Functional Data Analysis : week 2

August 2015

1 R tools

For fourier series, we write the basis objext as :

$$basisobj = create.fourier.basis(rangeval, nbasis, period)$$

and for B-spline,

$$basisobj = create.bspline.basis(rangeval, nbasis, norder, breaks)$$

We can create functional data object once we have a coefficient matrix.

$$tempfd = fd(coefmat, basisobj)$$

Coefficient matrix can be of three dimensions - first for no. of coefficients, second for no. of functions and the last for the dimension of variable (multivariate variable).
We can get the curves smooth by keeping the number of basis function small realtive to the amount of data being approximated. Taking value of K large, the curve may fit the training set very well (so the cost function would be almost zero), but at the same time will fail to generalize to new examples known as the problem of overfitting.
More about Spline : A spline is a numeric function that is piecewise-defined by polynomial functions, and which possesses a sufficiently high degree of smoothness at the places where the polynomial pieces connect (which are known as knots). Applications : In *Computer Graphics*, paramtric curves are given by splines because of its simplicity, ease and accuracy of evaluation.

2 Computing smooth curves from noisy data

2.1 Smoothing by Regression analysis

To make the curve fit to data, we minimize the sum of the squared errors or residuals as

$$SSE(x) = \sum_j (y_j - x(t_j))^2$$

where x is the basis function. Now the minimization problem is

$$SSE = \sum_j (y_j - \sum_k c_k f(t_j))^2 = \sum_j (y_j - F'(t_j)C)^2$$

This method is used when we get our work done by curves itself. To go into the derivatives, next approach is recommended

2.2 Smoothing by Roughness penalties

What to do if we recognize overfitting?
1. Reduce the value of K that appears in the summation of basis function
2. Regularization - Keep all those higher degree functions, but reduce the magnitude of their coefficients.How? We add a "penalty" for the them, and when we come to minimize the cost function it will take penalty into consideration and will choose small values of coefficients. c_j
Roughness of curve : Curvature is defined as the square of the second derivative of the curve at argument value t

$$[D^2x(t)]^2$$

a line has zero curvature. Roughness is integrated squared second derivative or total curvature

$$\int [D^2x(t)]^2 dt$$

When the function is highly variable, the squared of second derivative is large so as the penalty.Consider the equation

$$F(c) = \sum_j (y_j - F'(t_j)C)^2 + \lambda \int [D^2x(t)]^2 dt$$

The smoothing parameter λ is responsible for panalizing the curvature. When λ is sufficiently large, $[D^2x(t)]^2$ is zero resulting to a straight line - an example of underfitting where as when λ tends to zero, the function tries to fit the data as closely as possible resulting in overfitting. In this case the unbiased estimator of the regression coefficients become

$$c = (ff' + \lambda R)^{-1}$$

where R is the penalty matrix. To get the correct λ we get it generally by the choice of user, what level he wants.