Lecture 6: Polynomial Interpolation II

Instructor: Professor Amos Ron Scribes: Yunpeng Li, Mark Cowlishaw, Nathanael Fillmore

1 Polynomial Interpolation as a Linear Problem

Last time, we introduced the polynomial interpolation problem, which we now state in a different (but equivalent) fashion:

DEFINITION 1.1 (Polynomial Interpolation). *Given points $X = (x_0, x_1, \ldots, x_n)$ which are pairwise disjoint (that is, no two points are equal) and a function f defined at these points. Find a polynomial $p \in \Pi_n$ such that:*

$$p(x_i) = f(x_i) \quad \forall_{0 \le i \le n}$$

Note that the polynomial p may not fit the function f very well outside the given points, but for now, we will not concern ourselves with this.

If we choose to represent p as a linear combination of monomials, then the problem of finding p is equivalent to solving a linear problem with $n+1$ variables (the coefficients of the monomials) and $n+1$ constraints:

$$p(t) = a(1)t^n + a(2)t^{n-1} + \cdots + a(n)t + a(n+1)$$

We can rewrite this system of equations as:

$$V\vec{a} = \vec{F} \tag{1}$$

where $\vec{a} = (a(1), a(2), \ldots, a(n+1))^T$, $F = (f(x_0), \ldots, f(x_n))^T$ and

$$V = \begin{pmatrix} x_0^n & x_0^{n-1} & \ldots & x_0^1 & x_0^0 \\ x_1^n & x_1^{n-1} & \ldots & x_1^1 & x_1^0 \\ \vdots & \vdots & & \vdots & \vdots \\ x_{n-1}^n & x_{n-1}^{n-1} & \ldots & x_{n-1}^1 & x_{n-1}^0 \\ x_n^n & x_n^{n-1} & \ldots & x_n^1 & x_n^0 \end{pmatrix}$$

Transforming the problem of finding a polynomial p into an equivalent linear problem has helped us to understand polynomial interpolation. However, it has some significant drawbacks as a method for solving polynomial interpolation problems.

1. Solving a linear system of equations takes a significant amount of time.
2. This system of linear equations is often ill-conditioned and prone to round-off errors. In other words, although the Vandermonde matrix V is nonsingular, it may be close to singular, which makes it hard to solve accurately on a computer.

3. It is not well-suited to a common kind of exploratory analysis. Suppose we are given $(x_0, y_0), \ldots, (x_n, y_n)$. We solve $Va = y$, and we get a solution a. Suppose that we don't like the solution for some reason, so we add a new point (x_{n+1}, y_{n+1}). Now we make a new system $\tilde{V}\tilde{a} = \tilde{y}$ and solve for $\tilde{a}$. We have to do this from scratch – we can't reuse all the computation we already invested to solve $Va = y$. If $n = 1000$, say, this could be very bad.

This equivalence between solving a linear problem and finding a polynomial p rests on our representation of p as a linear combination of monomials. To find a better method, we should consider alternate representations. Two methods of polynomial interpolation we shall talk about involve different polynomial representations. These are *Lagrange*[1] *polynomials* and *Newton polynomials*.

2 Interpolation by Lagrange Polynomials

Given the $n+1$ interpolation points $x_0, x_1, \ldots, x_n$, suppose we can find $n+1$ polynomials $\ell_0(t), \ell_1(t), \ldots, \ell_n(t)$ of degree $\leq n$. These polynomials are fixed and independent of the function values $f(x_0), f(x_1), \ldots, f(x_n)$, and satisfy the following condition:

$$\ell_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{2}$$

For example, given x_0 and x_1, two such polynomials are shown in Figure 1.

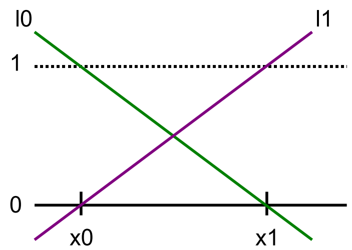

Figure 1: Lagrange Polynomials ℓ_0, ℓ_1 for two points x_0, x_1

How can we find such polynomials, and, more importantly, how can we use them for interpolation?

2.1 Using Lagrange Polynomials for Interpolation

Fact 2.1. *Any polynomial $p \in \Pi_n$ can be represented as a linear combination of $n+1$ Lagrange polynomials of degree $\leq n$.*

To illustrate the use of Lagrange polynomials, consider the following example.

[1] Joseph-Louis Comte de Lagrange. 1736-1813. French mathematician; published works on celestial mechanics, differential and variable calculus, theory of numbers, etc.

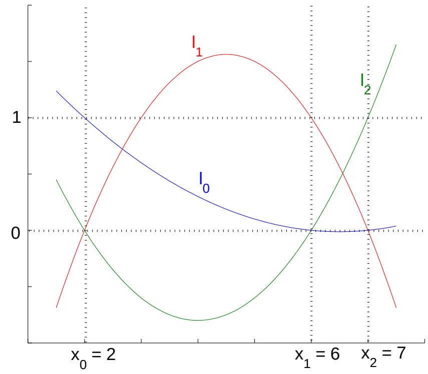

Figure 2: Quadratic Lagrange Polynomials ℓ_0, ℓ_1, ℓ_2 for $\vec{x} = (2\ 6\ 7)^T$

EXAMPLE 2.1. *Given the points* $\vec{x} = (2\ \ 6\ \ 7)^T$*, and the corresponding function values* $\vec{F} = (-1\ \ 8\ \ -3)^T$*, find an interpolating polynomial* $p \in \Pi_2$*.*

Consider that, if we had three quadratic Lagrange polynomials $\ell_0(t), \ell_1(t), \ell_2(t)$ (as shown in Figure 2) with

$$\begin{aligned} \ell_0(\vec{x}) &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T \\ \ell_1(\vec{x}) &= \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T \\ \ell_2(\vec{x}) &= \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T \end{aligned}$$

If we multiply each polynomial l_i by the corresponding function value $f(x_i)$, then add them together, the resulting polynomial will match f at each of the input points. So, we can define p as:

$$p(t) = -1 \cdot \ell_0(t) + 8 \cdot \ell_1(t) - 3 \cdot \ell_2(t)$$

Evaluating p at the input point $x_0 = 2$, we see:

$$\begin{aligned} p(2) &= -1 \cdot \ell_0(2) + 8 \cdot \ell_1(2) - 3 \cdot \ell_2(2) \\ &= -1 \cdot 1 + 8 \cdot 0 - 3 \cdot 0 \\ &= -1 \end{aligned}$$

In general, given a set of input points $x_0, x_1, \ldots, x_n$, the corresponding points on the graph of the function f, $f(x_0), f(x_1), \ldots, f(x_n)$, and the Lagrange polynomials for our input points $\ell_0(t), \ell_1(t), \ldots, \ell_n(t)$, we can define the following polynomial $p \in \Pi_n$ that interpolates f:

$$p(t) = \sum_{i=0}^{n} f(x_i)\ell_i(t) \tag{3}$$

What have we done? Initially, when we first formulated the problem of polynomial interpolation, we did not specify any particular representation of polynomials. When we started to do computation, we needed a computationally accessible definition, so we choose the most natural basis that we all are familiar with: monomials. For example, we used the monomials $\{t^0, t^1, t^2\}$ as a basis for Π_2. But this is only one basis. The Lagrange polynomials also form a basis for the space of polynomials. The difference is that the Lagrange basis is chosen so as to make our problem easier.

2.2 Finding Lagrange Polynomials

So how can we find a polynomial ℓ_i given its roots and a point x_i with $\ell_i(x_i) = 1$? Consider that, if a polynomial has roots r_1, r_2, then it must have factors $(t - r_1)$ and $(t - r_2)$. We can get our polynomial by first multiplying the factors together, obtaining a polynomial with the correct roots. We can then multiply by a constant scaling factor so that $\ell_i(x_i) = 1$.

To illustrate, consider Example 2.1. Since ℓ_0 is 0 at 6 and 7, it must have factors $(t - 6)$ and $(t-7)$. If we consider $(t-6)\cdot(t-7)$ evaluated at $x_0 = 2$, we see that its value will be $(2-6)\cdot(2-7)$, so, if we divide by this value, we will have the correct polynomial:

$$\ell_0(t) = \frac{(t-6)\cdot(t-7)}{(2-6)\cdot(2-7)}$$

We can define ℓ_1 and ℓ_2 similarly:

$$\ell_1(t) = \frac{(t-2)\cdot(t-7)}{(6-2)\cdot(6-7)}$$
$$\ell_2(t) = \frac{(t-2)\cdot(t-6)}{(7-2)\cdot(7-6)}$$

In general, we can calculate each Lagrange polynomial ℓ_i for a set of input points $x_0, x_1, \ldots, x_n$ as:

$$\ell_i(t) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \left(\frac{(t - x_j)}{(x_i - x_j)} \right) \tag{4}$$

2.3 Analysis of Interpolation by Lagrange Polynomials

So we have seen that we can solve a polynomial interpolation problem quite easily using a linear combination of Lagrange polynomials. We should ask ourselves how easy it is to use the resulting polynomial for numerical computation. For example:

- How easy is it to evaluate?
- How easy is it to evaluate the derivative?

We might argue that evaluating the polynomial is fairly straightforward. However, it is easy to see that evaluating the derivative could be very complicated (consider using the product rule on n terms with n factors each).

Thus, with Lagrange polynomials, we have struck on a polynomial representation that makes it easy to solve polynomial interpolation problems, but potentially difficult to use once the problem has been solved.

How important is it that the polynomial representation we end up with is easy to use? Consider that we have already used polynomial interpolation without knowing it, when we approximated the derivative of a function.

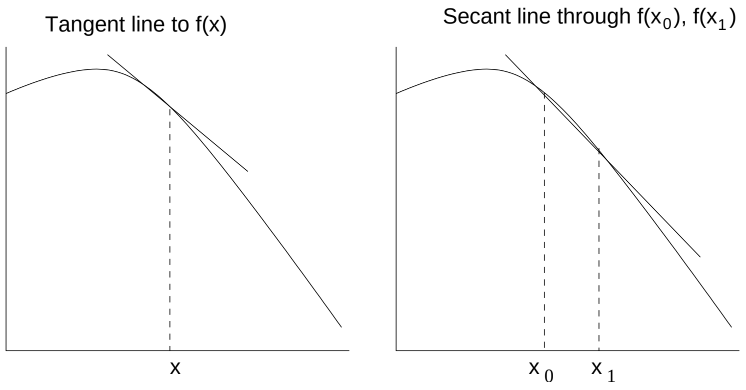

Figure 3: Approximating the Derivative Using a Secant Line

Recall that the derivative of a function f at a point $(x, f(x))$ is simply the slope of the tangent line at that point. We can approximate the tangent line by taking two points $(x_0, f(x_0))$, $(x_1, f(x_1))$, and interpolating a degree ≤ 1 polynomial (a secant line) that approximates the function between these two points (as shown in Figure 3). We can then approximate the slope at x by taking the derivative of the secant line. If we look at this problem as a polynomial interpolation problem, we might get a better estimate by interpolating over a larger number of points. In cases like this, clearly it is important that the representation of the resulting polynomial is easy to use.

2.4 Uniqueness of Solutions

The process of interpolation by Lagrange polynomials shows that we can *always* find a solution to a polynomial interpolation problem.

Recall that polynomial interpolation is equivalent to solving the linear problem:

$$V\vec{a} = \vec{F} \tag{5}$$

From linear algebra, we know that the solution to this problem hinges on whether or not the matrix V is *singular*. If V is singular, then whether or not the equation has a solution depends on $\vec{F}$:

$$\begin{array}{rcl} V \text{ is non-singular} & \Rightarrow & \forall \vec{F} \text{ equation 5 has a unique solution} \\ V \text{ is singular} & \Rightarrow & \begin{cases} \exists \vec{F} \text{ such that equation 5 has no solutions} \\ \exists \vec{F} \text{ such that equation 5 has infinitely many solutions} \end{cases} \end{array}$$

Now, since we know that we can *always* solve a polynomial interpolation problem using Lagrange polynomials, we know that equation 5 *always* has a solution, regardless of $\vec{F}$. Since this is true, V cannot be singular, and therefore, we know that not only does equation 5 always have a solution, but that solution is always unique.