Approximation of functions for AMSC 460
Lecture 1

Prof. Jacob Bedrossian
University of Maryland, College Park

These notes will supplement the lectures on approximation of functions.

Suppose one wants to solve a differential equation for a physical quantity of interest. In general, you won't be able to find, or represent, the exact solution using your computer. Instead, you will generate an *approximation.*

Let us start by being precise about what kinds of questions we want to answer with a specific interpolation example. Suppose one has a function $f : [0, 1] \to \mathbb{R}$ and for each $n \geq 1$ we build an approximation $f_n(x)$ as a piecewise linear function: define $h = 1/n$ and

$$f_n(x) = f(hj) - \frac{f(h(j+1) - f(hj)}{h}(x - hj) \quad x \in [jh, (j+1)h]. \tag{1}$$

This functon, $f_n(x)$, is the piecewise linear continuous function which satisfies $f(jh) = f_n(jh)$ for all $0 \leq j \leq n$. As in the interpolation problem, in real applications you often don't really know the 'true' function f but let us suppose for now that we do. The next question you want to ask is: "how good of an approximation of f is f_n" or "how close is f_n to f for n large?". The immediate next question you should ask though is "wait...what do I actually mean by *close*?" This brings us to the concept of a norm, which provides a measure of distance and size for functions.

Definition 1. Let V be a vectorspace. A function $\| \cdot \| : V \to \mathbb{R}$ is called a *norm* if

1. for all $f \in V$, $\|f\| \geq 0$ and $\|f\| = 0$ if and only if $f = 0$;
2. for all $\lambda \in \mathbb{R}$ and $f \in V$, $\|\lambda f\| = |\lambda| \, \|f\|$;
3. for all $f, g \in V$, $\|f + g\| \leq \|f\| + \|g\|$.

Property (3) is called *the triangle inequality.*

The idea is that $\|f\|$ measures the size of f and $\|f - g\|$ will measure the "distance" between f and g.

Example 1. For the vectorspace $\mathbb{R}^n$, the regular Euclidean distance

$$\|v\| = \left(\sum_{i=1}^{n} v_i^2 \right)^{1/2}, \tag{2}$$

is a norm (the proof of the triangle inequality in Definition 1 is not so obvious, and is covered in e.g. Math 405 and/or Intro linear algebra).

Hence, the idea would be to define a norm and then say that f_n and f are "close" if $\|f_n - f\|$ is small. However, there are many different norms defined for functions. We will only really discuss two or three of them:

$$\|f\|_{L^\infty(a,b)} = \max_{x \in [a,b]} |f(x)| \tag{3a}$$

$$\|f\|_{L^2(a,b)} = \left(\int_a^b |f(x)|^2 \, dx \right)^{1/2} \tag{3b}$$

$$\|f\|_{L^1(a,b)} = \int_a^b |f(x)| \, dx. \tag{3c}$$

referred to as the L-infinity, L-two, and L-one norms respectively. If the (a, b) is omitted from the subscript, assume $a = 0$ and $b = 1$ is meant. From the triangle inequality for $|\cdot|$, we can see that (3c) and (3a) satisfy the triangle inequality too. It is less clear that (3b) does as well, but this is true (and after a few more observations, can be proved in the same manner as the triangle inequality for vectors in $\mathbb{R}^n$).

You might hope that these norms are all pretty similar, in that $\|f\|_{L^2} \ll 1$ implies $\|f\|_{L^\infty} \ll 1$ and what not. However, an important, surprisingly subtle, point is that this is *false* as the next example shows:

Example 2. Consider the sequence of functions

$$f_n(x) = n \qquad 0 \le x \le 2^{-n} \tag{4}$$

$$f_n(x) = 0 \qquad \text{otherwise.} \tag{5}$$

Then

$$\|f_n\|_{L^\infty} = n \tag{6}$$

$$\|f_n\|_{L^2} = n2^{-n/2}. \tag{7}$$

Hence, $\|f_n\|_{L^\infty} \to \infty$ and $\|f_n\|_{L^2} \to 0$.

Example 2 shows that the same sequence of functions can be arbitrarily large when measured in one norm and arbitrarily small when measured in another. One can prove (in Math 411) that this is impossible for vectors in $\mathbb{R}^n$, its a something that can only happen in infinite dimensional vector-spaces. Hence, we see that when quantifying the efficacy of our approximations, we will need to be precise about what norm we are measuring in. Not everything is lost though, you can observe the following:

$$\|f\|_{L^2(a,b)} \le \|f\|_{L^\infty(a,b)} |b - a|^{1/2}\,, \tag{8}$$

so as long as $|b - a| < \infty$, a function can be arbitrarily large in L^∞ and arbitrarily small in L^2 but *not* vice-versa. Finally, for plenty scientific computing applications, the L^2 and L^∞ norms behave in a similar manner. However, I wanted to emphasize this subtlety for two reasons: (A) there are plenty of applications where this is not true, for example, in physics calculations in which the solution has sharp transitions, e.g. at the interface of water and air etc; (B) even in more mundane examples, you may end up considering a number of other norms, which will generally not behave the same.

Let us return to the example of f_n given by linear approximation. Let us compute how good of an approximation this is. As usual, we use Taylor's theorem. For $x \in [jx, (j+1)x]$,

$$f(x) - f_n(x) = f(x) - f(hj) - \frac{f(h(j+1) - f(hj)}{h}(x - hj) \tag{9}$$

Then we write the expansion:

$$f(x) = f(hj) + f'(hj)(x - hj) + O(h^2) \tag{10}$$

$$\frac{f(h(j+1) - f(hj)}{h} = f'(hj) + O(h) \tag{11}$$

and hence,

$$f(x) - f(hj) - \frac{f(h(j+1) - f(hj)}{h}(x - hj) = \left(f'(hj) - \frac{f(h(j+1) - f(hj)}{h}\right)(x - hj) + O(h^2) \tag{12}$$

$$= O(h^2). \tag{13}$$

Hence,

$$\|f - f_n\|_{L^\infty} \leq \sup_{0 \leq j \leq n-1} \|f - f_n\|_{L^\infty(jh,(j+1)h)} = O(h^2). \tag{14}$$

Hence, we say that the approximation is *second order* in L^∞. One can verify that the approximation is also $O(h^2)$ in L^2 (for example, this follows from (8)).

Consider again the piecewise linear approximation. Analogous to the Lagrange polynomials, one can write the approximation $f_n(x)$ as a linear combination of fixed, standard functions:

$$f_n(x) = \sum_{j=0}^{n} f(jh) T_j(x), \tag{15}$$

where T_j is defined as

$$T_j(x) = \frac{1}{h}(x - jh) \qquad x \in [(j-1)h, jh] \tag{16}$$

$$T_j(x) = \frac{1}{h}\left((j+1)h - x\right) \qquad x \in [jh, (j+1)h] \tag{17}$$

$$T_j(x) = 0 \qquad \text{otherwise.} \tag{18}$$

The T stands for "tent function" (terminology which you will understand if you draw a picture). Hence, one could say that we first chose an approximation space: $V_n = \mathrm{span}(T_0, T_1, ..., T_n)$ and then chose an $f_n \in V_n$ which is a good approximation of f. As we will see, this viewpoint is especially useful for working in L^2, because we have the *inner product*:

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx, \tag{19}$$

which is the analogue of the dot product between functions. For example, notice that

$$\|f\| = \sqrt{\langle f, f \rangle}, \tag{20}$$

and similarly

$$\|f + g\|_{L^2}^2 = \|f\|_{L^2}^2 + 2\langle f, g \rangle + \|g\|_{L^2}^2. \tag{21}$$

Hence, if $\langle f, g \rangle = 0$ then we have an analogue of the Pythagorean identity: if $\langle f, g \rangle = 0$ then

$$\|f + g\|_{L^2}^2 = \|f\|_{L^2}^2 + \|g\|_{L^2}^2. \tag{22}$$

This motivates the definition

Definition 2. Two functions f, g are called *orthogonal* if $\langle f, g \rangle = 0$. A set of functions F is called orthogonal if for all $f, g \in F$ with $f \neq g$, there holds $\langle f, g \rangle = 0$. We similarly call f, g *orthonormal* if $\|f\|_{L^2} = \|g\|_{L^2} = 1$.

Consider the problem of choosing an approximation space $V_n = \text{span}(g_1, ..., g_n)$ for a given set of linearly independent functions g_i and then finding the approximation $f_n \in V_n$ which is closest to a given function f in L^2. That is, we are looking to find $f_n \in V_n$ which solves the following minimization problem:

$$\|f - f_n\|_{L^2}^2 = \min_{v \in V_n} \|f - v\|_{L^2}^2. \tag{23}$$

Notice, however, that this is actually the familiar least squares problem. Indeed, if we write $v = \sum_{i=1}^n c_i g_i$ we can expland the problem we are trying to minimize to:

$$\|f - v\|_{L^2}^2 = \|f\|^2 - 2\sum_{i=1}^n c_i \langle g_i, f\rangle + \sum_{i=1}^n \sum_{j=1}^n c_i c_j \langle g_i, g_j\rangle := \phi(c) \tag{24}$$

Hence, we are trying to find a minimum of a quadratic polynomial, not some crazy infinite dimensional object you don't understand. As in $\mathbb{R}^n$, we want f_n to be the orthogonal projection of f onto V_n. Just like in finite dimensions, this is most easily formed if the g_i are orthonormal.

Theorem 1. *Let $V_n = \text{span}(g_1, ..., g_n)$ and that the g_i's are orthonormal. Then the unique solution to the minimization problem* (23) *is given by the orthogonal projection:*

$$f_n = \sum_{i=1}^n \langle g_i, f\rangle g_i. \tag{25}$$

Proof. The proof is exactly the same as it was in $\mathbb{R}^n$!. □

The piecewise linear example shows that there are probably important settings for which we want to be able to solve (23) when the g_i's are not an ONB. We have two options. One option is to use Gram-Schmidt to orthogonalize the basis, which is essentially what we did to compute to solve the least squares problem via QR factorization. Another option is to try and do it directly by looking for where the gradient of (24) vanishes: that is, the set of c_j's such that the following vanishes for each k:

$$0 = \partial_{c_k}\phi(c) = -2\langle g_k, f\rangle + 2\sum_{j=1}^n \langle g_k, g_j\rangle c_j. \tag{26}$$

This is simply a linear system, so if you can solve a linear system with the matrix A whose entries are $a_{kj} = \langle g_k, g_j\rangle$ you can find the critical point of ϕ. As it happens, this matrix is SPD and indeed the critical point is unique minimizer.

Lemma 1. *Let $\{g_1, ..., g_n\}$ be non-zero, linearly independent functions with $\|g_i\|_{L^2} < \infty$. Then, the matrix A whose entries are $a_{kj} = \langle g_k, g_j\rangle$ is symmetric positive definite.*

Proof. Recall the definition from the linear algebra notes: A is positive definite if $x^T Ax > 0$ for all $x \in \mathbb{R}^n$ with $x \neq 0$. Let $x \in \mathbb{R}^n$. Then,

$$x^T Ax = \sum_{i=1}^n \sum_{j=1}^n x_i x_j \langle g_i, g_j\rangle = \langle \sum_{i=1}^n x_i g_i, \sum_{j=1}^n x_j g_j\rangle = \|\sum_{i=1}^n x_i g_i\|_{L^2}^2. \tag{27}$$

Since the g_i are linearly independent and non-zero, this must be strictly positive provided that $x \neq 0$. Therefore A is symmetric positive definite. □

The above calculations show that we can find the orthogonal projection onto V_n without doing any sort of QR factorization if we are willing to solve an SPD linear system. SPD linear systems are faster to solve than QR factorizations are to compute (compare cost of QR and Cholesky – and the difference is even much more stark when you are looking to solve a large system, which would occur if you want a more accurate representation), so there are lots of applications when it would be preferred that we simply do this directly, rather than trying to find an ONB (for example in the case of the "tent functions" above).

There are still many cases when one uses one of a several sets of standard orthogonal functions (and so they do not need to be computed via Gram-Schmidt or QR). The two primary examples are *orthogonal polynomials* and (much more importantly) *Fourier analysis.* There are many families for orthogonal polynomials which arise from choosing slightly different versions of the L^2 inner product (for example $\langle f, g\rangle = \int_a^b f(x)g(x)e^{x^2}dx$ behaves a lot like the L^2 inner product in a lot of ways but is nevertheless different and different functions are orthogonal with this inner product than the standard one). Let us discuss just the example for the standard L^2 inner product, which are the Legendre polynomials $P_0(x), P_1(x), ...$ Defined on $x \in [-1, 1]$ these are formed by starting with the regular basis $\{1, x, x^2, x^3, ...\}$ and applying Gram-Schmidt. We can see from Gram-Schmidt that $P_0(x) = 1$ and $P_0(x) = x$. With some work we can prove that the rest are generated via the relatively simple iteration formula (and so a more complicated full Gram-Schmidt is redundant):

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x). \tag{28}$$

As defined, these polynomials will be orthogonal but not normalized:

$$\langle P_i, P_j\rangle = \begin{cases} \frac{2}{2n+1} & i = j \\ 0 & i \neq j. \end{cases} \tag{29}$$

The Fourier transform is one of the most revolutionary ideas in all of science and is the building block for many practical algorithms and widely used technologies as well as several entire branches of mathematics, so it makes sense that we briefly discuss it here.

Let us discuss the Fourier sine transform first. Let us consider the sequence of functions $\{\sin nx\}_{n=1}^{\infty}$ defined on $x \in [0, \pi]$. This set of functions is orthogonal, indeed, a trig calculation gives

$$\langle \sin nx, \sin mx\rangle = \begin{cases} \pi/2 & n = m \\ 0 & n \neq m. \end{cases} \tag{30}$$

Hence, for any finite n, we can define the *truncated sine transform*

$$f_n(x) = \sum_{j=1}^{n} c_j \sin jx, \tag{31}$$

with

$$c_j = \frac{2}{\pi}\int_0^{\pi} f(x)\sin jx dx. \tag{32}$$

The functions $\sin jx$ are called "sine modes" or "Fourier modes". Notice that for all $n < \infty$, $f_n(0) = f_n(\pi) = 0$. Measured in L^2, the sequence of functions f_n will nevertheless approximate any reasonable function f defined on $[0, \pi]$. However, its clear that if $f(0) \neq 0$ for example, that $\|f_n - f\|_{L^\infty} \geq |f(0)|$, and hence the approximation will not converge in L^∞. Another option is to use the Fourier cosine transform, which is formed from the orthogonal set of cosines: $\{\cos nx\}_{n=0}^{\infty}$ defined on $x \in [0, \pi]$, and notice that the behavior at the boundary is different from the sine transform. Specifically, we have instead that $f_n'(0) = f_n'(\pi) = 0$, whereas the functions themselves are generally non-zero. This transform will make a good approximation if this condition holds on f and will have issues otherwise. There are many applications where it is clear which one you should use (specifically, differential equations) and there are still other variants. An even more natural place to use the Fourier transform is to form the problem on $[-\pi, \pi]$ and use both sets of trigonometric functions $\{\sin nx\}_{n=1}^{\infty} \cup \{\cos nx\}_{n=0}^{\infty}$ (it is an exercise in trig to verify that this is an orthogonal set). Hence, we are defining the truncated Fourier transform:

$$f_n(x) = \sum_{j=1}^{\infty} b_j \sin jx + \sum_{j=0}^{\infty} c_j \cos jx, \tag{33}$$

and we verify that

$$b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin jx dx \tag{34}$$

$$c_0 = \frac{1}{2\pi} \int f(x) dx \tag{35}$$

$$c_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos jx dx \quad j \geq 1 \tag{36}$$

This sequence will make a good approximation of nice smooth f's provided $f(-\pi) = f(\pi)$ and $f'(\pi) = f'(-\pi)$, otherwise, it generally suffer like the sine transform and cosine transform. The Fourier transform is such an important part of differential equations and image/signal processing that a very clever and powerful algorithm was devised for computing the discrete versions of the Fourier transform and also for rapidly computing $f_n(x)$ at specified grid points given the coefficients b_j and c_j (the discrete inverse Fourier transform). This influential algorithm is called the *fast Fourier transform* (always referred to as an FFT) and operates in roughly $O(n^2 \log n)$ time. Logarithms are pretty small, so this is only a little slower than doing a back substitution solve and is much faster than something like a Cholesky factorization. If there's time and interest at the end of the semester I will discuss this algorithm.

Unfortunately, finding the approximation error for these least-squares approximations is a little beyond the scope of the course, however, in many cases, the least-squares approximations built via orthogonal polynomials and with Fourier series can converge *exponentially.* That is, if f is smooth and satisfies boundary conditions matching our type of Fourier transform, we often get error estimates of the form

$$\|f - f_n\|_{L^2} \leq Ce^{-\lambda n} \tag{37}$$

for some $C, \lambda > 0$. This is an absurdly fast convergence estimate, so there are lots of applications where using approximation via orthogonal functions can be a great idea if its possible (these are sometimes called 'spectral methods' for reasons you will learn if you take Math 462). However, if the function violates the boundary conditions or is not smooth, one can have slower convergence. For example, if we consider the following function on $[-\pi, \pi]$,

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0, \end{cases} \tag{38}$$

the convergence in L^2 will be slow and moreover the approximations $f_n(x)$ will have strange oscillatory artifacts (for example $f_n(x)$ will take negative values somewhere for all n). The effort to understand the convergence of Fourier series helped bring mathematical analysis into the modern age (for example, modern integration theory was devised by Lebesgue, in part, to study this problem). More details will be discussed in a course on harmonic analysis and/or signal processing.