Machine Learning Module

Week 1

Laboratory Exercise, Week 1

Linear Regression

Mark Girolami
girolami@dcs.gla.ac.uk
Department of Computing Science
University of Glasgow

January 8, 2006

1 Laboratory Exercise

In this laboratory you will gain experience of using Matlab and also explore some characteristics of the *Least Squares* linear models. We will use the *Long Jump* data which should be downloaded from the module website.

2 Matlab Tutorials

Matlab is a simple environment to use and write the simple scripts which will be required for this course. There are links to a number of online introductory Matlab tutorials from the class website and you are encouraged to use these if you require.

2.1 Loading the Data into Matlab

The data, which can be downloaded from the class website long_jump_data.txt, is arranged in two columns corresponding to time-elapsed and gold-distance within a text file.

Load the contents of the file into a Matlab matrix variable `x` using the command

```
x=load('long_jump_data.txt');
```

We will now want to make *train* and *test* data sets, so we will use the results from the first twenty games to obtain our *Least-Squares* models and employ these in predicting the winning long jump distance for the last five games. The following commands will achieve this.

```
x_train = x(1:20,1);
t_train = x(1:20,2);
x_test = x(21:end,1);
t_test = x(21:end,2);
```

Now that we have a *train* and *test* set a range of polynomial-order linear regression models should be fit to the *train* data and both the training (MSE) and the predictive error (test error) for each order of model should be measured. Use a range of $K = 0$ to $K = 4$ for this experiment.

1. How does the MSE (training error) change with the polynomial order of the model?

2. How does the quality of predictions, measured as MSE on test samples, vary with the polynomial order of the model?

3. What general model design principle emerges from your observations in this experiment?

The following Matlab code, downloadable from the website wk_1_matlab.m, will produce the required results for you to answer the above questions. Study the code so that you understand it.

What happens as you increase the range of K?

```
Polynomial_Order = 4;
X_train = [];
X_test = [];

for i = 0:Polynomial_Order
    X_train = [X_train x_train.^i];
    X_test = [X_test x_test.^i];

    w_hat = inv(X_train'*X_train)*X_train'*t_train;

    t_train_hat = X_train*w_hat;
    t_test_hat = X_test*w_hat;
    mse_train(i+1) = mean((t_train - t_train_hat).^2);
    mse_test(i+1) = mean((t_test - t_test_hat).^2);
end

subplot(2,2,1);
plot(0:Polynomial_Order,mse_train,'dg-');
title('Train Error');

subplot(2,2,2);
plot(0:Polynomial_Order,mse_test,'dr-');
title('Test Error');

[min_test_val,min_test_index] = min(mse_test);
```

```
[min_train_val,min_train_index] = min(mse_train);

t_train_hat_min = X_train(:,1:min_train_index)*...
inv(X_train(:,1:min_train_index)'*X_train(:,1:min_train_index))*...
X_train(:,1:min_train_index)'*t_train;

t_test_hat_min = X_train(:,1:min_test_index)*...
inv(X_train(:,1:min_test_index)'*X_train(:,1:min_test_index))*...
X_train(:,1:min_test_index)'*t_train;

subplot(2,2,3)
plot(x_train, t_train,'og');
hold on;
plot(x_train,t_train_hat_min);
title('Minimum Train Error Model');

subplot(2,2,4)
plot(x_train, t_train,'og');
hold on;
plot(x_train,t_test_hat_min);
title('Minimum Test Error Model');
```