Lab 5
CO902 – Probabilistic and statistical inference – 2012-13 Term 2
Lecturer: Tom Nichols

Supervised Learning Redux - "Prediction game"

Please split into two teams, "self-organize".

Each team must come up with a (secret) function $f: R^4 \rightarrow R$, which the other team must "learn". Each team will generate a dataset containing 100 pairs $\{X_i, Y_i\}$, where $Y_i = f(X_i)$ + "noise", which the other team will use to estimate the function f, creating f^{hat}. After the learning phase is over, the predictors – one from each team – are tested on hitherto unseen datasets of size n=1000 which the other team must provide. The goal is to have the smaller root mean squared error,

$$\sqrt{(\text{mean}(\{ f^{hat}(X_i) - Y_i)^2 \})}$$

on the held out dataset, i.e. to be able to predict hitherto unseen outputs from the unknown function. Of course, for testing the teams must exchange the noiseless functions, but no further "learning" is allowed.

Some ground rules (to prevent a race to the bottom):

- The X's are to be drawn uniformly from the unit hypercube (i.e. no concentrating on atypical regions of the domain), i.e.
  `Xtrain = rand(4,100);` and `Xtest = rand(4,1000);`
- Outputs, once produced, should be shifted and scaled to have zero mean, unit variance. Then Gaussian noise with standard deviation 0.1 is added additively, i.e.
  `+randn(4,100)/10` and `+randn(4,1000)/10;`
  Data should *once again* be re-scaled to have zero mean, unit variance (to ensure RMS errors are comparable).
- Please have your code produce corresponding "true" versions that have had no noise added, but that are identically standardized. In pseudo code…

```
Ytrue = f(X)
Ytrue = (Ytrue – mean(Ytrue))/std(Ytrue)
Yn    = Ytrue + noise;
Y     = (Yn    – mean(Yn))/std(Yn)
Ytrue = (Ytrue – mean(Yn))/std(Yn)
```

- Create two mat files, `Train.mat`, holding a length-100 `Y` vector and a 4 × 100 `X` matrix, and a `Test.mat`, holding two length-1000 vectors, `Y` and `Ytrue`, and a 4 × 1000 `X` matrix.
- Use any learning techniques you think appropriate – linear regression, general linear regression, cross-validation, test and train, ridge regression or others.

Note that the use of `Ytrue` is useful to get an idea of the minimal possible mean squared error. That is, due to noise there will always some error, but the best you can do is by using the true, noiseless function

$$\sqrt{(\text{mean}(\{ f(X_i) - Y_i)^2 \})}$$

where $f(X_i)$ corresponds to `Ytrue`. Looking at the difference between this MSE and the one above (with $f^{hat}(X_i)$) will give a sense of how far the prediction error is from the ideal minimum MSE.