Computational Statistics Nov 3, 2021

Quadrature

A quadrature is a set of nodes x_j and weights w_j used for approximating an integral, for example:

$$C_n = \int_{-1}^{1} f(x)\, q_n(x)\, dx$$

so that $$f(x) \approx \sum_{n=1}^{p} C_n q_n(x)$$

The approximation then takes the form

$$C_n = \int_{-1}^{1} f(x)\, q_n(x)\, dx \approx \sum_{1}^{M} f(x_j)\, q_n(x_j)\, w_j$$

may be = under certain circumstances.

There are lots of integrals in prob & stats:

- posterior distributions
- CDFs
- ...

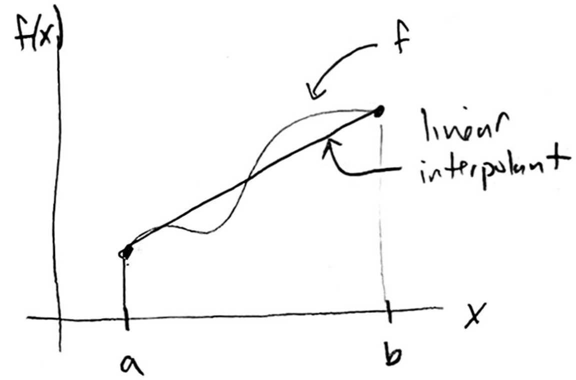
From a practical point of view, you only need two quadrature rules for smooth functions:

- trapezoidal rule
- Gaussian quadrature.

Trapezoidal Quadrature

One special case of what are known as Newton-Cotes Formulas:

interpolate the function, integrate the interpolant



$$\underbrace{\int_a^b f(x)\,dx \approx T[f] = \frac{b-a}{2}\left(f(a)+f(b)\right)}$$

This idea can be done to arbitrarily high order (interpolation, that is)

What is the error?

Write $f(x) = f\left(\frac{a+b}{2}\right) + f'\left(\frac{a+b}{2}\right)\left(x-\frac{a+b}{2}\right) + f''\left(\frac{a+b}{2}\right)\left(x-\frac{a+b}{2}\right)^2 + \ldots$

$$\int f = \int \left(f\left(\tfrac{a+b}{2}\right) + \ldots + \right)$$

$$\left[\frac{1}{3}\left(x-\frac{a+b}{2}\right)^3\right]_a^b = \frac{1}{3}\left[\left(\frac{b-a}{2}\right)^3 - \left(\frac{a-b}{2}\right)^3\right] = \frac{(b-a)^3}{3}$$

$$T[f] = T\left[\left(f\left(\tfrac{a+b}{2}\right) + \ldots\right)\right]$$

$$\int \left(f\left(\tfrac{a+b}{2}\right) + \ldots \right) - T\left[\left(f\left(\tfrac{a+b}{2}\right) + \ldots\right)\right] = f''\left(\frac{a+b}{2}\right)\overbrace{\int_a^b \left(x-\frac{a+b}{2}\right)^2 dx} + f^{(3)}\left(\frac{a+b}{2}\right)\int\left(x-\frac{a+b}{2}\right)^3 dx$$

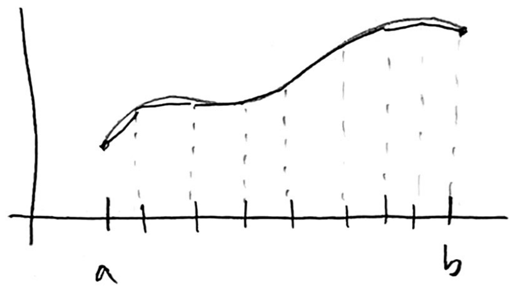
$$- f''\left(\frac{a+b}{2}\right)\frac{b-a}{2}\left[\left(\frac{a-b}{2}\right)^2 + \left(\frac{b-a}{2}\right)^2\right] + \ldots$$

$$\approx f''\left(\frac{a+b}{2}\right)\left(\frac{(b-a)^3}{3} - \frac{(b-a)^3}{4}\right)$$

$$\approx \boxed{f''\left(\frac{a+b}{2}\right)\frac{(b-a)^3}{12}}$$

This error is not great, $b-a$ has to be very small...

Obvious choice: split $[a,b]$ into many intervals of size $h = \frac{b-a}{n}$: Composite Trapezoidal rule



Then

$$\int_a^b f(x)\,dx \approx T_n[f]$$
$$= \sum_{i=0}^{n-1} \frac{(b-a)}{n}\left(\frac{f(x_i) + f(x_{i+1})}{2}\right)$$

where $x_i = a + hi$

Simplifying, we get

$$T_n[f] = h\left(\frac{f(x_0)+f(x_1)}{2} + \frac{f(x_1)+f(x_2)}{2} + \cdots + \frac{f(x_{n-1})+f(x_n)}{2}\right)$$

$$= h\left(\frac{f(a)}{2} + f(x_1) + \cdots + f(x_{n-1}) + \frac{f(b)}{2}\right)$$

$$= h\sum_{i=0}^{n} f(x_i) - \left(\frac{f(a)}{2} + \frac{f(b)}{2}\right)$$

The error on each subinterval is $O(h^3) = O\left(\frac{1}{n^3}\right)$ and since there are $O(n)$ intervals, the overall error of $T_n[f]$ is $O\left(\frac{1}{n^2}\right) = O(h^2)$.

There is a theorem that completely characterizes this error:

The Euler-MacLaurin Expansion

Theorem: Let $f \in C^{2k}[a,b]$, and $[a,b]$ be divided into n equal subintervals, $[x_{j-1}, x_j]$, with $x_j = a + jh$.

Then, $\int_a^b f(x)\,dx - T_n f$

I

$$= \sum_{r=1}^{k} c_r h^{2r} \left(f^{(2r-1)}(b) - f^{(2r-1)}(a) \right) - d_{2k} \left(\frac{h}{2}\right)^{2k}.$$

$$= \frac{h^2}{12}\left(f'(b) - f'(a) \right) - \frac{h^4}{720}\left(f^{(3)}(b) - f^{(3)}(a) \right) + \dots + (-1)^{k-1} \frac{B_{2n}}{(2k)!} h^{2k} f^{(2k)}(\xi)$$

$\xi \in [a,b]$

These coefficients are given by:

$$c_r = -\frac{B_{2r}}{(2r)!}, \quad B_{2r} \text{ is a Bernoulli number:}$$

$$\frac{x}{2}\coth\left(\frac{x}{2}\right) = \sum_{r=0}^{\infty} \frac{B_{2r}}{(2r)!} x^{2r}.$$

The calculation of B_{2r} was the output of arguably the first "computer program" written by Ada Lovelace and Charles Babbage.

Implications of Euler-MacLaurin

If $f \in C^{\infty}[a,b]$ and periodic with $f^{(j)}(a) = f^{(j)}(b)$ (think Fourier Series, or $\cos mx$, $\sin mx$, etc.) then this error $|I - T_n f|$ decays superalgebraically as $n \to \infty$.

Def: $\epsilon_n \to 0$ superalgebraically if

$$\lim_{n\to\infty} \frac{\epsilon_n}{h^{p \cdot n}} = 0 \quad \text{for } \underline{\text{any}} \ p > 0.$$

This means that $\epsilon_n \to 0$ faster than any power of h.

For this reason, the trapezoidal rule is very important important in various numerical methods.

Ex: $$J_0(x) = \frac{1}{\pi}\int_0^{\pi} e^{ix\cos\theta}\, d\theta.$$

$e^{ix\cos\theta}$ is not periodic on $[0,\pi]$, but it is on $[0,2\pi]$.

It can be shown that

$$J_0(x) = \frac{1}{\pi}\int_0^{\pi} e^{ix\cos\theta}\, d\theta = \underbrace{\frac{1}{2\pi}\int_0^{2\pi} e^{ix\cos\theta}\, d\theta}.$$

See mybessel.m for computing this via the trapezoidal rule.

The composite trapezoidal rule cannot be beat for integrating smooth periodic functions. What is a "high-order" alternative for non-periodic functions?

Gaussian Quadrature

Suppose we want to numerically compute $\int_{-1}^{1} f(x)\,dx$.

Let x_i be the i^{th} root of the degree n Legendre polynomial P_n. Compute the corresponding weights w_i such that

$$\int_{-1}^{1} x^j\,dx = \sum_{i=1}^{n} w_i x_i^j \quad \text{is } \underline{\text{exact}} \quad \text{if } j=0,\dots,n-1.$$

The collection of $\{x_i, w_i\}$ is known as a <u>Gaussian Quadrature</u> and is <u>in fact</u> exact for polynomials of degree $\leq 2n-1$.

<u>Proof</u> Let f be a polynomial of degree $\leq 2n-1$. Then divide f by P_n.

$$\Rightarrow \quad f(x) = \underbrace{q(x)}_{\deg \leq n-1} \underbrace{P_n(x)}_{\deg n} + \underbrace{r(x)}_{\deg \leq n-1} \qquad \left(\text{polynomial remainder theorem}\right)$$

Then $\int f(x)\,dx = \int (qP_n + r) = (q, P_n) + \int r = 0 + \int r$

And $G_n[f] = \sum_{i=1}^{n} \left(q(x_i) P_n(x_i) + r(x_i)\right) w_i$

$= \sum_{i=1}^{n} w_i\, r(w_i) = \int r(x)\,dx$

Exact.

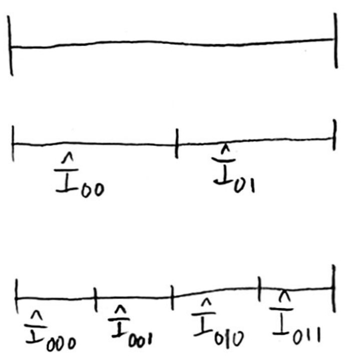
(Mention error...)

Adaptive Quadrature

While composite Gaussian quadrature is usefull, more efficient "adaptive" methods are quite practical.

Goal: Compute $\int_{-1}^{1} f(x)\,dx = I$ to precision ϵ using Gaussian quad. Call $\hat{I}$ our approximation to I: want $|\hat{I} - I| < \epsilon$.

<u>Idea</u>: Split intervals and check values:



$\hat{I}_0$

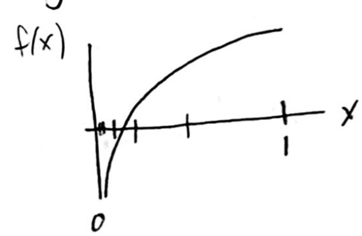
Check error: $|\hat{I}_0 - (\hat{I}_{00} + \hat{I}_{01})|$

Check <u>each</u> integrand: $|\hat{I}_{00} - (\hat{I}_{000} + \hat{I})|$

$|\hat{I}_{01} - (\hat{I}_{010} + \hat{I}_{011})|$

Only subdivide intervals on which requested precision has not been achieved (Mention possible accumulation of error.)

Behavior for singular integration:



$f(x) = \log x$

Algorithm will subdivide near the singularity.

(Briefly discuss error... complicated estimates.)