Machine Learning Module

Week 2

Laboratory Exercise, Week 2

Generalisation & Overfitting

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 further experience of using Matlab and you will also use cross-validation to study what would be the best linear regression model for predicting the Olympic gold winning distances in the long jump.

1.1 Studying Train, Test & Cross Validated Errors

The Matlab code on the next page (downloadable from the website) provides a simple demonstration of how increasing model complexity provides a better fit to the available data sample and there comes a point where your function is actually modeling the noise in the data thus causing a drop in the generalisation performance.

Study the code seeking to fully understand it and then run the script watching how the estimated function varies with increasing model complexity.

1. Experiment with fourth & fifth order functions, (as well as different noise levels), of your own choosing, how does LOOCV perform in identifying the *true* model order for these more complex functions. Print out the true function, a data sample and the estimated function for a range of polynomials from K=1 to K=10 and in addition the overall train, test and LOOCV error curves. How well does LOOCV do in locating the optimal model order for your chosen functions?

2. Using the long-jump data from last week use LOOCV to identify the optimal model order when the results for the first 20 games are available for training with the remaining five games being used for testing. Are the results as clear cut as you would hope for? Discuss your findings.

```
clear Range = 10;
Nos_Samps = 50;
Nd = 100;
Max_Model_Order = 10;
noise_var = 150;
T=[]; Tt=[]; Tcv=[]; Tcvs=[];

x = Range*rand(Nos_Samps,1)-Range/2;
eta = noise_var*randn(size(x));

f = 5*x.^3  - x.^2 + x;

f_n = 5*x.^3  - x.^2 + x + eta;

xt = (-Range/2:0.01:Range/2)';
tt = 5*xt.^3 - xt.^2 + xt;

[i,j]=sort(x);

X=x.^0; Xt=xt.^0;
for k=1:Max_Model_Order
    X=[X x.^k];
    Xt=[Xt xt.^k];

    w_hat = inv(X'*X)*X'*f_n;
    f_hat = X*w_hat;
    f_test = Xt*w_hat;

    [cve, cvs] = cross_val(X, f_n);
    T  = [T; mean((f_n - f_hat).^2)];
    Tt = [Tt; mean((tt - f_test).^2)];
    Tcv = [Tcv; cve];
    Tcvs=[Tcvs;cvs];

    plot(i,f(j),'-');
    hold on
    plot(i,f_n(j),'.g')
    plot(i,f_hat(j),'-r')
    hold off
    pause(1)
end

figure
plot(1:Max_Model_Order,T,'dr--'); hold;
plot(1:Max_Model_Order,Tt,'og-');
errorbar(1:Max_Model_Order,Tcv,Tcvs/sqrt(Nos_Samps-1),'sk-');
```