Rohan J. Dani

Machine Learning HW 4

1.

Design Problem Implement kernel estimation (Parzen windowing) and use it in a binary classifier. Place all your code in the file ps4p1.jl.

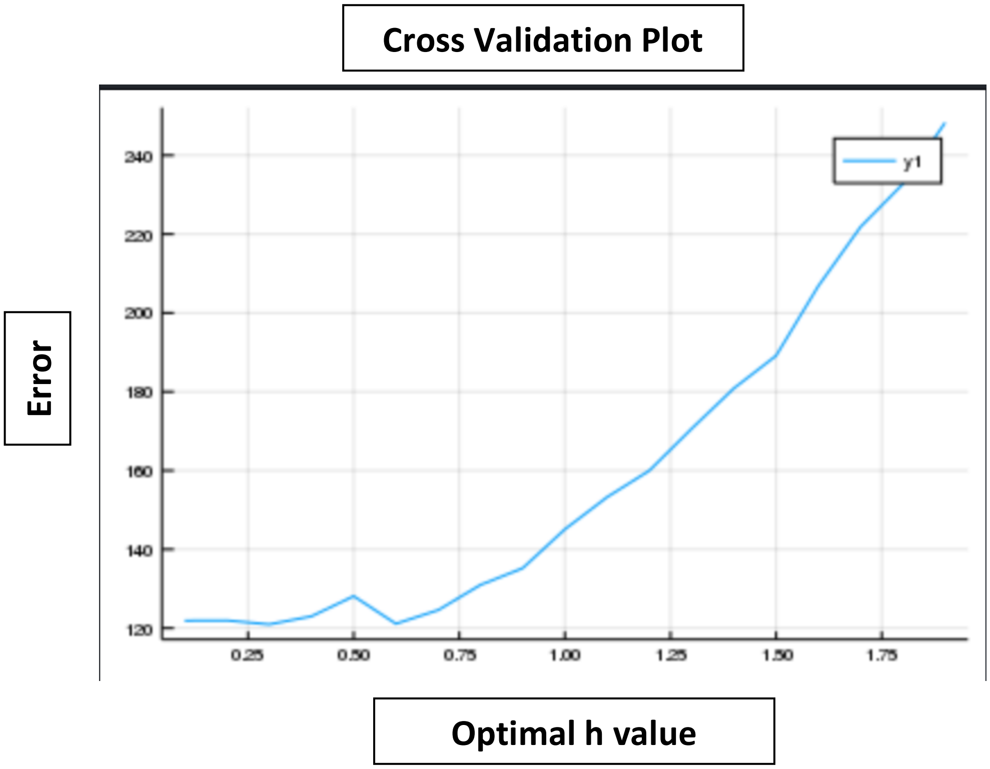
(a) Implement a julia function function kernel(x, data, h) To estimate p(x) at a point x (D-vector) given a data matrix (N x D) and a width h using a Gaussian kernel (window function).

(b) Using your kernel to estimate the class conditional densities, and a maximum-likelihood estimate of the class prior probabilities, implement a function to perform classification of new samples x, given the data and h, using a likelihood-ratio test (compare the ratio of the class conditionals to the ratio of the priors). function classify(x, data, h)

(c) Read the data file ps4p1.h5 containing an 1000 × 2 matrix (group name ”data”) representing a training set for a binary classification problem with a single real feature x. Perform leave-one-out cross validation to select the optimal value of h. Enter the optimal h found.

h = 0.30000000000000004

(d) Plot the cross-validation error as a function of h from part c.



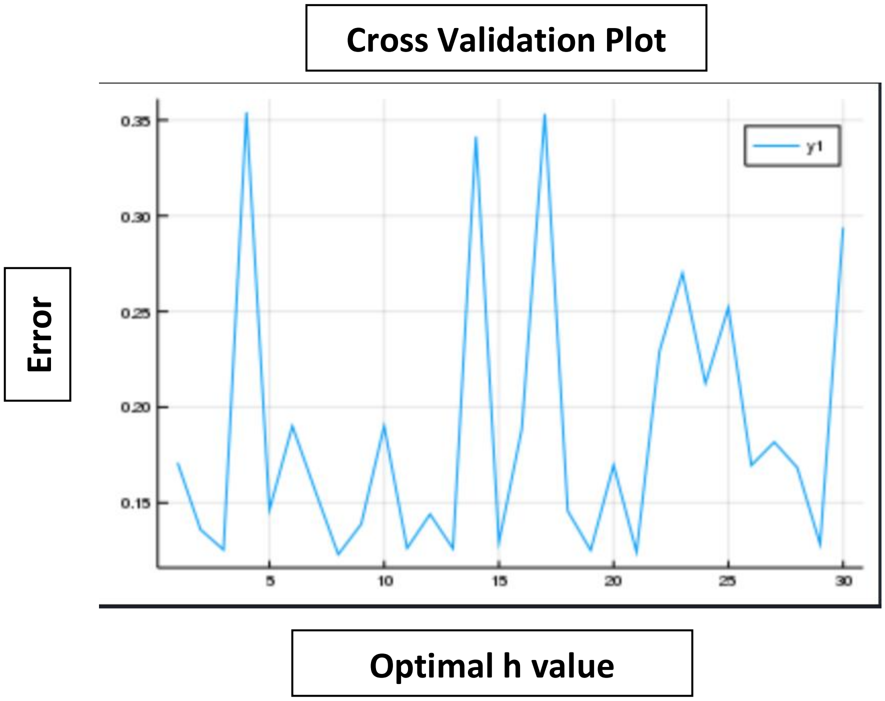
2. Design Problem k-Nearest-Neighbor classifier. Place all your code in the file ps4p2.jl.

(a) Implement k-Nearest-Neighbor classifier as a julia function function knn(x, data, k) where x are the new values to classify, data is the training data, and k is the number of neighbors to use.

(b) Read the data file ps4p1.h5 from problem 1. Perform leave-one-out cross validation to select the optimal value of k. Enter the optimal k found.

k = 8.0

(c) Plot the cross-validation error as a function of k from part c.


Cross Validation Plot
Error
y1
Optimal h value