Emily Schultz
(ess2183@columbia.edu)
COMS 4771: Machine Learning
Prof. Pe'er

Midterm Exam: Explanation

I implemented Problem 02: Predict integer output (`y_int`) given input (`X`) and real output (`y_real`).

I first began by trying to make some observations about the data. This was somewhat difficult since `D=9` in this case for `X`, and using `y_real` gives a total of 10 different facets to look at to predict one number (either 1 or 2). I did conclude that the `y_int` values of a data point are nearly always 2 if the first value of `X` (x_1) is negative. Also the 2 values do not exist at the extrema of `y_real`, but are only for more central values (between -109 and 132, rather than between -175 and 194). These are displayed in `observations.m`.

Next, based on the parameters of the problem, I knew this was a classification problem and a supervised learning problem. `Data.mat` provides the training data, but the testing data is what we will be graded on. Based on this information and what we've learned in class, I decided a SVM would be the best course of action.

All code related to training an SVM to find my final model is in `svm.m`. This file contains the function `svm()`, which takes at most one argument, the test number (between 1 and 8, inclusive). If no value (or an invalid value) is provided for `test`, then the default value of `test` (`test=8`) will be used, which is the best and final version. This function calculates an SVM model and saves it to `SVM.mat`. This is the file (and SVM model) that is used in `predict.m`. This function will also attempt to classify the inputs using the newly found SVM model and report the classification error (the percentage incorrectly classified).

I started out using a linear kernel function and only using the `X` as inputs to train the svm. To run this, use `svm(1)`. This yielded a classification error of 2.16%. Honestly, I didn't think that was really that bad at first.
Then I tried using a linear kernel function and using both `X` and `y_real` as inputs to train the SVM. To run this, use `svm(2)`. I figured it would be best to give the most information possible when training. It turned out this improved the classification slightly, now 2.13%.

At this point I decided to try non-linear kernel functions, just in case it improved my results. For each new kernel I tried, I tested it with both just the `X` as input and the `X` and `y_real`. Using a quadratic kernel function, I got classification errors of 0.38% and 0.23%, respectively, which was a drastic improvement from the linear kernel function. That's when I knew that polynomial was probably the right place to look for a kernel function. To run this, use `svm(3)` and `svm(4)`.

Next I tried a polynomial kernel function of degree 3. Once again, the classification error improved to 0.17% and 0.06% for `X` and `X` and `y_real` respectively. To run this, use `svm(5)` and `svm(6)`.

Finally, I used a polynomial of degree 4. Here, for both inputs I got a classification error of 0. Obviously this is the best I could ask for, and so I decided that my final SVM classifier would be the SVM with polynomial kernel function of degree 4 with both `X` and `y_real` as inputs, since that always had a higher classification rate than just `X`. To run with just X, use `svm(7)`, and to run the final (best version) use `svm(8)`.
**If you run another version, make sure to run `svm(8)` or just `svm()` before using `predict.m`.

Just to make sure, I tried using higher degree polynomials for my kernel function (ie 8 and 10), but these gave me high(er) classification errors, so I knew that degree 4 was best.

I hope that this is an accurate SVM for this distribution, since it worked so well on the training data, but it's always possible it's off and the testing data will not be as accurate.

To run/test this:

```
load('Data.mat');                          (to get the inputs for predict)
y_predicted = predict(X, y_real);          (to get the predicted y_int values)
```

I hereby state that I have neither given nor received help on this exam (beyond the course staff on Piazza) and that I have followed the guidelines regarding open book material and have not exceeded the 48 hour time limit.