

## ASSIGNMENT 6, due Thurs., April 19

*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 (300pts) Regularization

In class we discussed a smoothness regularization term that manifested itself as weight decay, which corresponded to adding a penalty term to the error function of the form  $\sum_i w_i^2$ , where the sum is over all the weights. This term encouraged the weights to be small, which corresponded to smoother functions. In this problem you are asked to design such penalty terms to enforce other types of constraints / prior information.

- Binary weights: Construct a term to enforce that the weights are close to  $\pm 1$ .
- Similar weights: Construct a term to enforce the weights to be close to each other.
- Weight sharing: Construct a term to enforce the weights to cluster into at most  $K$  groups where the weights in each group are as close to each other as possible. *Hint: You may want to define new parameters in addition to the weights.*
- Why are such terms called regularization terms? Do all the terms encourage smoothness?
- Symmetry: If you know that the target function is symmetric ( $f(-\mathbf{x}) = f(\mathbf{x})$ ), you might want to enforce that the weights represent a function that is symmetric by adding a penalty term. Generalize your penalty term to an arbitrary invariance  $\hat{L}$ . In other words, it is known that for some transformation  $\hat{L}$ ,  $f(\hat{L}\mathbf{x}) = f(\mathbf{x})$ , where  $\hat{L}\mathbf{x}$  represents the transformed  $\mathbf{x}$ . An example of such an invariance would be rotational invariance where (say) the target function for recognizing digits is invariant under rotations.
- If you multiplied such terms by a regularization parameter  $\lambda$  before adding it to the error function, how could one determine a good value for  $\lambda$  in practice?

### 2 (350pts) Nearest Neighbor Classifiers

Consider the following data set containing 7 points, generated by using a target function  $f : \mathbf{R} \mapsto \{-1, +1\}$ .

$$\begin{aligned} \mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, y_1 = -1 & \quad \mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, y_2 = -1 & \quad \mathbf{x}_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, y_3 = -1 & \quad \mathbf{x}_4 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, y_4 = +1 \\ \mathbf{x}_5 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, y_5 = +1 & \quad \mathbf{x}_6 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}, y_6 = +1 & \quad \mathbf{x}_7 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, y_7 = +1 \end{aligned}$$

- Plot the decision boundary (the boundary between the +1 and -1 regions) of the 1-nearest neighbor rule.
- Let the mean of all the -1 points be  $\mu_{-1}$  and the mean of all the +1 points be  $\mu_{+1}$ . Now imagine that the data set were condensed into the two points  $\{(\mu_{-1}, -1), (\mu_{+1}, +1)\}$  and use the 1-nearest neighbor rule based on these two points only. Plot the decision boundary.
- How many points in the training set would be classified incorrectly if the rule in b) is used?

### 3 (350pts) Radial Basis Functions

Consider the training set of problem 2. We will use the exact interpolation version of radial basis functions, with unit variance Gaussian kernels.

$$K(\mathbf{x}) = e^{-\frac{\|\mathbf{x}\|^2}{2}} \quad (1)$$

Hence,

$$g(\mathbf{x}) = \text{sign} \left( \sum_{n=1}^7 w_n e^{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2}} \right) \quad (2)$$

- (a) Find the values of the  $w_n$ 's numerically.
- (b) Plot the resulting decision boundary.