Daniel Campos November 21st, 2013

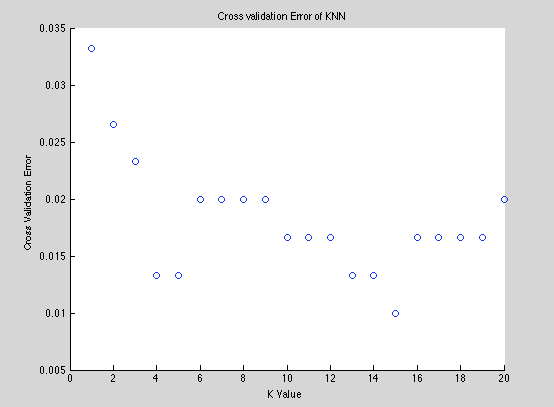
RCS: Campod2 RIN: 660996361

Machine Learning CS 4100

I worked on these problems with Zoe Konrad☺

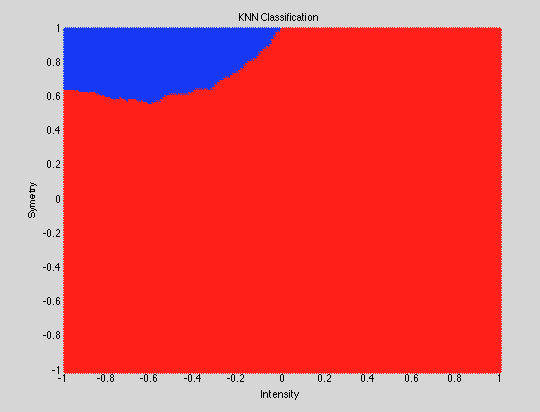
Problem Set 11

1. (450)
   1. Use cross validation with your training set to select the optimal value of k for the k-NN rule. Give a plot of Ecv versus k. What value of k do you choose?



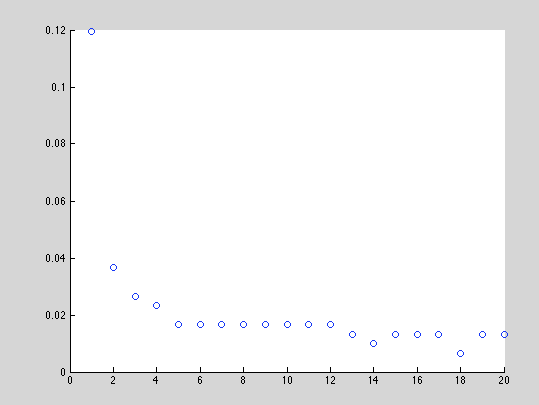
The value of k I choose was 15

* 1. For the value of k that you took, give a plot of the decision boundary.

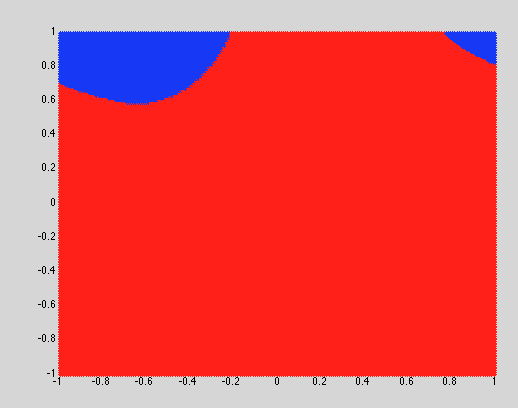
In-sample error=0.044

Cross validation error=.0155

* 1. What is the test error Etest=.00102

1. (450)
   1. 

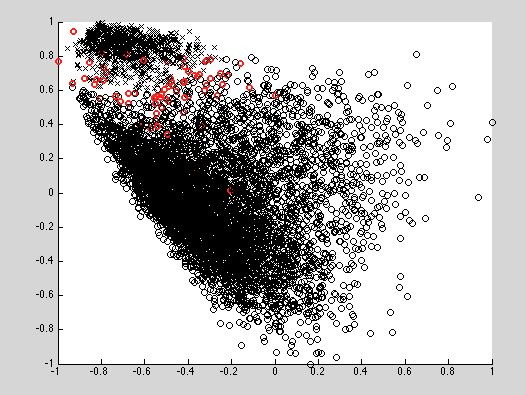
I chose k=17

* 1. For the value of k that you took, give a plot of the decision boundary.

In-sample error=.066

Cross validation error=0.0087

* 1. What is the test error Etest= 0.0103

All the miscategorized points are red, the rest black

1. (100)

Compare the ﬁnal test error from your three attempts to solve this problem:

(i)Linear model with 8th order polynomial transform and regularization selected by CV.

(ii) k-NN rule with k selected by CV.

(iii) RBF-network with number of centers selected by CV.

i)I had an etest of .01345

ii)Etest .0155

iii)Etest .0103

Both my knn and my rbf functions had bareley smaller Eouts that my 8th order transforms but they took much longer to run and used more memory. I was hoping that my linear model would present a better error because really there are only two clusters and thus thought that a linear model would work very well. With that the error for all three is so miniscule. That being said, on average the knn produced much more consistent results than my RBF function but that just might be because I had to write more of my own functions for the rbf. The rbf also produced a substantially lower cross validation error

RBF uses a lot more memory and takes and takes way longer to run. It has very high accuracy but when there are close to ambiguous outputs it messes up. It is much better when dealing with a lot of different categories. The Knn classifier uses a prohibitively large amount of memory and is much slower at classification. That being said it is still pretty good for classification. That being said the linear is really good for just 1 vs not 1