Ch 8.2: Continuous Function Approximation, "L2"

Thursday, October 23, 2025 11:09 AM

Let's look at the 2nd scenario: rather than fitting $\hat{y}_i \approx y_i$ $i = 1, \dots, m$ we instead want to approximate a function f with another (simpler or nicer) function p, so that $f(x) \approx p(x)$ $\forall x \in [a,b]$

As in the discrete case, we have options for how to make $f(x) \approx p(x)$ precise:

① (default) L^2 $E = \|f-p\|_{L^2}^2 := \int_a^b |f(x)-p(x)|^2 dx$
(continuous) least squares
or just write "2"
a, b inferred from context

② minimax $E = \|f-p\|_\infty := \sup_{x \in [a,b]} |f(x)-p(x)|$

③ L^1 $E = \|f-p\|_1 := \int_a^b |f(x)-p(x)|\,dx$

We'll focus on ① L^2 as it's nicest in some ways

Aside: If $f \in C([a,b])$, $\forall \varepsilon > 0$ $\exists$ a polynomial p s.t. $\|f-p\|_\infty < \varepsilon$
"Weierstrauss Approximation Thm"
You can construct an explicit polynomial using Bernstein polynomials

What kind of function is $p(x)$?

We'll write $p(x) = \sum_{k=0}^{n} a_k \cdot \phi_k(x)$

for some set of linearly independent functions $\{\phi_k\}_{k=0}^{n}$
we discussed 2 weeks ago or so

While our loss function seems continuous or "∞-dimensional", what actually matters is that $n < \infty$. Our parameters we're looking for, is the vector $\vec{a} \in \mathbb{R}^{n+1}$, very finite dimensional.

Recall (from APPM 3310) the notion of an inner product
vector space
over $\mathbb{R}$ (not $\mathbb{C}$), it is any bilinear function $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$

s.t. ① $\forall f, g \in V$, $\langle f, g \rangle = \langle g, f \rangle$ Symmetric

② $\forall f, g, h \in V$, $\forall \alpha \in \mathbb{R}$, $\langle f + \alpha g, h \rangle = \langle f, h \rangle + \alpha \langle g, h \rangle$ linear in 1st argument

③ $\forall f \in V$, $\langle f, f \rangle \geq 0$ and $\langle f, f \rangle = 0$ iff $f = \vec{0}$

Thursday, October 23, 2025 11:27 AM

⚠ Notation warning: $\vec{a}, \vec{b}$ are vectors w/ entries a_j, b_j. Not to be confused with our domain $[a,b]$. Sorry.

and any inner product *induces* a norm, $\|f\| = \sqrt{\langle f, f\rangle}$

In our case, $\langle f, g\rangle := \int_a^b f(x)g(x)\,dx$ so $\|f\|_2 = \sqrt{\int_a^b f(x)^2\,dx}$

"L^2" inner product — "L^2" norm

So, $\|f-p\|_2^2 = \langle f-p, f-p\rangle$ (E)

$$= \langle f - \sum_{k=0}^{n} a_k \phi_k,\ f - \sum_{j=0}^{n} a_j \phi_j \rangle$$

$$= \langle f, f\rangle - 2\langle f, \sum_{j=0}^{n} a_j \phi_j\rangle + \langle \sum_{k=0}^{n} a_k \phi_k, \sum_{j=0}^{n} a_j \phi_j\rangle$$

$$= \|f\|_2^2 - 2\sum_{j=0}^{n} a_j \underbrace{\langle f, \phi_j\rangle}_{b_j} + \sum_{k=0}^{n}\sum_{j=0}^{n} a_k a_j \underbrace{\langle \phi_k, \phi_j\rangle}_{G_{kj}}$$

$$= \|f\|_2^2 - 2\vec{b}^T\vec{a} + \vec{a}^T G \vec{a}$$

G_{kj}: Gram matrix. Guaranteed to be positive definite if $\{\phi_j\}$ lin. independent

So $\min_{\vec{a}\in\mathbb{R}^{n+1}} E(\vec{a}) = \min_{\vec{a}\in\mathbb{R}^{n+1}} \|f\|_2^2 - 2\vec{b}^T\vec{a} + \vec{a}^T G\vec{a}$

So $\nabla E(\vec{a}) = -2\vec{b} + 2G\vec{a}$

solve $\nabla E(\vec{a}) = 0$ means solve $G\vec{a} = \vec{b}$ for $\vec{a}$.

Finite dimensional.

The "∞-dimensional part" is calculating $\vec{b}$, G in the 1st place: $b_j = \langle f, \phi_j\rangle = \int f(x)\phi_j(x)\,dx$

$$G_{kj} = \langle \phi_k, \phi_j\rangle = \int \phi_k(x)\phi_j(x)\,dx$$

This part you do by hand, not on a computer.

For a fixed basis $\{\phi_j\}_{j=0}^{n}$ and interval $[a,b]$, you can compute G (i.e. it doesn't depend on f)

Ex: $n=2$, $\phi_0(x)=1$, $\phi_1(x)=x$, $\phi_2(x)=x^2$ so $\text{span}(\{\phi_j\}) = \mathcal{P}_2$ ← polynomials of degree 2 or less

monomial basis

$$G_{kj} = \langle x^k, x^j\rangle = \int_a^b x^{k+j}\,dx = \frac{b^{j+k+1} - a^{j+k+1}}{j+k+1}$$

} the Hilbert matrix (notoriously ill-conditioned)

eg. $a=0, b=1$ then $G = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{pmatrix}$

then if we have a function f, we can solve.

Ex: $f(x) = \sin(\pi\cdot x)$. Let's say $a=0, b=1$

$$b_j = \int_0^1 \sin(\pi\cdot x)\cdot x^j\,dx = \begin{cases} 2/\pi & j=0 \\ 1/\pi & j=1 \\ \frac{\pi^2-4}{\pi^3} & j=2 \end{cases}$$

So $\vec{a} = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 1/2 & \ddots & 1/3 \\ 1/3 & & \end{pmatrix}^{-1} \begin{pmatrix} 2/\pi \\ 1/\pi \\ \frac{\pi^2-4}{\pi^3} \end{pmatrix}$.

Thursday, October 23, 2025 4:08 PM

Nicer bases

If we want $p \in \mathbb{P}_n$ (polynomials of degree $\le n$), any basis $\{\phi_k\}_{k=0}^n$ works in principle, but some are nicer than others, i.e. leading to faster + more stable computation.

In particular, if we have an orthogonal basis then $\langle \phi_k, \phi_j \rangle = 0$ if $k \neq j$

so G is a diagonal matrix (hence G^{-1} is easy)

What's an orthogonal basis for $\mathbb{P}_n$? let $[a,b] = [-1,1]$

well, $\{\phi_k(x) = x^k\}_{k=0}^n$ is a basis, so we can orthogonalize it via Gram-Schmidt:

$$\tilde{\phi}_0 = \phi_0 = 1$$

$$\tilde{\phi}_1 = \phi_1 - \frac{\langle \phi_1, \tilde{\phi}_0 \rangle}{\|\tilde{\phi}_0\|^2} \cdot \tilde{\phi}_0 = x$$

$$\tilde{\phi}_2 = \phi_2 - \frac{\langle \phi_2, \tilde{\phi}_0 \rangle}{\|\tilde{\phi}_0\|^2} \tilde{\phi}_0 - \frac{\langle \phi_2, \tilde{\phi}_1 \rangle}{\|\tilde{\phi}_1\|^2} \tilde{\phi}_1 = x^2 - \tfrac{1}{3}$$

etc.

$\{1, x, x^2 - \frac{1}{3}, \dots\}$ are the Legendre polynomials and orthogonal on $[-1,1]$

(Often normalized in different ways, eg. $\|\tilde{\phi}_k\| = 1$ or $\tilde{\phi}_k(1) = 1$ etc.)

Well-studied, easy to generate

In practice, use the 3-term recurrence:

$$\phi_{k+1}(x) = (x - \beta_k)\phi_k(x) - \gamma_k \phi_{k-1}(x)$$

(define $\phi_{-1} = 0$, $\phi_0 = 1$)

$$\beta_k = \frac{\langle x \cdot \phi_k, \phi_k \rangle}{\|\phi_k\|^2} \qquad \gamma_k = \frac{\|\phi_k\|^2}{\|\phi_{k-1}\|^2}$$

If $\{\phi_k\}$ orthogonal and letting $\alpha_k = \|\phi_k\|^2$, then our L^2 approximation of $f \in C[a,b]$ over $\text{span}(\{\phi_k\})$ is

$$p(x) = \sum_{j=0}^{n} a_j \phi_j(x) \quad \text{with} \quad a_j = \frac{\langle \phi_j, f \rangle}{\alpha_j}.$$

Thursday, October 23, 2025 4:25 PM

Standard Legendre polynomials are orthogonal on the domain $[-1,1]$

ex: $\phi_0(x)=1$, $\phi_1(x)=x$, $\int_{-1}^{1} 1\cdot x\,dx = 0$ ✓

If you want a domain $[a,b]$, you must adjust the Legendre polynomials accordingly (since, ex: $a=0, b=2$, $\int_0^2 1\cdot x\,dx \neq 0$)

You don't have to recompute, just adjust them:

$$(k\neq j) \qquad 0=\int_{-1}^{1}\phi_k(x)\phi_j(x)\,dx = \int_a^b \phi_k\left(\tfrac{2y}{b-a}-\tfrac{a+b}{b-a}\right)\phi_j\left(\tfrac{2y}{b-a}-\tfrac{a+b}{b-a}\right)\tfrac{2}{b-a}\,dy$$

change of variables: $y=\frac{b-a}{2}\cdot x+\frac{a+b}{2}$ } so $x=-1 \Rightarrow y=a$, $x=+1 \Rightarrow y=b$

so $dy=\frac{(b-a)}{2}dx$

i.e. $x=\frac{2y}{b-a}-\frac{a+b}{b-a}$

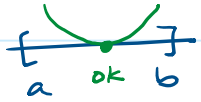
... so if $\tilde{\phi}_k(y)=\phi_k\left(\frac{2}{b-a}\cdot y+\frac{a+b}{b-a}\right)$ then $\{\phi_k\}$ orthog on $[-1,+]$ means $\{\tilde{\phi}_k\}$ orthog. on $[a,b]$.

SEE PG. 8 FOR MORE DETAILS (and Numpy info)

Weights and Chebyshev polynomials

We say $w(x)$ is a valid *weight function* on $[a,b]$ if it's integrable (a sufficient condition is being continuous) and it's non-negative $w(x)\geq 0$ and it's not 0 on any subinterval (

$w(x)=0$ at a single point or two is ok

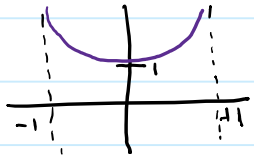


)

Then define $\langle f,g\rangle_w=\int_a^b f(x)g(x)\,w(x)\,dx$, $\|f\|_w=\sqrt{\langle f,f\rangle_w}$

and it's a valid inner product on $C[a,b]$ so all our theory still holds!

Main example $w(x)=\frac{1}{\sqrt{1-x^2}}$ on $[-1,1]$ Puts more weight near boundaries



Thursday, October 23, 2025 4:44 PM

The Legendre polynomials are orthogonal for $w(x) = 1$ but not orthogonal for $w(x) = \frac{1}{\sqrt{1-x^2}}$. What is?

The Chebyshev polynomials (of the 1st kind), $\{T_k\}$

2nd kind, U_k, use weight $\sqrt{1-x^2}$, less weight near edge.

Very well-studied

$T_0(x) = 1$ $\quad T_2(x) = 2x^2 - 1$

$T_1(x) = x$ $\quad T_3(x) = 4x^3 - 3x$

$T_{k+1}(x) = 2 \cdot x \cdot T_k(x) - T_{k-1}(x)$ "3-term recurrence"

T_k has even symmetry if k is even, odd symmetry if k is odd.

OR $\quad T_k(\underbrace{\cos\theta}_{x}) = \cos(k \cdot \theta) \qquad x = \cos^{-1}(\theta)$

i.e. $T_k(x) = \cos(k \cdot \cos^{-1}(x))$

so $k=0$, $T_0(x) = \cos(0) = 1$ ✓

$T_1(x) = \cos(\cos^{-1}(x)) = x$ ✓

To do a weighted least-squares fit ...

$\min_{\vec{a} \in \mathbb{R}^{n+1}} \| f - \sum_{k=0}^{n} a_k \phi_k \|_w^2$ we have $G \cdot \vec{a} = \vec{b}$ with

$G_{kj} = \langle \phi_k, \phi_j \rangle_w$, $b_j = \langle f, \phi_j \rangle_w$

So if $\{\phi_k\}$ is orthogonal on our domain, G is a diagonal matrix and $a_k = \dfrac{\int_a^b f(x)\phi_k(x)w(x)}{\int_a^b \phi_k^2(x)w(x)\,dx}$

Chebyshev polynomials are great

(Thm 8.9) Thm: The roots of $T_n(x)$ ($n \geq 1$ since T_0 has no roots) are all simple, in $[-1,1]$ and we have a formula for them: $r_k = \cos\left(\frac{2k-1}{2n} \cdot \pi\right)$, $k = 1, 2, \ldots, n$

proof

$T_n(x) = \cos(n \cdot \cos^{-1}(x))$

so $T_n(r_k) = \cos(n \cdot \cos^{-1}(r_k)) = \cos\left(n \cdot \frac{2k-1}{2n}\pi\right) = \cos\left((k-\tfrac{1}{2})\pi\right)$

$= \sin(k\pi) = 0$ since $k \in \mathbb{N}$

See book (§8.3) and web for more

Notation: $\tilde{T}_n(x) := \frac{1}{2^{n-1}} T_n(x)$ $(n \geq 1)$ so leading coefficient is 1,

i.e. $\tilde{T}_n$ is a monic polynomial

Thursday, October 23, 2025 5:11 PM

you can show Chebyshev nodes minimize bounds on interpolation error,
see § 8.3 Thm 8.10 + Corollary 8.11

More examples of orthogonal polynomials

- Suppose we want a domain $(0, \infty)$ not $[a,b]$.
  Can't use $w(x) = 1$ since most nice functions won't have a finite norm: $\int_0^\infty f(x)^2 dx = \infty$ is likely
  so usually choose $w(x) = e^{-x}$. Doing Gram-Schmidt on $\{1, x, x^2, \dots\}$ with this weight gives Laguerre polynomials
- Or domain $(-\infty, \infty)$, use $w(x) = e^{-x^2}$, get Hermite polynomials
- To generalize Chebyshev, take $[-1,1]$, let $w(x) = (1-x)^\alpha (1+x)^\beta$, get Jacobi polynomials, of which Chebyshev are a special case ($\alpha = \pm 1/2, \beta = \alpha$)
  as are Legendre ($\alpha = \beta = 0$)
- On $[a,b]$, do change of variables to get to $[-1,1]$

p. 7 (Bonus content, not covered in lecture nor on exam)

Friday, October 24, 2025 8:52 AM

Beyond L^2... L^∞: minimax approximation

Harder to approximate. Goal: find $p \in P_n$ to minimize $\|f-p\|_\infty$ $= \sup_{x\in[a,b]} |f(x)-p(x)|$

Use:

Thm (Chebyshev Equioscillation Thm)

Let $f \in C[a,b]$, and $n \geq 0$ an integer.

(1) There's a unique polynomial $q \in P_n$ s.t.

$$\|f-q\|_\infty = \inf_{p\in P_n} \|f-p\|_\infty \quad (\rho_n(f))$$

i.e. "inf" is a "min" and it's unique

(2) This q is the only polynomial in P_n for which there are at least $n+2$ distinct points $a \leq x_0 < x_1 < \dots < x_{n+1} \leq b$ s.t. $f(x_j) - q(x_j) = \sigma \cdot (-1)^j \cdot \rho_n(f)$ (oscillation)

where σ is either $+1$ or -1.

Suggests a method to find q: let $E(x) = f(x) - p(x)$, parameterize q by coefficients $\vec{a}$, then solve

$2n+4$ d.o.f.

$$\begin{cases} E(x_j) = (-1)^j \rho \\ E'(x_j) = 0 \end{cases} \quad j = 0,1,\dots,n+1$$

($2n+4$ eqn)

$E'(x_j)=0$ says it's a critical pt, i.e. max or min respectively

for unknowns $\vec{a}$ ($n+1$), ρ (1), $\{x_j\}$ ($n+2$)

Nonlinear system use, eg., Newton...

or...

Remez Exchange method:

w/ above equation, guess nodes $\{x_j\}$, then solve for $\vec{a}$ and ρ is linear. Do that, then find $n+2$ nodes where error is greatest, let these be next guess for nodes $\{x_j\}$, and iterate.

Extension to rational functions approximation is current research!

p. 8 More details on changing domains

Saturday, November 1, 2025 10:19 AM

MORE DETAILS ON CHANGING DOMAINS ... since it can be confusing

Recall:

If you want a domain $[a,b]$, you must adjust the Legendre polynomials accordingly (since, ex: $a=0, b=2$, $\int_0^2 1\cdot x\,dx \neq 0$, while $\int_{-1}^{1} 1\cdot x\,dx = 0$)

You don't have to recompute, just adjust them:

$(k\neq j)$ $$0=\int_{-1}^{1}\phi_k(x)\phi_j(x)dx=\int_a^b \phi_k\left(\tfrac{2y}{b-a}-\tfrac{a+b}{b-a}\right)\phi_j\left(\tfrac{2y}{b-a}-\tfrac{a+b}{b-a}\right)\tfrac{2}{b-a}dy$$

change of variables: $\boxed{y=\frac{b-a}{2}\cdot x+\frac{a+b}{2}}$ so $x=-1\Rightarrow y=a$, $x=+1\Rightarrow y=b$

So $dy=\frac{(b-a)}{2}dx$

i.e. $\boxed{x=\frac{2y}{b-a}-\frac{a+b}{b-a}}$ $x\in[-1,1]$, $y\in[a,b]$

We ignore $\frac{2}{b-a}$ scaling since won't affect orthog.

... so if $\tilde{\phi}_k(y)=\phi_k\left(\frac{2}{b-a}\cdot y+\frac{a+b}{b-a}\right)$ then $\{\phi_k\}$ orthog on $[-1,1]$ means $\{\tilde{\phi}_k\}$ orthog. on $[a,b]$.

Numpy does not add a $\sqrt{\frac{2}{b-a}}$ amplitude scaling.

In Numpy polynomial package, if you ever set "domain $=[a,b]$" then it interprets the basis to be $\{\tilde{\phi}_k\}$ instead of $\{\phi_k\}$

So if you have f on $[a,b]$ and want to find coeff. a_k s.t.

$f(y)=\sum a_k\tilde{\phi}_k(y)$ w/ $\tilde{\phi}_k(y)=\phi_k\left(\frac{2}{b-a}y+\frac{a+b}{b-a}\right)$, you have

two options:

(1) $\langle\tilde{\phi}_k, f\rangle = a_k\langle\tilde{\phi}_k,\tilde{\phi}_k\rangle$ where $\langle f,g\rangle:=\int_a^b f(y)g(y)dx$

i.e. $$a_k=\frac{\int_a^b f(y)\tilde{\phi}_k(y)dy}{\int_a^b\tilde{\phi}_k(y)^2dy}=\frac{\int_a^b f(y)\phi_k\left(\frac{2}{b-a}y+\frac{a+b}{b-a}\right)dy}{\int_a^b\phi_k\left(\frac{2}{b-a}y+\frac{a+b}{b-a}\right)^2dy}$$

"keep f on $[a,b]$, shift ϕ to $[a,b]$"

or

(2) let $g(x):=f\left(\frac{b-a}{2}x+\frac{a+b}{2}\right)$ w/ $x\in[-1,1]$ $\quad g(-1)=f(a)$, $g(+1)=f(b)$

write $g(x)=\sum a_k\phi_k(x)$ as usual,

$$a_k=\frac{\langle\phi_k,g\rangle}{\langle\phi_k,\phi_k\rangle}=\frac{\int_{-1}^{1}g(x)\phi_k(x)dx}{\int_{-1}^{1}\phi_k(x)^2dx}\Big\}=\int_{-1}^{1}f\left(\tfrac{b-a}{2}x+\tfrac{a+b}{2}\right)\phi_k(x)dx$$

"Shift f to $[-1,1]$, then shift back at end" so

$$f(y)=g\left(\frac{2}{b-a}y-\frac{a+b}{b-a}\right)=\sum_k a_k\phi_k\left(\frac{2}{b-a}y-\frac{a+b}{b-a}\right)=\sum_k a_k\tilde{\phi}_k(y)$$

In our (and numpy's) definition of $\tilde{\phi}$, it is not true $\int_a^b\tilde{\phi}_k(y)^2dy=\int_{-1}^{1}\phi_k(x)^2dx$ but that's ok