EE5471: Fitting of Functions

September 8, 2011

1 Reading Portion

Chapter 5.0, 5.1, 5.3, 5.5, 5.7-5.13, on function evaluation of Numerical Recipes.

2 Aliasing

A Fourier or Chebyshev fit is exact for a function that is expressible as a finite sum of the basis functions. But generally, we have extra terms that are discarded. What this means is that the function is not band-limited.

As we know from DSP, if a function that is not band-limited is sampled, the digital representation suffers from aliasing. Why is this the case? After all, suppose a function is given by

$$f_i = \sum_{k=0}^{\infty} c_k p_k(x_i)$$

Then,

$$c_k = \langle f | p_k \rangle$$

which does not care if higher frequencies are present in f or not. So why is c_k corrupted?

The answer lies in the evaluation of $\langle f | p_k \rangle$. In the analog domain, this inner product has an exact definition as

$$\int_a^b w(x) f(x) p_k(x) dx$$

So the analog representation does not suffer from aliasing. But the *digital inner product depends on N*. i.e.,

$$\langle f | p_k \rangle_N = \sum_{m=1}^{N} w_m f(x_m) p_k(x_m)$$

If I have more terms in my representation (i.e., higher sampling rate), *I am sampling at a different set of points, and indeed I am sampling at more points*.

What does this imply for the problems in this assignment? If you want to get an N term representation (either Fourier or Chebyshev), you need to start with a p_M, $M \gg N$ and then discard the higher order terms. In DSP language that means:

- Oversample the analog signal.
- Digitally low pass filter the signal
- Subsample the filtered digital signal

We will do all this in the signal processing assignment for Fourier analysis. But here, the last step is not needed since we are generating an *analog* approximation to the signal, i.e., the output is not f_i but $f(x)$ for any x.

3 Generating a Fourier Fit

For Chebyshev, the routines to compute the coefficients and to evaluate the function are given. The Fourier equivalent is not considered important in the book, since FFT does the job for us. The method is simply

- Sample the signal at $M = 2^m$ points. The points are the zeros of $\cos(M(x+1)\pi/2)$. The zeros are at $x+1 = 2\frac{k+1/2}{M}$, or

$$x_k = -1 + \frac{2k+1}{M}$$

- Perform the FFT (look up `fft` in scilab)
- Drop all but the first N terms.

  Here some care is required. The `FFT` returns a complex Fourier transform and we want a cosine transform. So combine the complex conjugate terms as follows:

  - $d_0 = c_0$ is unchanged.
  - $d_1 = 0.5\left(c_1 + c_{M-1}\right)$
  - $d_k = 0.5\left(c_k + c_{M-k}\right)$

That is your Fourier approximation. Alternatively, ofcourse, you have that

$$\begin{aligned}\sum_{k=1}^{M} f(x_k)\cos\left(m\left(x_k+1\right)\pi/2\right) &= c_m\sum_{k=1}^{M}\cos^2\left(m\left(x_k+1\right)\pi/2\right)\\ &= c_m\frac{M}{2}\end{aligned}$$

But this is $\mathcal{O}(MN)$ which can be slow compared to FFT which is f{$\mathcal{O}(M\log_2 M)$. Use Clenshaw to compute the Fourier series at a given x.

$$\cos(nx) = 2\cos x\cos((n-1)x) - \cos((n-2)x)$$

So for given coefficients, the fourier series can be computed efficiently for any x.

4 Programming Portion

The following four functions are given, for $-1 < x < 1$.

$$\begin{aligned}f(x) &= \exp(x) &(1)\\ g(x) &= \frac{1}{x^2+\delta^2} &(2)\\ h(x) &= \frac{1}{\cos^2(\pi x)+\delta^2} &(3)\\ u(x) &= \exp(-|x|) &(4)\\ v(x) &= \sqrt{x+1} &(5)\end{aligned}$$

The goal is to fit series approximations to the four functions and to study how well the series converge to the functions. Note that the five functions have the following distinct properties:

- $f(x)$ is a rapidly increasing function that is neither even nor periodic.
- $g(x)$ is a rational, even, aperiodic function that is smooth.
- $h(x)$ is an even, periodic function that has continuous derivatives to all orders.
- $u(x)$ is an even function that has a discontinuous derivitive at $x = 0$.
- $v(x)$ has a branch cut at $x = -1$. Its derivitives have singularities at $x = -1$.

1. Fit Eqs. (1), (2), (3), (4) and (5) to Chebyshev series. Determine the error of the fits as a function of the number of terms kept, for different δ. In the case of $u(x)$, try breaking the range into two parts $-1 \leq x \leq 0$ and $0 < x \leq 1$ and fit. How do the fits compare to fitting $u(x)$ over the whole range?

2. Fit Eq. (1), (2), (3), (4) and (5) to Fourier series, and study the rate at which the coefficients decay in magnitude, for $\delta = 3$. Can you predict this decay rate from properties of the function? For $\delta = 1$ and $\delta = 0.3$ how do the series change? Plot the magnitude of the coefficients vs n for all cases on a log-log plot (you may have to exclude zero coefficients to keep the log-log plot routine from complaining).

   Note: We expect Fourier to do well with a truly periodic function, so $h(x)$ should be handled well. The others will give trouble.

   Note: Remember aliasing! If you wish to use `FFT`, sample the function at far more points than required, take the transform and then truncate. Else, even $u(x)$ will not show quick convergence of coefficients. (see the section above)

3. Fit Eq. (1) to a Taylor series about $x = 0$. This has already been discussed in the orthogonality assignment. But it is instructive to see the error and compare it to chebyshev.

4. Fit Eqs. (1), (2), (3), (4) and (5) to a Rational Chebyshev fit. How does the error scale with δ? For large δ, is it better or worse than a Chebyshev fit with the same number of coefficients?

For which types of problems would you use which of the methods above? Discuss with examples.

5 General Rules of Thumb

- If the function is analytic and periodic, use Fourier. Nothing works as well.
- If the function is analytic and aperiodic, use Chebyshev.
- If the function is piecewise smooth, use Chebyshev on each piece separately. Trying to fiit the $e^{|x|}$ does not work well, while fitting e^x did work well. So just fit e^x in $0 < x < 1$ and e^{-x} in $-1 < x < 0$.

- Even if the function has singularities near the range of fitting, the Chebyshev series coefficients do fall off exponentially. However, there should not be a singularity within the range. Then, no matter how large an M we choose, we always have slowly decreasing coefficients, and are unable to truncate. This is true even when the function is
$$f(x) = \sqrt{x+1}$$
which is well defined $(-1,1)$. However, its derivitive is not nice at -1, despite the function being continuous. So the Chebyshev series does not do a good job.

- The rate at which the coefficients decay is clearly linked to the distance of the poles to the region, i.e., to the radius of convergence.

- Sometimes we have a tricky function to fit. Suppose we have a function like
$$f(x) = e^{-x} + e^{-0.01x}, \quad 0 < x < 100$$
Then, over most of the range, it is $e^{-0.01x}$ that we see, yet the series has to have enough terms to capture the e^{-x} for small x. This can lead to a large number of terms in the Chebyshev series. Such a problem can be solved by breaking the domain into $0 < x < 5$ and $5 \leq x < 100$. Now series in both regions will converge quickly.

- If $f(x)$ has a chebyshev series,
$$f(x) = \sum_{k=0}^{N} c_k T_k(x)$$
we can deduce the chebyshev series for the indefinite integral of $f(x)$. This is given by
$$\int^x f(x)dx = C + \sum_{k=1}^{N-1} \frac{c_{k-1} - c_{k+1}}{2k} T_k(x) + \frac{c_{N-1}}{2N} T_N(x)$$
where C is the constant of integration.

- There is a similar formula for the derivitive. Perhaps the most important thing about the dervitive is that unlike even the fourier series where the higher order coefficients get amplified and make the signal very choppy, the derivitive is one order lower than the fit. So it is actually smoother in some sense than the original function. The derivitive recursion formula is (for coefficients starting with index 1)
$$d_{i+1} - d_{i-1} = -2\,(i-1)\,c_i, \quad i = 2, 3, \ldots n-1$$
As usual, it is best to implement this using Clenshaw algorithm and get the new series.