interpolation

L. Olson

Department of Computer Science
University of Illinois at Urbana-Champaign

semester plan

Tu Nov 10	Least-squares and error
Th Nov 12	Case Study: Cancer Analysis
Tu Nov 17	**Building a basis for approximation (interpolation)**
Th Nov 19	non-linear Least-squares
Tu Dec 01	non-linear Least-squares
Th Dec 03	optimization methods
Tu Dec 08	Elements of Simulation + Review

interpolation

Today's ojbectives:

1. Take a few points and **interpolate** instead of **fit**
2. Write the interpolant as a combination of *basis* functions
3. Implemente interpolation with several types of basis functions
4. Construct interpolation through a linear algebra problem

interpolation: introduction

Objective

Approximate an unknown function $f(x)$ by an easier function $g(x)$, such as a polynomial.

Objective (alt)

Approximate some data by a function $g(x)$.

Types of approximating functions:

1. Polynomials
2. Piecewise polynomials
3. Rational functions
4. Trig functions
5. Others (inverse, exponential, Bessel, etc)

interpolation: introduction

How do we approximate $f(x)$ by $g(x)$? In what sense is the approximation a good one?

1. Least-squares: $g(x)$ must deviate as little as possible from $f(x)$ in the sense of a 2-norm: minimize $\int_a^b |f(t) - g(t)|^2 \, dt$
2. Chebyshev: $g(x)$ must deviate as little as possible from $f(x)$ in the sense of the ∞-norm: minimize $\max_{t \in [a,b]} |f(t) - g(t)|$.
3. **Interpolation: $g(x)$ must have the same values of $f(x)$ at set of given points.**

polynomial interpolation

Given $n+1$ distinct points $x_0, \ldots, x_n$, and values $y_0, \ldots, y_n$, find a polynomial $p(x)$ of degree n so that

$$p(x_i) = y_i \quad i = 0, \ldots, n$$

- A polynomial of degree n has $n+1$ degrees-of-freedom:

$$p(x) = a_0 + a_1 x + \cdots + a_n x^n$$

- $n+1$ constraints determine the polynomial uniquely:

Theorem

If points $x_0, \ldots, x_n$ are distinct, then for arbitrary $y_0, \ldots, y_n$, there is a *unique* polynomial $p(x)$ of degree at most n such that $p(x_i) = y_i$ for $i = 0, \ldots, n$.

monomials

First attempt: try picking

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

So for each x_i we have

$$p(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n = y_i$$

OR

$$\begin{aligned}
a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n &= y_0 \\
a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n &= y_1 \\
a_0 + a_1 x_2 + a_2 x_2^2 + \cdots + a_n x_2^n &= y_2 \\
a_0 + a_1 x_3 + a_2 x_3^2 + \cdots + a_n x_3^n &= y_3 \\
&\vdots \\
a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n &= y_n
\end{aligned}$$

monomial: the problem

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ & & & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Question

- Is this a “good” system to solve?

example

Consider Gas prices (in cents) for the following years:

x	year	1986	1988	1990	1992	1994	1996
y	price	133.5	132.2	138.7	141.5	137.6	144.2

```
year = np.array([1986, 1988, 1990, 1992, 1994, 1996])
price= np.array([133.5, 132.2, 138.7, 141.5, 137.6, 144.2])

M = np.vander(year)
a = np.linalg.solve(M,price)

x = np.linspace(1986,1996,200)
p = np.polyval(a,x)
plt.plot(year,price,'o',x,p,'-')
```

back to the basics...

Example

Find the interpolating polynomial of least degree that interpolates

x	1.4	1.25
y	3.7	3.9

Directly

$$\begin{aligned} p_1(x) &= \left(\frac{x-1.25}{1.4-1.25}\right)3.7 + \left(\frac{x-1.4}{1.25-1.4}\right)3.9 \\ &= 3.7 + \left(\frac{3.9-3.7}{1.25-1.4}\right)(x-1.4) \\ &= 3.7 - \frac{4}{3}(x-1.4) \end{aligned}$$

lagrange

What have we done? We've written $p(x)$ as

$$p(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) y_0 + \left(\frac{x - x_0}{x_1 - x_0}\right) y_1$$

- the sum of two linear polynomials
- the first is zero at x_1 and 1 at x_0
- the second is zero at x_0 and 1 at x_1
- these are the two linear Lagrange basis functions:

$$\ell_0(x) = \frac{x - x_1}{x_0 - x_1} \qquad \ell_1(x) = \frac{x - x_0}{x_1 - x_0}$$

Example

Write the Lagrange basis functions for

x	$\frac{1}{3}$	$\frac{1}{4}$	1
y	2	-1	7

Directly

$$\ell_0(x) = \frac{(x - \frac{1}{4})(x - 1)}{(\frac{1}{3} - \frac{1}{4})(\frac{1}{3} - 1)}$$

$$\ell_1(x) = \frac{(x - \frac{1}{3})(x - 1)}{(\frac{1}{4} - \frac{1}{3})(\frac{1}{4} - 1)}$$

$$\ell_2(x) = \frac{(x - \frac{1}{3})(x - \frac{1}{4})}{(1 - \frac{1}{3})(1 - \frac{1}{4})}$$

lagrange

The general Lagrange form is

$$\ell_k(x) = \prod_{i=0, i \neq k}^{n} \frac{x - x_i}{x_k - x_i}$$

The resulting interpolating polynomial is

$$p(x) = \sum_{k=0}^{n} \ell_k(x) y_k$$

example

Find the equation of the parabola passing through the points (1,6), (-1,0), and (2,12)

$x_0 = 1, x_1 = -1, x_2 = 2; \qquad y_0 = 6, y_1 = 0, y_2 = 12;$

$$\begin{aligned}
\ell_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} &= \frac{(x+1)(x-2)}{(2)(-1)} \\
\ell_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} &= \frac{(x-1)(x-2)}{(-2)(-3)} \\
\ell_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} &= \frac{(x-1)(x+1)}{(1)(3)}
\end{aligned}$$

$$\begin{aligned}
p_2(x) &= y_0\ell_0(x) + y_1\ell_1(x) + y_2\ell_2(x) \\
&= -3 \times (x+1)(x-2) + 0 \times \frac{1}{6}(x-1)(x-2) \\
&\quad +4 \times (x-1)(x+1) \\
&= (x+1)[4(x-1) - 3(x-2)] \\
&= (x+1)(x+2)
\end{aligned}$$

summary so far:

- Monomials: $p(x) = a_0 + a_1 x + \cdots + a_n x^n$ results in poor conditioning
- Monomials: but evaluating the Monomial interpolant is cheap (nested iteration)
- Lagrange: $p(x) = \ell_0(x) y_0 + \cdots + \ell_n(x) y_n$ is very well behaved.
- Lagrange: but evaluating the Lagrange interpolant is expensive (each basis function is of the same order and the interpolant is not easily reduced to nested form)

fixing monomials, fixing lagrange

Back to the gas price example. Suppose we use a better basis like

$$(x - \bar{x})^k$$

instead of

$$x^k$$

For example, $\bar{x} = average(x_i), i = 0, \ldots, n$.

The basis $(x - \bar{x})^k$ are called *shifted monomials* because x is shifted by $\bar{x}$.

recall: monomials

Obvious attempt: try picking

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

So for each x_i we have

$$p(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n = y_i$$

OR

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ & & & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

That is,

$$a = M^{-1} y$$

Very bad matrix: terribly ill-conditioned, inverse entries are **large**

Very bad evaluation: values are **huge**

recall: lagrange

The general Lagrange form is

$$\ell_k(x) = \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i}$$

The resulting interpolating polynomial is

$$p(x) = \sum_{k=0}^{n} \ell_k(x) y_k$$

example

Find the equation of a quadratic passing through the points (0,-1), (1,-1), and (2,7).

$x_0 = 0, x_1 = 1, x_2 = 2 \qquad y_0 = -1, y_1 = -1, y_2 = 7$

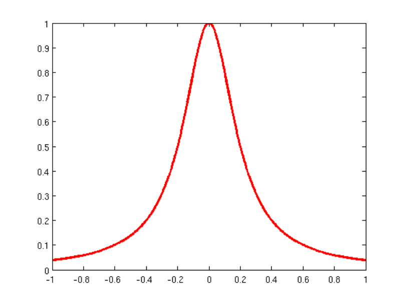
1. Form the Lagrange basis functions, $\ell_i(x)$ with $\ell_i(x_j) = \delta_{ij}$
2. Combine the Lagrange basis functions

$$\begin{aligned} p_2(x) &= y_0\ell_0(x) + y_1\ell_1(x) + y_2\ell_2(x) \\ &= (-1)\frac{(x-1)(x-2)}{2} + (-1)\frac{x(x-2)}{-1} + (7)\frac{x(x-1)}{2} \end{aligned}$$

Evaluate is *nice*, but *expensive*: no easy nested form.

how bad is polynomial interpolation?

Let's take something very smooth function



How does interpolation behave?

some analysis...

what can we say about

$$e(t) = f(t) - p_n(t)$$

at some point x? Consider $p = 1$: linear interpolation of a function at $x = x_0, x_1$

- want: error at x, $e(x)$
- look at

$$g(t) = e(t) - \frac{(t - x_0)(t - x_1)}{(x - x_0)(x - x_1)} e(x)$$

- $g(t)$ is 0 at $t = x_0, x_1, x$
- so $g'(t)$ is zero at two points, $g''(t)$ is zero at one point, call it c

$$0 = g''(c) = e''(t) - 2\frac{e(x)}{(x - x_0)(x - x_1)}$$

$$= f''(t) - 2\frac{e(x)}{(x - x_0)(x - x_1)}$$

$$e(x) = \frac{(x - x_0)(x - x_1)}{2} f''(c)$$

Theorem: Interpolation Error I

If $p_n(x)$ is the (at most) n degree polynomial interpolating $f(x)$ at $n+1$ distinct points and if $f^{(n+1)}$ is continuous, then

$$e(x) = f(x) - p_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(c) \prod_{i=0}^{n} (x - x_i)$$

Theorem: Bounding Lemma

Suppose x_i are equispaced in $[a, b]$ for $i = 0, \ldots, n$. Then

$$\prod_{i=0}^{n} |x - x_i| \leqslant \frac{h^{n+1}}{4} n!$$

Theorem: Interpolation Error II

Let $|f^{(n+1)}(x)| \leqslant M$, then with the above,

$$|f(x) - p_n(x)| \leqslant \frac{Mh^{n+1}}{4(n+1)}$$

fixes

We have two options:

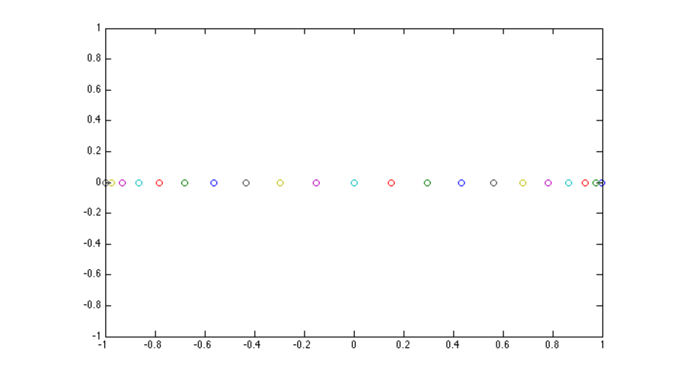
1. move the nodes: Chebychev nodes
2. piecewise polynomials (splines)

Option #1: Chebychev nodes in $[-1, 1]$

$$x_i = cos(\pi \frac{2i+1}{2n+2}), \quad i = 0, \dots, n$$

Option #2: piecewise polynomials...

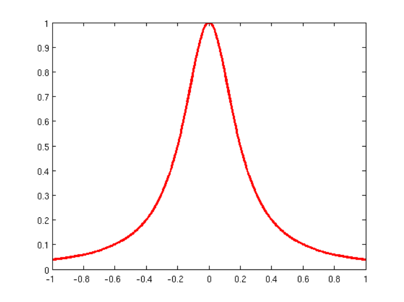
chebychev nodes



- Can obtain nodes from equidistant points on a circle projected down
- Nodes are non uniform and non nested

chebychev nodes

High degree polynomials using equispaced points suffer from many oscillations



- Points are bunched at the ends of the interval
- Error is distributed more evenly