Approximation theory

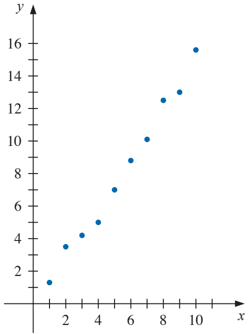
Xiaojing Ye, Math & Stat, Georgia State University

Spring 2019

Least squares approximation

Given N data points $\{(x_i, y_i)\}$ for $i = 1, \ldots, N$, can we determine a linear model $y = a_1 x + a_0$ (i.e., find a_0, a_1) that fits the data?

x_i	y_i	x_i	y_i
1	1.3	6	8.8
2	3.5	7	10.1
3	4.2	8	12.5
4	5.0	9	13.0
5	7.0	10	15.6



Matrix formulation

We can simplify notations by using matrices and vectors:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \quad X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} \in \mathbb{R}^{N\times 2}$$

So we want to find $a = (a_0, a_1)^\top \in \mathbb{R}^2$ such that $y \approx Xa$.

Several types of fitting criteria

There are several types of criteria for "best fitting":

- Define the error function as

$$E_\infty(a) = \|y - Xa\|_\infty$$

  and find $a^* = \arg\min_a E_\infty(a)$. This is also called the **minimax** problem since the problem $\min_a E_\infty(a)$ can be written as

$$\min_a \max_{1\le i\le n} |y_i - (a_0 + a_1 x_i)|$$

- Define the error function as

$$E_1(a) = \|y - Xa\|_1$$

  and find $a^* = \arg\min_a E_1(a)$. E_1 is also called the **absolute deviation**.

Least squares fitting

In this course, we focus on the widely used **least squares**.

Define the least squares error function as

$$E_2(a) = \|y - Xa\|_2 = \sum_{i=1}^{n} |y_i - (a_0 + a_1 x_i)|^2$$

and the least squares solution a^* is

$$a^* = \arg\min_a E_2(a)$$

Least squares fitting

To find the optimal parameter a, we need to solve

$$\nabla E_2(a) = 2X^\top (Xa - y) = 0$$

This is equivalent to the so-called **normal equation**:

$$X^\top Xa = X^\top y$$

Note that $X^\top X \in \mathbb{R}^{2\times 2}$ and $X^\top y \in \mathbb{R}^2$, so the normal equation is easy to solve!

Least squares fitting

It is easy to show that

$$X^\top X = \begin{bmatrix} N & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \end{bmatrix}, \quad X^\top y = \begin{bmatrix} \sum_{i=1}^N y_i \\ \sum_{i=1}^N x_i y_i \end{bmatrix}$$

Using the close-form of inverse of 2-by-2 matrix, we have

$$(X^\top X)^{-1} = \frac{1}{N\sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \begin{bmatrix} \sum_{i=1}^N x_i^2 & -\sum_{i=1}^N x_i \\ -\sum_{i=1}^N x_i & N \end{bmatrix}$$

Least squares fitting

Therefore we have the solution

$$a^* = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = (X^\top X)^{-1}(X^\top y)$$

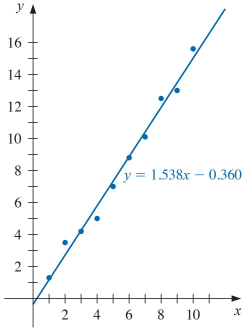
$$= \begin{bmatrix} \frac{\sum_{i=1}^N x_i^2 \sum_{i=1}^N y_i - \sum_{i=1}^N x_i y_i \sum_{i=1}^N x_i}{N \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \\ \frac{N \sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i}{N \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2} \end{bmatrix}$$

Least squares fitting

Example

Least squares fitting of the data gives $a_0 = -0.36$ and $a_1 = 1.538$.

x_i	y_i	x_i	y_i
1	1.3	6	8.8
2	3.5	7	10.1
3	4.2	8	12.5
4	5.0	9	13.0
5	7.0	10	15.6



Polynomial least squares

The least squares fitting presented above is also called **linear least squares** due to the linear model $y = a_0 + a_1 x$.

For general least squares fitting problems with data $\{(x_i, y_i) : i = 1, \dots, N\}$, we may use polynomial

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

as the fitting model. Note that $n = 1$ reduces to linear model.

Now the **polynomial least squares** error is defined by

$$E(a) = \sum_{i=1}^{N} |y_i - P_n(x_i)|^2$$

where $a = (a_0, a_1, \dots, a_n)^\top \in \mathbb{R}^{n+1}$.

Matrices in polynomial least squares fitting

Like before, we use matrices and vectors:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^n \end{bmatrix} \in \mathbb{R}^{N\times(n+1)}$$

So we want to find $a = (a_0, a_1, \dots, a_n)^\top \in \mathbb{R}^{n+1}$ such that $y \approx Xa$.

Polynomial least squares fitting

Same as above, we need to find a such that

$$\nabla E_2(a) = 2X^\top(Xa - y) = 0$$

which has **normal equation**:

$$X^\top X a = X^\top y$$

Note that now $X^\top X \in \mathbb{R}^{(n+1)\times(n+1)}$ and $X^\top y \in \mathbb{R}^{n+1}$. From normal equation we can solve for the fitting parameter

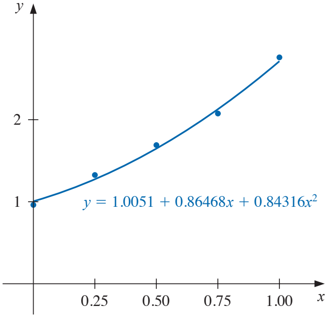
$$a^* = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = (X^\top X)^{-1}(X^\top y)$$

Polynomial least squares

Example

Least squares fitting of the data using $n = 2$ gives $a_0 = 1.0051$, $a_1 = 0.86468$, $a_2 = 0.84316$.

i	x_i	y_i
1	0	1.0000
2	0.25	1.2840
3	0.50	1.6487
4	0.75	2.1170
5	1.00	2.7183



Other least squares fitting models

In some situations, one may design model as

$$y = be^{ax}$$
$$y = bx^{a}$$

as well as many others.

To use least squares fitting, we note that they are equivalent to, respectively,

$$\log y = \log b + ax$$
$$\log y = \log b + a \log x$$

Therefore, we can first convert (x_i, y_i) to $(x_i, \log y_i)$ and $(\log x_i, \log y_i)$, and then apply standard linear least squares fitting.

Approximating functions

We now consider fitting (approximation) of a given function

$$f(x) \in C[a, b]$$

Suppose we use a polynomial $P_n(x)$ of degree n to fit $f(x)$, where

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

with fitting parameters $a = (a_0, a_1, \ldots, a_n)^\top \in \mathbb{R}^{n+1}$. Then the least squares error is

$$E(a) = \int_a^b |f(x) - P_n(x)|^2 \, \mathrm{d}x = \int_a^b \Big| f(x) - \sum_{k=0}^n a_k x^k \Big|^2 \, \mathrm{d}x$$

Approximating functions

The fitting parameter a needs to be solved from $\nabla E(a) = 0$.

To this end, we first rewrite $E(a)$ as

$$E(a) = \int_a^b (f(x))^2 \, \mathrm{d}x - 2 \sum_{k=0}^{n} a_k \int_a^b x^k f(x) \, \mathrm{d}x + \int_a^b \Big(\sum_{k=0}^{n} a_k x^k \Big)^2 \, \mathrm{d}x$$

Therefore $\nabla E(a) = (\frac{\partial E}{\partial a_0}, \frac{\partial E}{\partial a_1}, \dots, \frac{\partial E}{\partial a_n})^\top \in \mathbb{R}^{n+1}$ where

$$\frac{\partial E}{\partial a_j} = -2 \int_a^b x^j f(x) \, \mathrm{d}x + 2 \sum_{k=0}^{n} a_k \int_a^b x^{j+k} \, \mathrm{d}x$$

for $j = 0, 1, \dots, n$.

Approximating functions

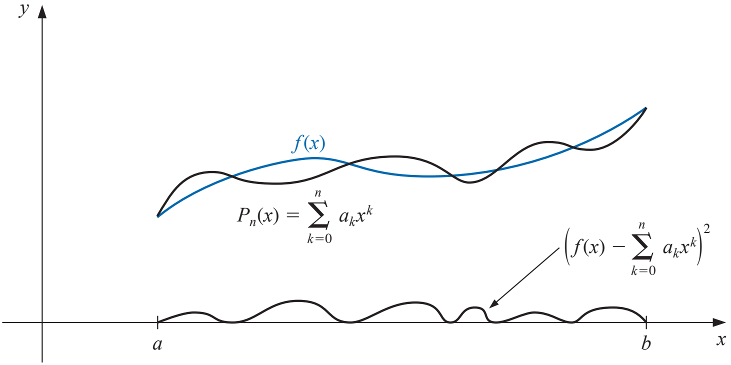
By setting $\frac{\partial E}{\partial a_j} = 0$ for all j, we obtain the **normal equation**

$$\sum_{k=0}^{n} \left(\int_a^b x^{j+k} \, \mathrm{d}x \right) a_k = \int_a^b x^j f(x) \, \mathrm{d}x$$

for $j = 0, \ldots, n$. This is a linear system of $n+1$ equations, from which we can solve for $a^* = (a_0, \ldots, a_n)^\top$.

Approximating functions

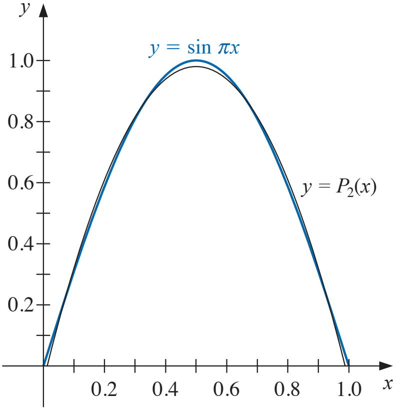
For the given function $f(x) \in C[a, b]$, we obtain least squares approximating polynomial $P_n(x)$:



Approximating functions

Example

Use least squares approximating polynomial of degree 2 for the function $f(x) = \sin(\pi x)$ on the interval $[0, 1]$.



Least squares approximations with polynomials

Remark

- *The matrix in the normal equation is called* **Hilbert matrix***, with entries of form*

$$\int_a^b x^{j+k}\,\mathrm{d}x = \frac{b^{j+k+1} - a^{j+k+1}}{j+k+1}$$

which is prune to round-off errors.

- *The parameters $a = (a_0, \dots, a_n)^\top$ we obtained for polynomial $P_n(x)$ cannot be used for $P_{n+1}(x)$ – we need to start the computations from beginning.*

Linearly independent functions

Definition

The set of functions $\{\phi_1, \dots, \phi_n\}$ *is called* **linearly independent** *on* $[a, b]$ *if*

$$c_1\phi_1(x) + c_2\phi_2(x) + \cdots + c_n\phi_n(x) = 0, \quad \text{for all } x \in [a, b]$$

implies that $c_1 = c_2 = \cdots = c_n = 0$.

Otherwise the set of functions is called **linearly dependent**.

Linearly independent functions

Example

Suppose $\phi_j(x)$ is a polynomial of degree j for $j = 0, 1, \ldots, n$, then $\{\phi_0, \ldots, \phi_n\}$ is linearly independent on any interval $[a, b]$.

Proof.

Suppose there exist $c_0, \ldots, c_n$ such that

$$c_0\phi_0(x) + \cdots + c_n\phi_n(x) = 0$$

for all $x \in [a, b]$. If $c_n \neq 0$, then this is a polynomial of degree n and can have at most n roots, contradiction. Hence $c_n = 0$. Repeat this to show that $c_0 = \cdots = c_n = 0$. $\square$

Linearly independent functions

Example

Suppose $\phi_0(x) = 2, \phi_1(x) = x - 3, \phi_2(x) = x^2 + 2x + 7$, and $Q(x) = a_0 + a_1 x + a_2 x^2$. Show that there exist constants c_0, c_1, c_2 such that $Q(x) = c_0\phi_0(x) + c_1\phi_1(x) + c_2\phi_2(x)$.

Solution: Substitute ϕ_j into $Q(x)$, and solve for c_0, c_1, c_2.

Linearly independent functions

We denote $\Pi_n = \{a_0 + a_1 x + \cdots + a_n x^n \mid a_0, a_1, \ldots, a_n \in \mathbb{R}\}$, i.e., Π_n is the set of polynomials of degree $\leq n$.

Theorem

Suppose $\{\phi_0, \ldots, \phi_n\}$ is a collection of linearly independent polynomials in Π_n, then any polynomial in Π_n can be written uniquely as a linear combination of $\phi_0(x), \ldots, \phi_n(x)$.

$\{\phi_0, \ldots, \phi_n\}$ is called a **basis** of Π_n.

Orthogonal functions

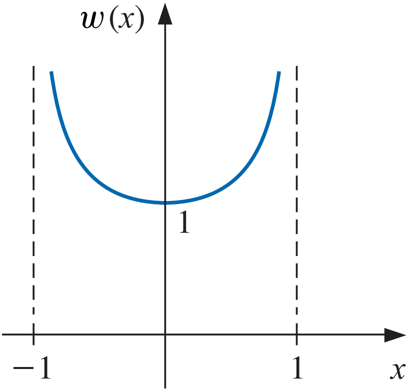
Definition

An integrable function w is called a **weight function** *on the interval I if $w(x) \geq 0$, for all $x \in I$, but $w(x) \not\equiv 0$ on any subinterval of I.*

Orthogonal functions

Example

Define a weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$ on interval $(-1, 1)$.



Orthogonal functions

Suppose $\{\phi_0, \dots, \phi_n\}$ is a set of linearly independent functions in $C[a, b]$ and w is a weight function on $[a, b]$. Given $f(x) \in C[a, b]$, we seek a linear combination

$$\sum_{k=0}^{n} a_k \phi_k(x)$$

to minimize the least squares error:

$$E(a) = \int_a^b w(x) \left[f(x) - \sum_{k=0}^{n} a_k \phi_k(x) \right]^2 \mathrm{d}x$$

where $a = (a_0, \dots, a_n)$.

Orthogonal functions

As before, we need to solve a^* from $\nabla E(a) = 0$:

$$\frac{\partial E}{\partial a_j} = \int_a^b w(x) \left[f(x) - \sum_{k=0}^{n} a_k \phi_k(x) \right] \phi_j(x) \, \mathrm{d}x = 0$$

for all $j = 0, \dots, n$. Then we obtain the normal equation

$$\sum_{k=0}^{n} \left(\int_a^b w(x) \phi_k(x) \phi_j(x) \, \mathrm{d}x \right) a_k = \int_a^b w(x) f(x) \phi_j(x) \, \mathrm{d}x$$

which is a linear system of $n + 1$ equations about $a = (a_0, \dots, a_n)^\top$.

Orthogonal functions

If we chose the basis $\{\phi_0, \ldots, \phi_n\}$ such that

$$\int_a^b w(x)\phi_k(x)\phi_j(x)\,\mathrm{d}x = \begin{cases} 0, & \text{when } j \neq k \\ \alpha_j, & \text{when } j = k \end{cases}$$

for some $\alpha_j > 0$, then the LHS of the normal equation simplifies to $\alpha_j a_j$. Hence we obtain closed form solution a_j:

$$a_j = \frac{1}{\alpha_j} \int_a^b w(x) f(x) \phi_j(x)\,\mathrm{d}x$$

for $j = 0, \ldots, n$.

Orthogonal functions

Definition

A set $\{\phi_0, \dots, \phi_n\}$ is called **orthogonal** *on the interval $[a, b]$ with respect to weight function w if*

$$\int_a^b w(x)\phi_k(x)\phi_j(x)\,\mathrm{d}x = \begin{cases} 0, & \text{when } j \neq k \\ \alpha_j, & \text{when } j = k \end{cases}$$

for some $\alpha_j > 0$ for all $j = 0, \dots, n$.

If in addition $\alpha_j = 1$ for all $j = 0, \dots, n$, then the set is called **orthonormal** *with respect to w.*

The definition above applies to general functions, but for now we focus on orthogonal/orthonormal polynomials only.

Gram-Schmidt process

Theorem

A set of orthogonal polynomials $\{\phi_0, \dots, \phi_n\}$ on $[a, b]$ with respect to weight function w can be constructed in the recursive way

- *First define*

$$\phi_0(x) = 1, \quad \phi_1(x) = x - \frac{\int_a^b xw(x)\,\mathrm{d}x}{\int_a^b w(x)\,\mathrm{d}x}$$

- *Then for every $k \geq 2$, define*

$$\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x)$$

where

$$B_k = \frac{\int_a^b xw(x)[\phi_{k-1}(x)]^2\,\mathrm{d}x}{\int_a^b w(x)[\phi_{k-1}(x)]^2\,\mathrm{d}x}, \ C_k = \frac{\int_a^b xw(x)\phi_{k-1}(x)\phi_{k-2}(x)\,\mathrm{d}x}{\int_a^b w(x)[\phi_{k-2}(x)]^2\,\mathrm{d}x}$$

Orthogonal polynomials

Corollary

Let $\{\phi_0, \dots, \phi_n\}$ *be constructed by the Gram-Schmidt process in the theorem above, then for any polynomial* $Q_k(x)$ *of degree* $k < n$*, there is*

$$\int_a^b w(x)\phi_n(x)Q_k(x)\,\mathrm{d}x = 0$$

Proof.

$Q_k(x)$ can be written as a linear combination of $\phi_0(x), \dots, \phi_k(x)$, which are all orthogonal to ϕ_n with respect to w. $\square$

Legendre polynomials

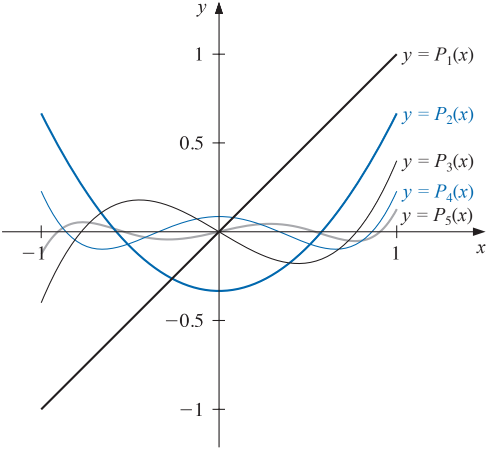
Using weight function $w(x) \equiv 1$ on $[-1, 1]$, we can construct **Legendre polynomials** using the recursive process above to get

$$\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x \\
P_2(x) &= x^2 - \frac{1}{3} \\
P_3(x) &= x^3 - \frac{3}{5}x \\
P_4(x) &= x^4 - \frac{6}{7}x^2 + \frac{3}{35} \\
P_5(x) &= x^5 - \frac{10}{9}x^3 + \frac{5}{21}x \\
&\vdots
\end{aligned}$$

Use the Gram-Schmidt process to construct them by yourself.

Legendre polynomials

The first few Legendre polynomials:



Chebyshev polynomials

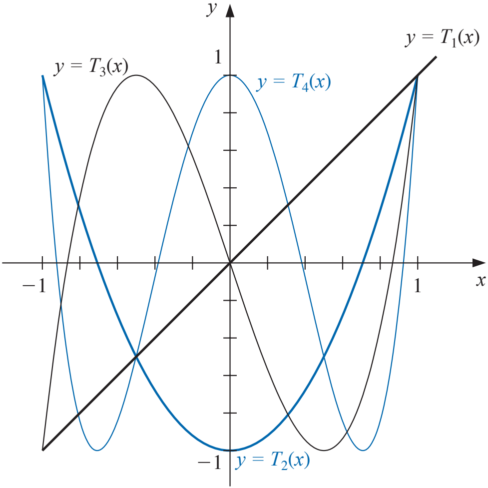
Using weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$ on $(-1, 1)$, we can construct **Chebyshev polynomials** using the recursive process above to get

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
&\vdots
\end{aligned}
$$

It can be shown that $T_n(x) = \cos(n \arccos x)$ for $n = 0, 1, \ldots$

Chebyshev polynomials

The first few Chebyshev polynomials:



Chebyshev polynomials

The Chebyshev polynomials $T_n(x)$ of degree $n \geq 1$ has n simple zeros in $[-1, 1]$ at

$$\bar{x}_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad \text{for each } k = 1, 2, \ldots, n$$

Moreover, T_n has maximum/minimum at

$$\bar{x}'_k = \cos\left(\frac{k\pi}{n}\right) \ \textit{where} \ T_n(\bar{x}'_k) = (-1)^k \text{ for each } k = 0, 1, 2, \ldots, n$$

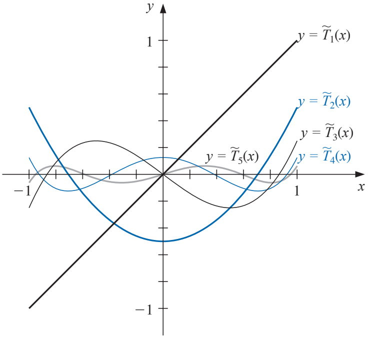
Therefore $T_n(x)$ has n distinct roots and $n+1$ extreme points on $[-1, 1]$. They are in order of min, zero, max, zero, min ...

Monic Chebyshev polynomials

The monic Chebyshev polynomials $\tilde{T}_n(x)$ are given by $\tilde{T}_0 = 1$ and

$$\tilde{T}_n = \frac{1}{2^{n-1}} T_n(x)$$

for $n \geq 1$.



Monic Chebyshev polynomials

The monic Chebyshev polynomials are

$$
\begin{aligned}
\tilde{T}_0(x) &= 1 \\
\tilde{T}_1(x) &= x \\
\tilde{T}_2(x) &= x^2 - \frac{1}{2} \\
\tilde{T}_3(x) &= x^3 - \frac{3}{4}x \\
\tilde{T}_4(x) &= x^4 - x^2 + \frac{1}{8} \\
&\vdots
\end{aligned}
$$

Monic Chebyshev polynomials

The monic Chebyshev polynomials $\tilde{T}_n(x)$ of degree $n \ge 1$ has n simple zeros in $[-1, 1]$ at

$$\bar{x}_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad \text{for each } k = 1, 2, \dots, n$$

Moreover, T_n has maximum/minimum at

$$\bar{x}'_k = \cos\left(\frac{k\pi}{n}\right) \text{ \textit{where} } T_n(\bar{x}'_k) = \frac{(-1)^k}{2^{n-1}}, \text{ for each } k = 1, 2, \dots, n$$

Therefore $\tilde{T}_n(x)$ also has n distinct roots and $n+1$ extreme points on $[-1, 1]$.

Monic Chebyshev polynomials

Denote $\tilde{\Pi}_n$ be the set of monic polynomials of degree n.

Theorem

For any $P_n \in \tilde{\Pi}_n$, there is

$$\frac{1}{2^{n-1}} = \max_{x\in[-1,1]} |\tilde{T}_n(x)| \le \max_{x\in[-1,1]} |P_n(x)|$$

The "$=$" holds only if $P_n \equiv \tilde{T}_n$.

Monic Chebyshev polynomials

Proof.

Assume not, then $\exists P_n(x) \in \tilde{\Pi}_n$, s.t. $\max_{x\in[-1,1]} |P_n(x)| < \frac{1}{2^{n-1}}$.

Let $Q(x) := \tilde{T}_n(x) - P_n(x)$. Since $\tilde{T}_n, P_n \in \tilde{\Pi}_n$, we know $Q(x)$ is a ploynomial of degree at most $n-1$. At the $n+1$ extreme points $\bar{x}'_k = \cos\left(\frac{k\pi}{n}\right)$ for $k = 0, 1, \dots, n$, there are

$$Q(\bar{x}'_k) = \tilde{T}_n(\bar{x}'_k) - P_n(\bar{x}'_k) = \frac{(-1)^k}{2^{n-1}} - P_n(\bar{x}'_k)$$

Hence $Q(\bar{x}'_k) > 0$ when k is even and < 0 when k odd. By intermediate value theorem, Q has at least n distinct roots, contradiction to $\deg(Q) \le n-1$. □

Minimizing Lagrange interpolation error

Let $x_0, \ldots, x_n$ be $n+1$ distinct points on $[-1,1]$ and $f(x) \in C^{n+1}[-1,1]$, recall that the Lagrange interpolating polynomial $P(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$ satisfies

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

for some $\xi(x) \in (-1,1)$ at every $x \in [-1,1]$.

We can control the size of $(x-x_0)(x-x_1)\cdots(x-x_n)$ since it belongs to $\tilde{\Pi}_{n+1}$: set $(x-x_0)(x-x_1)\cdots(x-x_n) = \tilde{T}_{n+1}(x)$. That is, set $x_k = \cos\left(\frac{2k-1}{2n}\pi\right)$, the kth root of $\tilde{T}_{n+1}(x)$ for $k = 1, \ldots, n+1$. This results in the minimal $\max_{x\in[-1,1]} |(x-x_0)(x-x_1)\cdots(x-x_n)| = \frac{1}{2^n}$.

Minimizing Lagrange interpolation error

Corollary

Let $P(x)$ be the Lagrange interpolating polynomial with $n+1$ points chosen as the roots of $\tilde{T}_{n+1}(x)$, there is

$$\max_{x\in[-1,1]} |f(x) - P(x)| \le \frac{1}{2^n(n+1)!} \max_{x\in[-1,1]} |f^{(n+1)}(x)|$$

Minimizing Lagrange interpolation error

If the interval of apporximation is on $[a, b]$ instead of $[-1, 1]$, we can apply change of variable

$$\tilde{x} = \frac{1}{2}[(b-a)x + (a+b)]$$

Hence, we can convert the roots $\bar{x}_k$ on $[-1, 1]$ to $\tilde{x}_k$ on $[a, b]$,

Minimizing Lagrange interpolation error

Example

Let $f(x) = xe^x$ on $[0, 1.5]$. Find the Lagrange interpolating polynomial using

1. the 4 equally spaced points $0, 0.5, 1, 1.5$.
2. the 4 points transformed from roots of $\tilde{T}_4$.

Minimizing Lagrange interpolation error

Solution: For each of the four points $x_0 = 0, x_1 = 0.5, x_2 = 1, x_3 = 1.5$, we obtain $L_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}$ for $i = 0, 1, 2, 3$:

$$
\begin{aligned}
L_0(x) &= -1.3333x^3 + 4.0000x^2 - 3.6667x + 1, \\
L_1(x) &= 4.0000x^3 - 10.000x^2 + 6.0000x, \\
L_2(x) &= -4.0000x^3 + 8.0000x^2 - 3.0000x, \\
L_3(x) &= 1.3333x^3 - 2.000x^2 + 0.66667x
\end{aligned}
$$

so the Lagrange interpolating polynomial is

$$
P_3(x) = \sum_{i=0}^{3} f(x_i) L_i(x) = 1.3875x^3 + 0.057570x^2 + 1.2730x.
$$

Minimizing Lagrange interpolation error

Solution: (cont.) The four roots of $\tilde{T}_4(x)$ on $[-1, 1]$ are $\bar{x}_k = \cos(\frac{2k-1}{8}\pi)$ for $k = 1, 2, 3, 4$. Shifting the points using $\tilde{x} = \frac{1}{2}(1.5x + 1.5)$, we obtain four points

$$\tilde{x}_0 = 1.44291, \tilde{x}_1 = 1.03701, \tilde{x}_2 = 0.46299, \tilde{x}_3 = 0.05709$$

with the same procedure as above to get $\tilde{L}_0, \dots, \tilde{L}_3$ using these 4 points, and then the Lagrange interpolating polynomial:

$$\tilde{P}_3(x) = 1.3811x^3 + 0.044652x^2 + 1.3031x - 0.014352.$$

Minimizing Lagrange interpolation error

Now compare the approximation accuracy of the two polynomials

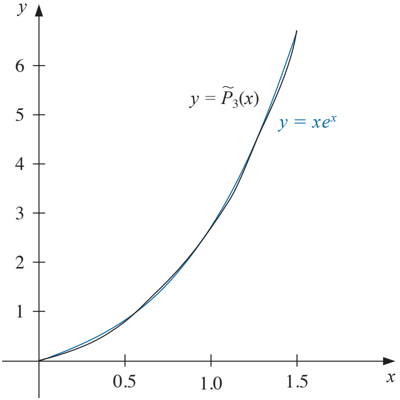
$$P_3(x) = 1.3875x^3 + 0.057570x^2 + 1.2730x$$
$$\tilde{P}_3(x) = 1.3811x^3 + 0.044652x^2 + 1.3031x - 0.014352$$

x	$f(x) = xe^x$	$P_3(x)$	$\lvert xe^x - P_3(x)\rvert$	$\tilde{P}_3(x)$	$\lvert xe^x - \tilde{P}_3(x)\rvert$
0.15	0.1743	0.1969	0.0226	0.1868	0.0125
0.25	0.3210	0.3435	0.0225	0.3358	0.0148
0.35	0.4967	0.5121	0.0154	0.5064	0.0097
0.65	1.245	1.233	0.012	1.231	0.014
0.75	1.588	1.572	0.016	1.571	0.017
0.85	1.989	1.976	0.013	1.974	0.015
1.15	3.632	3.650	0.018	3.644	0.012
1.25	4.363	4.391	0.028	4.382	0.019
1.35	5.208	5.237	0.029	5.224	0.016

Minimizing Lagrange interpolation error

The approximation using $\tilde{P}_3(x)$

$$\tilde{P}_3(x) = 1.3811x^3 + 0.044652x^2 + 1.3031x - 0.014352$$



Reducing the degree of approximating polynomials

As Chebyshev polynomials are efficient in approximating functions, we may use approximating polynomials of smaller degree for a given error tolerance.

For example, let $Q_n(x) = a_0 + \cdots + a_n x^n$ be a polynomial of degree n on $[-1, 1]$. Can we find a polynomial of degree $n - 1$ to approximate Q_n?

Reducing the degree of approximating polynomials

So our goal is to find $P_{n-1}(x) \in \Pi_{n-1}$ such that

$$\max_{x\in[-1,1]} |Q_n(x) - P_{n-1}(x)|$$

is minimized. Note that $\frac{1}{a_n}(Q_n(x) - P_{n-1}(x)) \in \tilde{\Pi}_n$, we know the best choice is $\frac{1}{a_n}(Q_n(x) - P_{n-1}(x)) = \tilde{T}_n(x)$, i.e., $P_{n-1} = Q_n - a_n \tilde{T}_n$. In this case, we have approximation error

$$\max_{x\in[-1,1]} |Q_n(x) - P_{n-1}(x)| = \max_{x\in[-1,1]} |a_n \tilde{T}_n| = \frac{|a_n|}{2^{n-1}}$$

Reducing the degree of approximating polynomials

Example

Recall that $Q_4(x)$ be the 4th Maclaurin polynomial of $f(x) = e^x$ about 0 on $[-1, 1]$. That is

$$Q_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}$$

which has $a_4 = \frac{1}{24}$ and truncation error

$$|R_4(x)| = |\frac{f^{(5)}(\xi(x))x^5}{5!}| = |\frac{e^{\xi(x)}x^5}{5!}| \le \frac{e}{5!} \approx 0.023$$

for $x \in (-1, 1)$. Given error tolerance 0.05, find the polynomial of small degree to approximate $f(x)$.

Reducing the degree of approximating polynomials

Solution: Let's first try Π_3. Note that $\tilde{T}_4(x) = x^4 - x^2 + \frac{1}{8}$, so we can set

$$\begin{aligned} P_3(x) &= Q_4(x) - a_4 \tilde{T}_4(x) \\ &= \left(1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}\right) - \frac{1}{24}\left(x^4 - x^2 + \frac{1}{8}\right) \\ &= \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3 \in \Pi_3 \end{aligned}$$

Therefore, the approximating error is bounded by

$$\begin{aligned} |f(x) - P_3(x)| &\le |f(x) - Q_4(x)| + |Q_4(x) - P_3(x)| \\ &\le 0.023 + \frac{|a_4|}{2^3} = 0.023 + \frac{1}{192} \le 0.0283. \end{aligned}$$

Reducing the degree of approximating polynomials

Solution: (cont.) We can further try Π_2. Then we need to approximate P_3 (note $a_3 = \frac{1}{6}$) above by the following $P_2 \in \Pi_2$:

$$\begin{aligned} P_2(x) &= P_3(x) - a_3 \tilde{T}_3(x) \\ &= \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3 - \frac{1}{6}\left(x^3 - \frac{3}{4}x\right) \\ &= \frac{191}{192} + \frac{9}{8}x + \frac{13}{24}x^2 \in \Pi_2 \end{aligned}$$

Therefore, the approximating error is bounded by

$$\begin{aligned} |f(x) - P_2(x)| &\le |f(x) - Q_4(x)| + |Q_4(x) - P_3(x)| + |P_3(x) - P_2(x)| \\ &\le 0.0283 + \frac{|a_3|}{2^2} = 0.0283 + \frac{1}{24} = 0.0703. \end{aligned}$$

Unfortunately this is larger than 0.05.

Reducing the degree of approximating polynomials

Although the error bound is larger than 0.05, the actual error is much smaller:

x	e^x	$P_4(x)$	$P_3(x)$	$P_2(x)$	$\lvert e^x - P_2(x)\rvert$
-0.75	0.47237	0.47412	0.47917	0.45573	0.01664
-0.25	0.77880	0.77881	0.77604	0.74740	0.03140
0.00	1.00000	1.00000	0.99479	0.99479	0.00521
0.25	1.28403	1.28402	1.28125	1.30990	0.02587
0.75	2.11700	2.11475	2.11979	2.14323	0.02623

Pros and cons of polynomial approxiamtion

Advantages:

- Polynomials can approximate continuous function to arbitrary accuracy;
- Polynomials are easy to evaluate;
- Derivatives and integrals are easy to compute.

Disadvantages:

- Significant oscillations;
- Large max absolute error in approximating;
- Not accurate when approximating discontinuous functions.

Rational function approximation

Rational function of degree $N = n + m$ is written as

$$r(x) = \frac{p(x)}{q(x)} = \frac{p_0 + p_1 x + \cdots + p_n x^n}{q_0 + q_1 x + \cdots + q_m x^m}$$

Now we try to approximate a function f on an interval containing 0 using $r(x)$.

WLOG, we set $q_0 = 1$, and will need to determine the $N + 1$ unknowns $p_0, \dots, p_n, q_1, \dots, q_m$.

Padé approximation

The idea of **Padé approximation** is to find $r(x)$ such that

$$f^{(k)}(0) = r^{(k)}(0), \quad k = 0, 1, \dots, N$$

This is an extension of Taylor series but in the rational form.

Denote the Maclaurin series expansion $f(x) = \sum_{i=0}^{\infty} a_i x^i$. Then

$$f(x) - r(x) = \frac{\sum_{i=0}^{\infty} a_i x^i \sum_{i=0}^{m} q_i x^i - \sum_{i=0}^{n} p_i x^i}{q(x)}$$

If we want $f^{(k)}(0) - r^{(k)}(0) = 0$ for $k = 0, \dots, N$, we need the numerator to have 0 as a root of multiplicity $N + 1$.

Padé approximation

This turns out to be equivalent to

$$\sum_{i=0}^{k} a_i q_{k-i} = p_k, \quad k = 0, 1, \ldots, N$$

for convenience we used convention $p_{n+1} = \cdots = p_N = 0$ and $q_{m+1} = \cdots = q_N = 0$.

From these $N+1$ equations, we can determine the $N+1$ unknowns:

$$p_0, p_1, \ldots, p_n, q_1, \ldots, q_m$$

Padé approximation

Example

Find the Padé approximation to e^{-x} of degree 5 with $n = 3$ and $m = 2$.

Solution: We first write the Maclaurin series of e^{-x} as

$$e^{-x} = 1 - x + \frac{1}{2}x^2 - \frac{1}{6}x^3 + \frac{1}{24}x^4 + \cdots = \sum_{i=0}^{\infty} \frac{(-1)^i}{i!} x^i$$

Then for $r(x) = \frac{p_0+p_1x+p_2x^2+p_3x^3}{1+q_1x+q_2x^2}$, we need

$$\left(1 - x + \frac{1}{2}x^2 - \frac{1}{6}x^3 + \cdots\right)(1+q_1x+q_2x^2) - p_0+p_1x+p_2x^2+p_3x^3$$

to have 0 coefficients for terms $1, x, \dots, x^5$.

Padé approximation

Solution: (cont.) By solving this, we get p_0, p_1, p_2, q_1, q_2 and hence

$$r(x) = \frac{1 - \frac{3}{5}x + \frac{3}{20}x^2 - \frac{1}{60}x^3}{1 + \frac{2}{5}x + \frac{1}{20}x^2}$$

x	e^{-x}	$P_5(x)$	$\lvert e^{-x} - P_5(x)\rvert$	$r(x)$	$\lvert e^{-x} - r(x)\rvert$
0.2	0.81873075	0.81873067	8.64×10^{-8}	0.81873075	7.55×10^{-9}
0.4	0.67032005	0.67031467	5.38×10^{-6}	0.67031963	4.11×10^{-7}
0.6	0.54881164	0.54875200	5.96×10^{-5}	0.54880763	4.00×10^{-6}
0.8	0.44932896	0.44900267	3.26×10^{-4}	0.44930966	1.93×10^{-5}
1.0	0.36787944	0.36666667	1.21×10^{-3}	0.36781609	6.33×10^{-5}

where $P_5(x)$ is Maclaurin polynomial of degree 5 for comparison.

Chebyshev rational function approximation

To obtain more uniformly accurate approximation, we can use Chebyshev polynomials $T_k(x)$ in Padé approximation framework.

For $N = n + m$, we use

$$r(x) = \frac{\sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

where $q_0 = 1$. Also write $f(x)$ using Chebyshev polynomials as

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$$

Chebyshev rational function approximation

Now we have

$$f(x) - r(x) = \frac{\sum_{k=0}^{\infty} a_k T_k(x) \sum_{k=0}^{m} q_k T_k(x) - \sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

We again seek for $p_0, \dots, p_n, q_1, \dots, q_m$ such that coefficients of $1, x, \dots, x^N$ are 0.

To that end, the computations can be simplified due to

$$T_i(x) T_j(x) = \frac{1}{2} \Big(T_{i+j}(x) + T_{|i-j|}(x) \Big)$$

Also note that we also need to compute Chebyshev series coefficients a_k first.

Chebyshev rational function approximation

Example

Approximate e^{-x} using the Chebyshev rational approximation of degree $n = 3$ and $m = 2$. The result is $r_T(x)$.

x	e^{-x}	$r(x)$	$\lvert e^{-x} - r(x)\rvert$	$r_T(x)$	$\lvert e^{-x} - r_T(x)\rvert$
0.2	0.81873075	0.81873075	7.55×10^{-9}	0.81872510	5.66×10^{-6}
0.4	0.67032005	0.67031963	4.11×10^{-7}	0.67031310	6.95×10^{-6}
0.6	0.54881164	0.54880763	4.00×10^{-6}	0.54881292	1.28×10^{-6}
0.8	0.44932896	0.44930966	1.93×10^{-5}	0.44933809	9.13×10^{-6}
1.0	0.36787944	0.36781609	6.33×10^{-5}	0.36787155	7.89×10^{-6}

where $r(x)$ is the standard Padé approximation shown earlier.

Trigonometric polynomial approximation

Recall the Fourier series uses a set of $2n$ orthogonal functions with respect to weight $w \equiv 1$ on $[-\pi, \pi]$:

$$\phi_0(x) = \frac{1}{2}$$
$$\phi_k(x) = \cos kx, \quad k = 1, 2, \dots, n$$
$$\phi_{n+k}(x) = \sin kx, \quad k = 1, 2, \dots, n-1$$

We denote the set of linear combinations of $\phi_0, \phi_1, \dots, \phi_{2n-1}$ by $\mathcal{T}_n$, called the set of trigonometric polynomials of degree $\leq n$.

Trigonometric polynomial approximation

For a function $f \in C[-\pi, \pi]$, we want to find $S_n \in \mathcal{T}_n$ of form

$$S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1} (a_k \cos kx + b_k \sin kx)$$

to minimize the least squares error

$$E(a_0, \dots, a_n, b_1, \dots, b_{n-1}) = \int_{-\pi}^{\pi} |f(x) - S_n(x)|^2 \, \mathrm{d}x$$

Due to orthogonality of Fourier series $\phi_0, \dots, \phi_{2n-1}$, we get

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx \, \mathrm{d}x, \quad b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx \, \mathrm{d}x$$

Trigonometric polynomial approximation

Example

Approximate $f(x) = |x|$ for $x \in [-\pi, \pi]$ using trigonometric polynomial from $\mathcal{T}_n$.

Solution: It is easy to check that $a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} |x| \, \mathrm{d}x = \pi$ and

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} |x| \cos kx \, \mathrm{d}x = \frac{2}{\pi k^2}((-1)^k - 1), \quad k = 1, 2, \dots, n$$

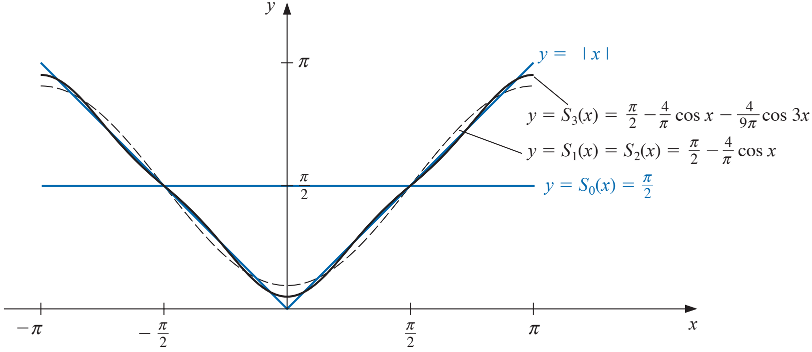
$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} |x| \sin kx \, \mathrm{d}x = 0, \quad k = 1, 2, \dots, n-1$$

Therefore

$$S_n(x) = \frac{\pi}{2} + \frac{2}{\pi}\sum_{k=1}^{n} \frac{(-1)^k - 1}{k^2} \cos kx$$

Trigonometric polynomial approximation

$S_n(x)$ for the first few n are shown below:



Discrete trigonometric approximation

If we have $2m$ paired data points $\{(x_j, y_j\}_{j=0}^{2m-1}$ where x_j are equally spaced on $[-\pi, \pi]$, i.e.,

$$x_j = -\pi + \left(\frac{j}{m}\right)\pi, \quad j = 0, 1, \dots, 2m-1$$

Then we can also seek for $S_n \in \mathcal{T}_n$ such that the discrete least square error below is minimized:

$$E(a_0, \dots, a_n, b_1, \dots, b_{n-1}) = \sum_{k=0}^{2m-1} (y_i - S_n(x_j))^2$$

Discrete trigonometric approximation

Theorem

Define

$$a_k = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \cos kx_j, \quad b_k = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \sin kx_j$$

Then the trigonometric $S_n \in \mathcal{T}_n$ defined by

$$S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1}(a_k \cos kx + b_k \sin kx)$$

minimizes the discrete least squares error

$$E(a_0, \ldots, a_n, b_1, \ldots, b_{n-1}) = \sum_{k=0}^{2m-1}(y_i - S_n(x_j))^2$$

Fast Fourier transforms

The **fast Fourier transform (FFT)** employs the Euler formula $e^{z\mathrm{i}} = \cos z + \mathrm{i} \sin z$ for all $z \in \mathbb{R}$ and $\mathrm{i} = \sqrt{-1}$, and compute the discrete Fourier transform of data to get

$$\frac{1}{m} \sum_{k=0}^{2m-1} c_k e^{kx\mathrm{i}}, \text{ where } c_k = \sum_{j=0}^{2m-1} y_j e^{k\pi \mathrm{i}/m} \;\; k = 0, \dots, 2m-1$$

Then one can recover $a_k, b_k \in \mathbb{R}$ from

$$a_k + \mathrm{i} b_k = \frac{(-1)^k}{m} c_k \in \mathbb{C}$$

Fast Fourier transforms

The discrete trigonometric approximation for $2m$ data points requires a total of $(2m)^2$ multiplications, not scalable for large m.

The cost of FFT is only

$$3m + m \log_2 m = O(m \log_2 m)$$

For example, if $m = 1024$, then $(2m)^2 \approx 4.2 \times 10^6$ and $3m + m \log_2 m \approx 1.3 \times 10^4$. The larger m is, the more benefit FFT gains.