## ASSIGNMENT 7

LFD is the class textbook

1. (500) Classifying Handwritten Digits: 1 vs. 5

Pick one of the following 3 classification algorithms for non-separable data:

- (i) Linear Regression for classification followed by pocket for improvement.
- (ii) Linear Programming for classification.
- (iii) Logistic regression for classification using gradient descent.

Use your chosen algorithm to find the best separator you can using the training data only (use your 2 features from a previous assignment as the inputs). The output is +1 if the example is a 1 and -1 for a 5.

- (a) Give separate plots of the training and test data, together with the separators.
- (b) Compute  $E_{\rm in}$  on your training data and  $E_{\rm test}$ , the test error on the test data.
- (c) Obtain a bound on the true out-of-sample error. You should get two bounds, one based on  $E_{\rm in}$  and one based on  $E_{\rm test}$ . Use a tolerance  $\delta = 0.05$ . Which is the better bound?
- (d) Now repeat using a 3rd order polynomial transform.
- (e) As your final deliverable to a customer, would you use the linear model with or without the 3rd order polynomial transform? Explain.
- 2. (200) Gradient Descent on a "Simple" Function

Consider the function  $f(x,y) = x^2 + 2y^2 + 2\sin(2\pi x)\sin(2\pi y)$ .

- (a) Implement gradient descent to minimize this function. Let the initial values be  $x_0 = 0.1$ ;  $y_0 = 0.1$ , let the learning rate be  $\eta = 0.01$  and let the number of iterations be 50; Give a plot of the how the function value drops with the number of iterations performed.
  - Repeat this problem for a learning rate of  $\eta = 0.1$ . What happened?
- (b) Obtain the "minimum" value and the location of the minimum you get for gradient descent using the same  $\eta$  and number of iterations as in part (a), starting from the following initial points: (0.1, 0.1), (1, 1), (-0.5, -0.5), (-1, -1). A table with the location of the minimum and the minimum values will suffice. You should now appreciate why finding the "true" global minimum of an arbitrary function is a hard problem.
- 3. (300) Problem 3.16 in LFD