1. [25 points] L1 vs. L2 Regularization

In this problem, we explore the differences between L1 and L2 regularization, discussed in lecture 2. You are given two data \_les, hw2x.dat and hw2y.dat (containing inputs and outputs, respectively).

(a) [10 points] Split the data into a training set, containing 90% of the instances, and

a test set, containing 10% of the instances (since this is exploratory, we will not

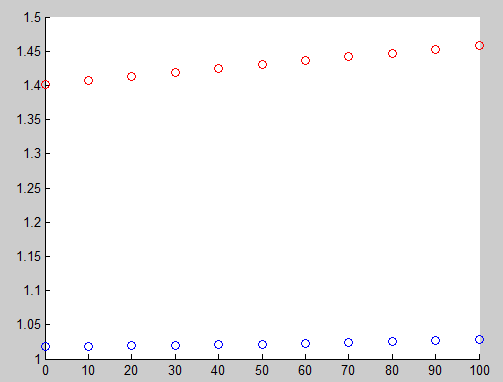
do full cross-validation). Write Matlab code to perform L2 regularization. Plot

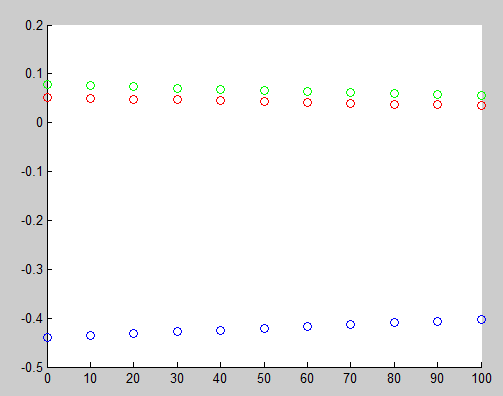
on one graph the root mean squared error on the training set, and the root mean

squared error on the test set, as a function of the regularization parameter .

Vary \_ starting at 0, and go high enough that you can see an \interesting" range

of behavior. Plot, on a different graph, all of the weights, as a function of





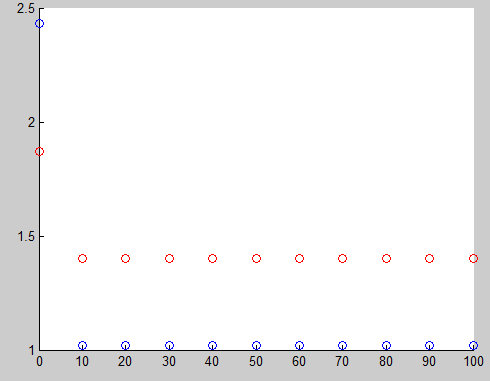
(b) [10 points] Using the quadprog function of Matlab, write a function that performs

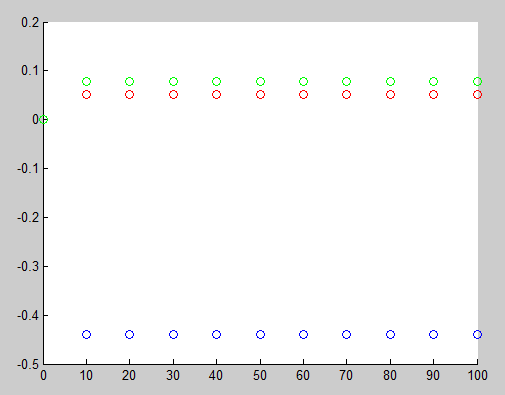
L1 regularization

see the attached Q1.m

(c) [5 points] Plot the same graphs as above for the L1 regularization. Explain what

you observe, and comment on how you think the data was generated.





2. [10 points] Dealing with missing data

Suppose that you use a Gaussian discriminant classifier, in which you model explicitly P(y = 1) (using a binomial) and P(x|y = 0) and P(x|y = 1). The latter have

distinct means and , and a shared covariance matrix(a frequent assumption

in practice). Suppose that you are asked to classify an example for which you know

inputs x1; : : : xn􀀀1, but the value of xn is missing. In practice, a common approach

in this case is to fill in" the value of xn by its class-conditional means, E(xn|y = 0)

and E(xn|y = 1). Using the log-odds ratio, give a mathematical justification for this

approach.

In order to calculate the posterior P(y|x), we apply the Bayes theorem:

it can be written into logistic sigmoid function if put into log-odd ratio form:

where

Since the latter have distinct means and , and a shared covariance matrix

We have the multivariate Gaussian form for P(x|y = 0) and P(x|y = 1)

plug in to

can be rewrite to

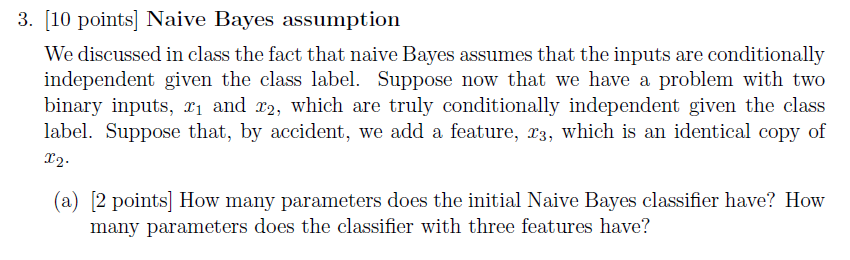
denote

This question is under assumption that distinct means and are not dependent on the missing data. Therefore, and will not change whether we have or not. And shared covariance matrix(a frequent assumption in practice), will not change.

If we only look at the last term, where value of xn by its class-conditional means, E(xn|y = 0) and E(xn|y = 1)

Under assumption E(xn|y = 0) and E(xn|y = 1) will be even. The last term will cancel itself in

Therefore, for missing data, we can use the class-conditional means to replace.



a) The parameters of the native Bayes are:

Hence for two binary input x1, x2, parameters are

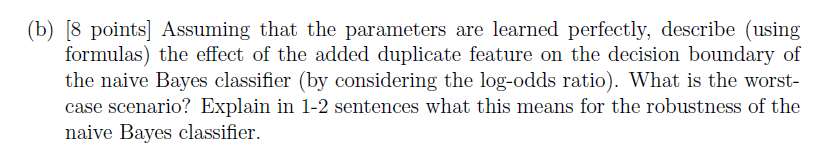
,

There are 5 parameters.

For two input x1,x2 and duplicate x3, parameters are:

,

There are 7 parameters.



b)

due to x2, x3 are identical and

Class label has binomial distribution and class conditional distributions are multivariate Bernoulli

The decision surface:

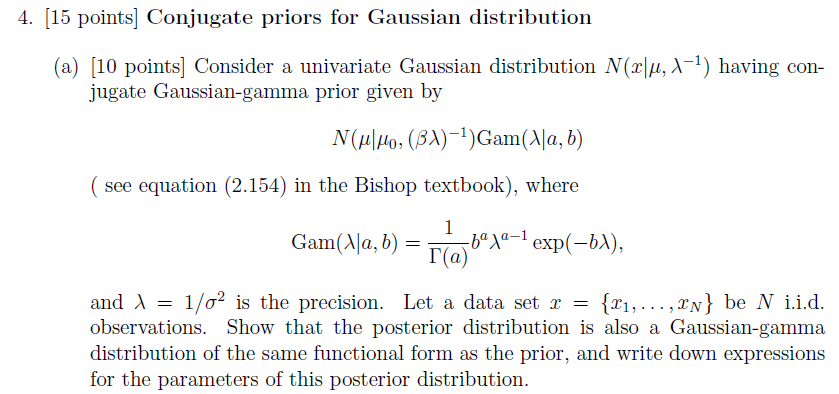
Use the lock trick, and x3=x2 we get:

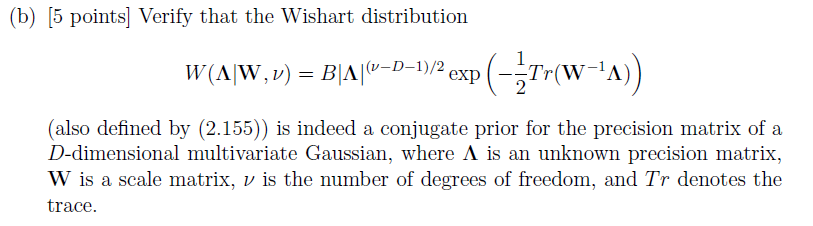
denote

This is still linear decision boundary. We have additional term

Or is the additional term added to decision surface, if when the distribution of feature is highly unbalanced, the additional term will impact on the decision surface. For example, the worst case is then reach infinity.

however, the x3 is a duplicate of x2. In this extreme case, even without adding the x3, such distribution is already unbalanced. Thus, in both way, the additional term will not change the linear behavior of decision boundary, and even in extreme case, the influence would be very small.





5. [40 points] Using discriminative vs. generative classifiers

For this problem, you will experiment with a version of the Wisconsin data set that we

use as an illustration in class. The data is available in files wpbcx.dat and wpbcy.dat.

(a) [10 points] Implement logistic regression. If you use a learning-rate version, you

will need to set up your code in such a way as to be able to search for a good learning rate. You can also use the iterative recursive least-squares version (whichever

you prefer).

I use the iterative recursive least-squares version.

Logistic function of linear combination input is

Apply Newton-Raphson for logistic regression:

The Hessian has the form:

where R is the diagonal matrix of (1-,

is the input file wpbcx.dat. we will assume that , to create a bias term.

Update weight using this formula:

(b) [10 points] In a first experiment, use just a bias term and the first feature (first

column in the wpbcx.dat file). Set up a Gaussian naive Bayes classifier, and compare its results with logistic regression, using 10-fold cross-validation. Comment

on what you observe.