” accomplish part (c) of this problem, with larger data sets, and much more complicated  $Z$ -spaces, and the amazing thing is that they perform their learning (construction of the optimal hyperplane in  $Z$ -space) and classification (transformation to the  $Z$ -space and then classification using the optimal  $Z$ -hyperplane) without ever having to explicitly visit the  $Z$ -space (as we had to here) but rather they use the kernel function  $K(\mathbf{x}, \mathbf{y})$  in the  $X$ -space alone.