FUNCTION APPROXIMATION

QUANTITATIVE ECONOMICS 2024

Piotr Żoch

November 19, 2024

WHAT IS IT?

Goal: approximate a complicated function $f : \mathbb{R}^n \to \mathbb{R}$ by a simpler function $\hat{f} : \mathbb{R}^n \to \mathbb{R}$.

- We know f only at a finite number of points and we want to approximate it at other points x.
- f is too complicated to work with directly (e.g. non-analytic) and we need to represent on a computer.

I will talk only about the most basic ideas.

FUNCTION APPROXIMATION

- What data should be produced and used?
- What family of “simpler” functions should be used?
- What notion of approximation do we use?
- How good can the approximation be?
- How simple can a good approximation be?

Notice similarities and differences between function approximation and statistical regression.

NOTATION

- Today we will focus on continuous functions $f : \mathbb{R}^n \to \mathbb{R}$.
- We can represent every continuous function in a particular function space by a linear combination of basis functions.
- Analogy: Every vector in a vector space V can be represented by a linear combination of basis vectors.

NOTATION

- Let F by the space of continuous real-valued functions with domain $X \subset \mathbb{R}^n$.
- Define the inner product of two functions $f, g \in F$ as

$$\langle f, g \rangle = \int_X f(x) g(x) w(x) dx$$

  where $f, g, w \in F$, w is a weighting function.
- $\{F, \langle \cdot, \cdot \rangle\}$ is an inner product space.
- We want to approximate a known function $f : X \to \mathbb{R}$ in $\{F, \langle \cdot, \cdot \rangle\}$.

NOTATION

- Let $\hat{f}(\cdot, \beta)$ be a parametric approximation of f. We have

$$\hat{f}(x, \beta) = \sum_{j=0}^{J} \beta_j \phi_j(x)$$

  - $\phi_j(x)$ are basis functions. Write $\Phi_J = \{\phi_0, \phi_1, \ldots, \phi_J\}$.
  - $\beta = [\beta_0, \beta_1, \ldots, \beta_J]$ is a vector of coefficients.
  - J is the order of interpolation.
- We want to find β such that $\hat{f}(\cdot, \beta)$ is a "good" approximation of f.
- Define the residual function $r(x, \beta) f(x) - \hat{f}(x, \beta)$. We want to make it small in some sense.

SPECTRAL METHODS

SPECTRAL METHODS

- Spectral methods: basis functions are non-zero on the entire domain of f.
- Polynomial interpolation: basis functions are polynomials
- Fourier series: basis functions are sines and cosines

INTERPOLATION

- Suppose we know f at $N = J + 1$ points $\{x_i\}_{i=0}^{J}$. We call these points interpolation nodes.
- Note: we have the same number of points as basis functions.
- Let r_i be the residual at x_i. We have

$$\begin{bmatrix} r_0 \\ \vdots \\ r_J \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_0) \end{bmatrix} - \begin{bmatrix} \phi_0(x_0) & \cdots & \phi_0(x_J) \\ \vdots & \ddots & \vdots \\ \phi_J(x_0) & \cdots & \phi_J(x_J) \end{bmatrix} \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_J \end{bmatrix}$$

- Abuse notation and write it as

$$r = f - \Phi\beta$$

.

INTERPOLATION

- The idea of interpolation is to find β such that $r_i = 0$ for all i.
- The unknowns are the coefficients β.
- This is equivalent to solving the linear system of equations

$$\Phi\beta = f$$

with an obvious solution $\beta = \Phi^{-1}f$.

REGRESSION

- If we have more points than basis functions, $M > J + 1$ we cannot interpolate.
- We have to define a loss function $L(\cdot, r)$ and minimize it.
- If we use the squared loss function

$$L(x, r) = \sum_{i=0}^{M} r(x_i, \beta)^2,$$

we get the least squares problem with solution

$$\beta = \left(\Phi^T \Phi\right)^{-1} \Phi^T f$$

WEIERSTRASS THEOREM

Theorem

Let $f : [a, b] \to \mathbb{R}$ be a continuous function. Then for every $\varepsilon > 0$ there exists a polynomial p such that

$$\sup_{x \in [a,b]} |f(x) - p(x)| \leq \varepsilon.$$

- We can approximate any continuous function on a compact set by a polynomial as closely as we want.

BASIS FUNCTION CHOICE

- Which basis functions should we use?
- The most natural choice seems to be monomials: $\phi_j(x) = x^j$.
- Problem: consecutive monomials are very similar to each other – does x^{10} really add much to x^9?
- The resulting matrix Φ is a Vandermonde matrix, which is ill-conditioned.

BASIS FUNCTION CHOICE

- If Φ looks like a diagonal matrix, then Φ^{-1} is easy to compute.
- Intiuition: use basis functions that give us different information about f.
- Orthogonal polynomials are a good choice: $\langle \phi_i, \phi_j \rangle = 0$ for $i \neq j$.

ORTHOGONAL POLYNOMIALS

- For orthogonal polynomials, we have

$$\beta_j = \int_X f(x)\phi_j(x)w(x)dx$$

- Intiuition: use basis functions that give us different information about f.
- Orthogonal polynomials are a good choice: $\langle \phi_i, \phi_j \rangle = 0$ for $i \neq j$.
- Examples: Legendre, Chebyshev, Hermite, Laguerre, Jacobi polynomials.

CHEBYSHEV POLYNOMIALS

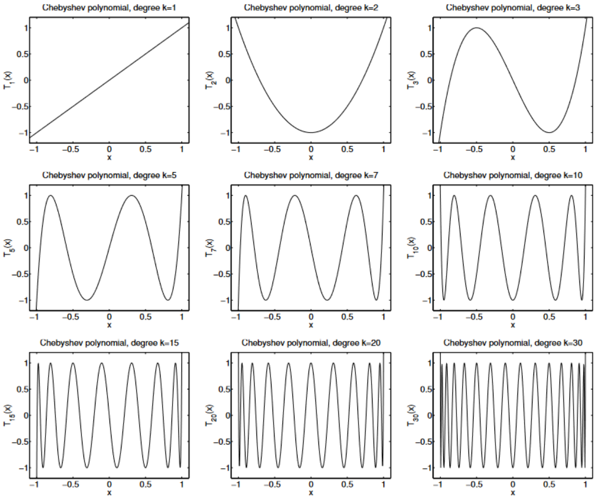
- Chebyshev polynomials $T_n(x) : [-1, 1] \to \mathbb{R}$ are given by

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x)$$

- Let $X = [-1, 1]$ and $w(x) = \frac{1}{\sqrt{1-x^2}}$. We have

$$\langle T_i, T_j \rangle = \int_{-1}^{1} T_i(x) T_j(x) \frac{1}{\sqrt{1-x^2}} dx = \begin{cases} 0 & i \neq j \\ \pi & i = j = 0 \\ \frac{\pi}{2} & i = j \neq 0 \end{cases}$$

CHEBYSHEV POLYNOMIALS



CHEBYSHEV NODES

- Orthogonality is not the only nice property of Chebyshev polynomials.
- Chebyshev nodes are roots of Chebyshev polynomials on $[-1, 1]$:

$$x_i = \cos\left(\frac{2i-1}{2n}\pi\right) \quad \text{for} \quad i = 1, \ldots, n.$$

  It can be verified that T_n equals 0 at these points.
- Chebyshev nodes are not equally spaced, they are clustered at the endpoints of the interval.

CHEBYSHEV NODES

- In practice we want to work on $[a, b]$. We can use an affine transformation to get

$$x_i = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{2i-1}{2n}\pi\right) \quad \text{for} \quad i = 1, \ldots, n.$$

NODES CHOICE

- Why are these nodes useful?
- We know that at the interpolation nodes x_i we have $r_i = 0$.
- We want to make the residual as small as possible at other points x.
- A silly choice of nodes (e.g. equidistant) can lead to a large residual at other points, even with a high order of interpolation – Runge phenomenon.
- Minmax approximation: polynomial approximation using Chebyshev nodes is very close to the polynomial approximation that minimizes the maximum absolute error on $[-1, 1]$.

CHEBYSHEV REGRESSION

1. Obtain $M \geq J + 1$ Chebyshev nodes z_m for $m = 0, \dots, M$ on $[-1, 1]$.
2. Transform the nodes to $[a, b]$:

$$x_m = (z_m + 1)\,\frac{b - a}{2} \quad \text{for } m = 0, \dots, M.$$

3. Evaluate f at x_m to get f_m.
4. Compute β_j for $j = 0, \dots, J$ using the least squares formula:

$$\beta_j = \frac{\sum_{m=0}^{M} f_m T_j(z_m)}{\sum_{m=0}^{M} T_j(z_m)^2} \text{ for } j = 0, \dots, J.$$

   to get the approximation of $f(x)$ on $[a, b]$

$$\hat{f}(x) = \sum_{j=0}^{J} \beta_j T_j\left(2\frac{x - a}{b - a} - 1\right).$$

BOYD (2000)

- When in doubt, use Chebyshev polynomials unless the solution is spatially periodic, in which case an ordinary Fourier series is better.
- Unless you’re sure another set of basis functions is better, use Chebyshev polynomials.
- Unless you’re really, really sure that another set of basis functions is better, use Chebyshev polynomials.

WARNING

- Beware! Potentially big problems if you want to evaluate $\hat{f}$ outside of $[a, b]$!
- Extrapolation with Chebyshev polynomials is very bad.
- Jesus Fernandez-Villaverde once said: *Chebyshev polynomials are like a Downton Abbey set. Everything in the frame is so beautiful, but you move a little bit and it's total chaos.*
- In economics we often want to evaluate functions outside of the domain of the data – be careful.

FINITE ELEMENTS METHOD

SPLINES

- Splines are piecewise polynomials.
- Idea: approximate function on many intervals, on each interval by a separate polynomial. Then "glue" the polynomials together.
- Flexible: use only local information about f. You can have polynomials of different orders on different intervals.
- Compare with spectral methods – there "one size fits all".

SPLINES

- Let z be a knot vector of length b. We want to approximate f on $[a, b]$ so $z_1 = a, z_p = b$.
- Elements of z are in an ascending order, $z_1 < z_2 < \ldots < z_p$.
- Knots divide $[a, b]$ into $p - 1$ intervals.
- On each of these interval use a different polynomial.
- "Glue" these polynomials: require that the resulting approximation is continuous and (perhaps) smooth at the knots.

SPLINES

- For simplicity assume that all polynomial are of the same order k, they have $k + 1$ coefficients.
- In total there is $(p - 1)(k + 1)$ coefficients.
- There is $p - 2$ interior knots.
- Require that the resulting approximation is continuous and $k - 1$ times differentiable at the interior knots: this is $k(p - 2)$ conditions.
- We are left with $N = p + k - 1$ free parameters. We can write $\hat{f}(x, \beta)$ as a linear combination of N basis functions.

B-SPLINES

- We usually use B-splines (Basis splines).
- Denote the j-th B-spline of order k by $B_{j,k}(x)$. B-splines are defined recursively:

$$B_{j,0}(x) = \begin{cases} 1 & \text{if } z_j \leq x < z_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

and

$$B_{j,k}(x) = \frac{x - z_j}{z_j - z_{j-k}} B_{j-1,k-1}(x) + \frac{z_{j+1} - x}{z_{j+1} - z_{j+1-k}} B_{j,k-1}(x)$$

with $z_j = z_1$ for $j < 1$, $z_j = z_p$ for $j > p$ and $B_{0,k-1}(x) = B_{p,k-1}(x) = 0$.

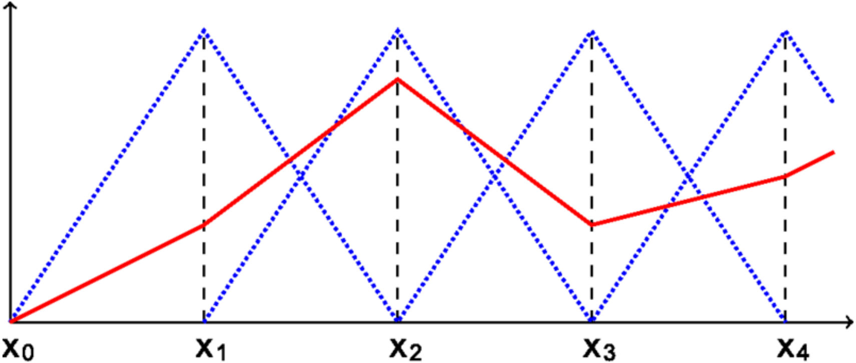
B-SPLINES

- $B_{j,0}$ are step functions - equal 1 on the interval $\left[z_j, z_{j+1}\right]$ and 0 otherwise.
- To get more intuition suppose the grid is uniform (knots are equidistant): $z_j - z_{j-1} = d$.
- In this case $B_{j,1}$ becomes

$$B_{j,1}(x) = \begin{cases} 1 - \frac{\left|x - z_j\right|}{d} & \text{if } \left|x - z_j\right| < d \\ 0 & \text{otherwise} \end{cases}$$

so it is a tent function.

LINEAR B-SPLINES



LINEAR B-SPLINES

- Linear B-splines are piecewise linear functions.
- They are continuous but not differentiable at the knots.
- This is simply connecting points with straight lines!
- Easy to implement and shape-preserving. Easy to evaluate.