Functional Data Analysis (AMS Spring 498B)

Homework 2

This assignment can be handed in as a hard copy or sent in electronically. You can work together in groups but submit the homework separately.

⋄ For the parts that involve computing submit a copy of your source code and output and/or plots. (You should use the notebook function in R studio or the publish function in Matlab to create a nice report.) Use the rMarkdown format in R studio to include written answers along with your analysis. Be thrifty in what you include in your output and avoid listing extraneous matrices and vectors.

⋄ All figures should be well-crafted and include labels for the axes and a title.

⋄ Ten (10) points are given for each separate item, so problems count for more than others.

⋄ (GRAD) items are required by the 500 level students and count as extra credit for the 400 level students.

---

400 [1,30], [2,20], [3,EXTRA CREDIT] , [4,10] = 60
500 [1,30], [2,20], [3, 10] , [4,20] = 80

1. (30) Here is an R function that I made up as a test function:

```
HW2Test<- function(x){
pgamma( x, shape =10, scale =.02) - .5*x
 }
```

   (a) Interpolate this function with polynomials using equally spaced points on the interval [0,1].
   Consider polynomials of increasing degree until the linear system becomes numerically singular (about around degree 17). For each of your interpolating polynomials find the maximum absolute difference (aka sup norm) between the polynomial and the true function on a fine grid of 200 points. (`xGrid <- seq( 0,1, length.out=200)`).

   (b) Make a plot that superimposes a few examples of your interpolations on a fine grid along with the true function. Identify your curves in a legend using color and/or line types.

For some R details see the class script `polyExamples.R`.

(c) Make a plot of how the error of your interpolation changes based on the degree of polynomial. For this exercise for each degree, quantify the error as the root mean squared differences (RMSE) between the interpolation and the true function.

NOTE: If `yHat` is the interpolation (at the 200 points) and `y` is the true function then RMSE is the one liner: `sqrt( mean( ( y - yHat)^2 ))` .

2. (20) Approximate the test function `HW2Test` using a B-spline basis with natural boundary conditions (NBS) and make a similar plot of your interpolants. We have not covered natural splines in lecture but you can use the handy R function `naturalSplineBasis` in the source code file `naturalSplineBasis.R`. The points that you use for interpolation are called the "knots" and as an example here are the 10 NBS basis functions for 10 equally spaced knots in [0,1] and evaluated at 150 points.

```
xGrid<- seq( 0,1,length.out=150)
KN<- seq( 0,1,length.out=10)
B<- naturalSplineBasis( xGrid, sKnots=KN)
matplot( xGrid, B, type="l", lty=1)
```

(a) How does this basis compare to the Bernstein polynomials? How many NBS basis functions have the same "shape".

(b) Here is a function that works the same as `polyFit` but fits a NBS to x and y values. To keep things simple it also assumes that the `x` values are also the knots and the fitted spline is evaluated at the points `xGrid`.

```
NBSFit<- function( x, y, xGrid){
  N<- length( x)
  AData<- naturalSplineBasis( x, sKnots=x)
  coef<- solve( AData, y)
  AGrid<-  naturalSplineBasis( xGrid,  sKnots=x )
  yFit <-AGrid%*%coef
  return( yFit)
}
```

Referring back to Problem 1 now fit the test function to different numbers of points similar to what you did for the polynomials. Go up to 20 equally spaced knot values. Also make a plot of the RMSE error as a function of the number of basis functions. (Remember that for a given number of basis functions, K, you are taking K equally spaced points on [0,1].)

3. (10) (GRAD) Referring to the previous problem. You should be able to interpolate using a NBS basis in R for a large number of basis functions. I cranked

up to 150 basis functions and get an error of 3E-8. Explain why one is able to interpolate in such a stable manner with so many basis functions.

4. (10,20) Consider the set of 4 basis functions,

$$\{\phi_1(x), ..., \phi_4(x)\} = \{\alpha sin(2\pi x), \alpha cos(2\pi x), \alpha sin(4\pi x), \alpha cos(4\pi x)\}$$

on the interval $[0, 1]$ where $\alpha = \frac{1}{\sqrt{2}}$ and with the inner product

$$\langle f, g \rangle = \int_0^{2\pi} f(x)g(x)dx$$

and the norm

$$\|f\| = \sqrt{\langle f, f \rangle}$$

(a) Explain why this basis is orthogonal and each member has a norm of one.
(To do the integrals you can simply give a reference for the indefinite integrals and then evaluate them.)

(b) (GRAD) For the function $f(x) = x(1 - x) = x - x^2$ find the coefficients $\{c_1, c_2, c_3, c_4\}$ so that

$$\phi_1(x)c_1 + \ldots \phi_4(x)c_4$$

is a good approximation to f. I.e. $c_k = \langle f, \phi_k \rangle$.
Feel free to just give a reference for these standard integrals.