An Introduction to Numerical Analysis

Ping Li

Department of Statistics and Biostatistics

Department of Computer Science

Rutgers University, the State University of New Jersey

Pisctaway, NJ 08854, USA

Taylor Approximation

A function $f(x)$ in the neighborhood of a can be approximated by

$$f(x) = P_n(x) + R_n(x)$$

where $P_n(x)$ is an nth-order polynomial and $R_n(x)$ is the error term.

$$\begin{aligned} P_n(x) =& f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 \\ &+ ... + \frac{f^{(n)}(a)}{n!}(x-a)^n \end{aligned}$$

$$R_n(x) = \frac{f^{(n+1)}(z)}{(n+1)!}(x-a)^{n+1}, \qquad \text{where } z \text{ is between } x \text{ and } a.$$

If we choose $a = 0$, then in the neighborhood of 0, we have

$$P_n(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + ... + \frac{f^{(n)}(0)}{n!}x^n$$

$$R_n(x) = \frac{f^{(n+1)}(z)}{(n+1)!}x^{n+1}, \qquad \text{where } z \text{ is between } x \text{ and } 0.$$

When $n = 0$, the Taylor approximation becomes

$$f(x) = f(a) + f'(z)(x-a), \text{ or } f'(z) = \frac{f(x) - f(a)}{x - a}, \quad \text{where } z \text{ is between } x \text{ and } a$$

In other words, there must be a point z at which the slope $f'(z)$ is the same as $\frac{f(x)-f(a)}{x-a}$.
This is also known as the “mean-value theorem” and can be easily visualized by a plot.

$$\boxed{f(x) = e^x}$$

All derivatives are the same: $f^{(n)}(x) = e^x$, $n = 0, 1, 2, ...$

If we choose $a = 0$, then in the neighborhood of 0, the approximation is

$$P_n(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... + \frac{x^n}{n!}$$

and the error term can be written as, for z between 0 and x,

$$R_n(x) = \frac{e^z x^{n+1}}{(n+1)!}$$

Note that x can be negative. To assess the error, we need the upper bound of $|R_n(x)|$.

$$|R_n(x)| \leq \frac{e^z |x|^{n+1}}{(n+1)!}$$

Approximation of $n!$

$$n! = 1 \times 2 \times 3 \times ... \times n \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \approx n^n$$

Or more precisely

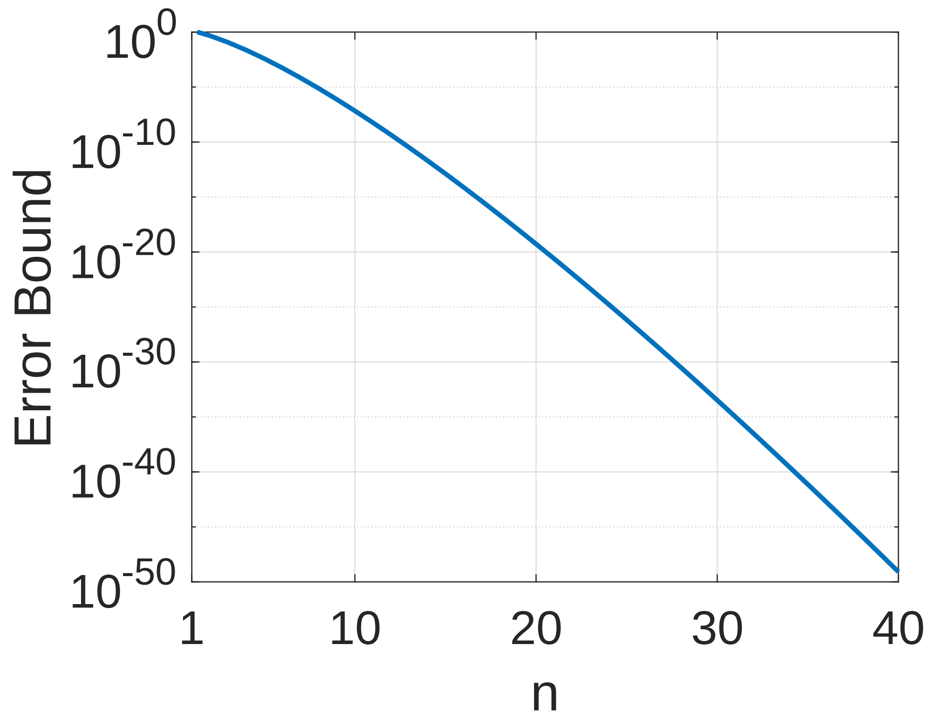
$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n < n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}$$

This result allows to upper bound the approximation error (recall z is between 0 and x):

$$|R_n(x)| \leq \frac{e^z|x|^{n+1}}{(n+1)!} < \frac{e^z|x|^{n+1}}{\sqrt{2\pi(n+1)}\left(\frac{n+1}{e}\right)^{n+1}} = \frac{e^z}{\sqrt{2\pi(n+1)}}\left|\frac{ex}{n+1}\right|^{n+1}$$

Thus, as a rule of thumb, we should let $n > e|x|$ (for $x \gg 1$).

For $x \in [0, 1]$, it is safe to control a bound $R_n(x) < \frac{e}{\sqrt{2\pi(n+1)}} \left| \frac{e}{n+1} \right|^{n+1}$, which is plotted:



Also, note that we do not have to always expand around $x = 0$. Once we have an accurate estimate of e, we can find good estimate of e^x by many means including Taylor expansion.

$$\boxed{f(x) = \sin(x)}$$

$$f^{(1)}(x) = \cos(x), \;\; f^{(2)}(x) = -\sin(x), \;\; f^{(3)}(x) = -\cos(x), \;\; f^{(4)}(x) = \sin(x)$$

It is particularly convenient to expand $\sin(x)$ around $x = 0$, as $\cos(0) = 1$ and $\sin(0) = 0$.

$$f(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

Consider $x \in [0, \pi/4]$, we can bound the approximation error:

$$|R_n(x)| = \left| \frac{f^{(n+1)}(z)}{(n+1)!} x^{n+1} \right| \leq \frac{(\pi/4)^{n+1}}{(n+1)!} < \frac{1}{\sqrt{2\pi(n+1)}} \left| \frac{e\pi/4}{n+1} \right|^{n+1}$$

Therefore, if $n = 3$, i.e., the approximation $f(x) \approx x - \frac{x^3}{3!}$, is already very good.

$$\boxed{f(x) = \cos(x)}$$

$$f^{(1)}(x) = -\sin(x),\;\; f^{(2)}(x) = -\cos(x),\;\; f^{(3)}(x) = \sin(x),\;\; f^{(4)}(x) = \cos(x)$$

Again, it is convenient to expand $\cos(x)$ around $x = 0$, as $\cos(0) = 1$ and $\sin(0) = 0$.

$$f(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + ...$$

Consider $x \in [0, \pi/4]$, we have the same bound of the approximation error:

$$|R_n(x)| = \left| \frac{f^{(n+1)}(z)}{(n+1)!} x^{n+1} \right| \leq \frac{(\pi/4)^{n+1}}{(n+1)!} < \frac{1}{\sqrt{2\pi(n+1)}} \left| \frac{e\pi/4}{n+1} \right|^{n+1}$$

Again, if $n = 3$, i.e., the approximation $f(x) \approx 1 - \frac{x^2}{2!}$, is already very good, although we effectively have only used $n = 2$.

For values outside of $[0, \pi/4]$, we can take advantage of many trigonometric relationships.

$$\boxed{f(x) = \log(x)}$$

$$f^{(1)}(x) = \frac{1}{x}, \;\; f^{(2)}(x) = \frac{-1}{x^2}, \;\; f^{(3)}(x) = \frac{2}{x^3}, \;\; f^{(4)}(x) = \frac{-6}{x^4} = \frac{-3!}{x^4}$$

We should not expand $f(x)$ at $x = 0$ because $\log(0) = -\infty$. Let $x = 1$, then

$$f(1) = 0, \; f^{(1)}(1) = 1, \;\; f^{(2)}(1) = -1, \;\; f^{(3)}(1) = 2!, \;\; f^{(4)}(1) = -3!$$

$$f^{(n)}(1) = (-1)^{n+1}(n-1)!, \qquad \frac{f^{(n)}(1)}{n!} = \frac{(-1)^{n+1}}{n}$$

Expand $f(x) = P_n(x) + R_n(x)$ around $x = 1$, we have

$$P_n(x) = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + ... + (-1)^{n+1}\frac{(x-1)^n}{n}$$

$$R_n(x) = (-1)^{n+2}\frac{(x-1)^{n+1}}{n+1}\frac{1}{z^{n+1}}, \qquad \text{where } \; z \; \text{ is between } \; 1 \; \text{ and } \; x$$

If $x > 1$, then $|R_n(x)| \leq \frac{(x-1)^{n+1}}{n+1}$. The convergence is not so fast.

if $x < 1$, then $|R_n(x)| \leq \frac{(x-1)^{n+1}}{n+1} \frac{1}{z^{n+1}} \leq \frac{(x-1)^{n+1}}{n+1} \frac{1}{x^{n+1}} = \frac{1}{n+1} \left(1 - \frac{1}{x}\right)^{n+1}$, which could be really bad if x is very small. We need a fix.

One strategy is to expand $\log(x)$ around e^{-m}, as $\log(e^{-m}) = -m$ exactly. Equivalently, we can expand $\log(x)$ around e^m for $x > 1$ and a good m, and use $\log(1/x)$ if $x < 1$.

Or evenly more simply, we expand $\log(x/e^m) = \log(x) - \log(e^m) = \log(x) - m$. Thus, we really only need to find good approximation for $\log(x)$ with x between 1 and $e - 1$.

Evaluating R_n by Computing $n!$ Directly

$$R_n(x) = \frac{f^{(n+1)}(z)}{(n+1)!}(x-a)^{n+1}, \qquad \text{where } z \text{ is between } x \text{ and } a.$$

When n is not too large, it is feasible to compute $n!$ directly:

$$0! = 1,\ 1! = 1,\ 2! = 2,\ 3! = 6,\ 4! = 24,\ 5! = 120,\ 6! = 720, ...$$

Recall when $f(x) = e^x$, $a = 0$, and $0 \leq x \leq 1$, we have $|R_n(x)| \leq \frac{e^z |x|^{n+1}}{(n+1)!} \leq \frac{e}{(n+1)!}$. If we hope to achieve $|R_n(x)| \leq \epsilon$, then it suffices to have $(n+1)! \geq e/\epsilon$. Suppose we choose $\epsilon = 0.01$ (i.e., $e/\epsilon = 271.83$). Because $6! = 720$ and $5! = 120$, we need $(n+1) = 6$, i.e., $n = 5$.

We can do similar calculations for $\sin(x)$, $\cos(x)$, etc.

Obviously, this is also related to the topic of root-finding.

Evaluating $P_n(x)$

At the very least, we should be able re-write

$$P_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + ... + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

to be a recursive formula

$$P_0(x) = f(a), \qquad T_0 = 1$$

$$P_i(x) = P_{i-1}(x) + f^{(i)}(a)T_i, \qquad T_i = T_{i-1}\frac{x-a}{i}, \;\; i = 1, 2, ..., n$$

Check book lecture notes for a more sophisticated way to halve the number of multiplications.

Root-Finding

In numerous scenarios, we might need to solve $f(x) = 0$, for some function f, which can be linear, polynomial, or more sophisticated forms.

If $f(x) = a_0 + a_1 x$, then the solution is simply $x = -a_0/a_1$.

If $f(x) = a_0 + a_1 x + a_2 x^2$, then the solution to $f(x) = 0$ has two roots:

$$x = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_0 a_2}}{2a_2}$$

If $f(x)$ is a 3rd-order polynomial, there is also closed-form solution, i.e., Cardano's method.

But in general, we have to solve $f(x) = 0$ numerically.

Bisection Method

If we are able to find an interval $x \in [a, b]$ so that $f(a)f(b) < 0$, i.e., one point is below zero and another point is above 0, then we can always use bisection method to find root $f(x) = 0$.

In general, we can only hope to find a point which is close enough to the exact root. Let x^* be the point such that $f(x^*) = 0$. We hope to find an x within $|x - x^*| < \epsilon$ for some small ϵ.

Algorithm: **Bisec**(f, a, b, ϵ):

1: $c = (a + b)/2$

2: If $|c - a| < \epsilon$ or $|b - c| < \epsilon$, then accept c as the solution.

3: If $f(a)f(c) < 0$, then let $b = c$. Go to step 1.

4: Otherwise let $a = c$. Go to step 1.

An Example

Consider $f(x) = x^2 - e^x/5$. We will use bisection method to find the solution to $f(x) = 0$ in the interval of $[0, 1]$.

iteration	a	b	c	$f(a)$	$f(b)$	$f(c)$
0	0	1	0.5	< 0	> 0	< 0
1	0.5	1	0.75	< 0	> 0	> 0
2	0.5	0.75	0.625	< 0	> 0	> 0
3	0.5	0.625	0.5625	< 0	> 0	< 0
4	0.5625	0.6250	0.5938	< 0	> 0	< 0

The solution is 0.6053... which can be reached after 10 iterations. Thus, we can see that the bisection method may take many iterations to converge.

Rate of Convergence

Denote initial interval by $[a_1,\ b_1]$ and we know that the true solution x^* is in the interval. Denote $c_j = (a_j + b_j)/2$. Then obviously

$$|c_1 - x^*| < \frac{1}{2}|b_1 - a_1|$$

$$|c_2 - x^*| < \left(\frac{1}{2}\right)^2 |b_1 - a_1|$$

$$|c_n - x^*| < \left(\frac{1}{2}\right)^n |b_1 - a_1|$$

To ensure the error $|c_n - x^*| < \epsilon$, it suffices to choose n so that

$$\left(\frac{1}{2}\right)^n |b_1 - a_1| < \epsilon, \ \ i.e., \ \ n > \frac{\log \frac{b_1 - a_1}{\epsilon}}{\log 2}$$

Note that we always use $\log$ for natural log.

Pros and Cons of the Bisection Method

Advantages:

- Guaranteed to find all the roots within $[a,\ b]$.
- Guaranteed rate of convergence.

Disadvantages:

- The rate of convergence is quite slow.
- Need to find an appropriate $[a,\ b]$, which also affects rate of convergence.

Note that the bisection method did not take advantage of the derivatives such as $f'(x)$.

Newton's Method

Suppose we approximate $f(x)$ by a first-order polynomial at x_0.

$$f(x) \approx f(x_0) + (x - x_0)f'(x_0), \implies x \approx x_0 + \frac{f(x) - f(x_0)}{f'(x_0)}$$

Now suppose x is very close to x^* (i.e., $f(x^*) = 0$). Then

$$x \approx x_0 - \frac{f(x_0)}{f'(x_0)}$$

This leads to Newton's method for root-finding:

1: start with $x = x_0$

2: $x_{i+1} \leftarrow x_i - \frac{f(x_i)}{f'(x_i)}$, i = 0, 1, ...,

3: Terminate when a stopping criterion is met.

An Example

Consider $f(x) = x^2 - e^x/5$, i.e., $f'(x) = 2x - e^x/5$; We choose to start from $x = 1$.

iteration	x	$f(x)$	$f'(x)$
0	1.0000	0.4563	1.4563
1	0.6867	0.0741	0.9759
2	0.6107	0.0046	0.8531
3	0.6053	0.0000	0.8442
4	0.6053	0.0000	0.8442

The solution 0.6053 is found with merely 3 iterations. Thus, Newton's method can be much faster than bisection method.

Error Analysis of Newton's Method

Apply Taylor approximation around x_n (Recall z is between x_n and x^*).

$$f(x^*) = f(x_n) + (x^* - x_n)f'(x_n) + \frac{1}{2}f''(z)(x^* - x_n)^2 = 0$$

$$\Longrightarrow$$

$$\frac{f(x_n)}{f'(x_n)} + (x^* - x_n) + \frac{1}{2}\frac{f''(z)}{f'(x_n)}(x^* - x_n)^2 = 0$$

$$\Longrightarrow$$

$$x^* - x_{n+1} + \frac{1}{2}\frac{f''(z)}{f'(x_n)}(x^* - x_n)^2 = 0$$

Therefore, after n iteration, the error becomes

$$x^* - x_{n+1} = -\frac{f''(z)}{2f'(x_n)}(x^* - x_n)^2$$

which is typically much faster than the rate of bisection method. However, the rate of convergence depends on the derivatives and the initial value. In fact, it may not converge.

Root-Finding and MLE Equation

The maximum likelihood estimator (MLE) is a basic concept in statistics. Suppose a random variable x follows a distribution with density $f(x)$ (don't worry if some words are new to you) with a parameter θ.

For example, when we toss a coin, the outcome x is a random variable taking values $\{head, tail\}$. The parameter θ here is the probability of seeing head ($\theta = 0.5$ for a fair coin). Suppose we repeat the experiment n times with n observations x_i, $i = 1$ to n, we can estimate the parameter θ.

In the coin-tossing example, let $x_i = 1$ if head and $x_i = 0$ otherwise. Then we can estimate θ, the probability of seeing heads, as $\hat{\theta} = \sum_{i=1}^n x_i/n$.

For the general case, we want to find the θ to maximize the joint likelihood $\prod_{i=1}^n f(x_i;\theta)$. Or equivalently to maximize the log-likelihood $l(\theta) = \sum_{i=1}^n \log f(x_i;\theta)$. To find the maximum, we need to take derivative with respect to θ and set it to zero, i.e.,

$$\sum_{i=1}^n \frac{f'(x_i;\hat{\theta})}{f(x_i;\hat{\theta})} = 0$$

which is the MLE equation. The derivative is with respect to θ.

MLE for Coin-Tossing

Let $p =$ probability of heads, i.e., $1 - p$ = probability of tails.

Define $x_i = 1$ if a toss is head and $x_i = 0$ otherwise.

We repeat the coin-tossing n times and record x_i for $i = 1$ to n.

The likelihood for the i-th toss is $p^{x_i}(1-p)^{1-x_i}$.

The joint-likelihood for all n tosses is $L = \prod_{i=1}^n p^{x_i}(1-p)^{1-x_i}$.

The log joint-likelihood is a function of p:

$$l(p) = \sum_{i=1}^{n} x_i \log p + (1 - x_i) \log(1-p)$$

We hope to find the p to maximize $l(p)$, which amounts to solving $l'(p) = 0$.

$$l'(p) = \sum_{i=1}^{n} \frac{x_i}{p} - \frac{1-x_i}{1-p} = \sum_{i=1}^{n} \frac{x_i - p}{p(1-p)}$$

Setting $l'(p) = 0$ yields $\sum_{i=1}^{n}(x_i - p) = 0$, which confirms the intuitive solution

$$\hat{p} = \frac{\sum_{i=1}^{n} x_i}{n}$$

In statistics, we often use $\hat{\theta}$ to indicate an estimate of θ.

This is a simple example of MLE with exact solution. In most cases, we do not have exact solutions.

Contingency Table Estimations

Original Contingency Table

N_{11}	N_{12}
N_{21}	N_{22}

Sample Contingency Table

n_{11}	n_{12}
n_{21}	n_{22}

Suppose we only observe the sample contingency table, how can we estimate the original table, if $N = N_{11} + N_{12} + N_{21} + N_{22}$ is known?

(Almost) equivalently, how can we estimate $\pi_{ij} = \frac{N_{ij}}{N}$?

An Example of Contingency Table

The task is to estimate how many times two words (e.g., Rutgers and University) co-occur in all the Web pages.

	Word 2	No Word 2
Word 1	N_{11}	N_{12}
No Word 1	N_{21}	N_{22}

N_{11}: number of documents containing both word 1 and word 2.

N_{22}: number of documents containing neither word 1 nor word 2.

Google Pagehits

Google tells user the number of Web pages containing the input query word(s).

Pagehits by typing the following queries in Google (numbers can change):

- a : 25,270,000,000 pages (a surrogate for N, the total # of pages).
- Rutgers : 38,400,000 pages. $(N_{11} + N_{12})$
- University : 3,050,000,000 pages. $(N_{11} + N_{21})$
- Rutgers University : 38,500,000 pages. (N_{11})

	Word 2	No Word 2
Word 1	N_{11}	N_{12}
No Word 1	N_{21}	N_{22}

How much do we believe these numbers?

Suppose there are in total $n = 10^7$ word items.

It is easy to store 10^7 numbers (how many documents each word occurs in), but it would be difficult to store a matrix of $10^7 \times 10^7$ numbers (how many documents a pair of words co-occur in).

Even if storing $10^7 \times 10^7$ is not a problem (it is Google), it is much more difficult to store $10^7 \times 10^7 \times 10^7$ numbers, for 3-way co-occurrences (e.g., Rutgers, University, Statistics).

Even if we can store 3-way or 4-way co-occurrences, most of the items will be so rare that they will almost never be used.

Therefore, it is realistic to believe that the counts for individual words are exact, but the numbers of co-occurrences may be estimated, eg, from some samples.

Estimating Contingency Tables by MLE of Multinomial Sampling

Original Contingency Table

π_{11}	π_{12}
π_{21}	π_{22}

Sample Contingency Table

n_{11}	n_{12}
n_{21}	n_{22}

Observations: $(n_{11}, n_{12}, n_{21}, n_{22})$, $\quad n = n_{11} + n_{12} + n_{21} + n_{22}$.

Parameters $(\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22})$, $\quad (\pi_{11} + \pi_{12} + \pi_{21} + \pi_{22} = 1)$

The likelihood is proportional to

$$\pi_{11}^{n_{11}} \pi_{12}^{n_{12}} \pi_{21}^{n_{21}} \pi_{22}^{n_{22}}$$

The log likelihood is mainly

$$l = n_{11} \log \pi_{11} + n_{12} \log \pi_{12} + n_{21} \log \pi_{21} + n_{22} \log \pi_{22}$$

We can choose to write $\pi_{22} = 1 - \pi_{11} - \pi_{12} - \pi_{21}$.

Finding the maximum (setting first derivatives to be zero)

$$\frac{\partial l}{\pi_{11}} = \frac{n_{11}}{\pi_{11}} + \frac{-n_{22}}{1 - \pi_{11} - \pi_{12} - \pi_{21}} = 0,$$

$$\Longrightarrow \frac{n_{11}}{\pi_{11}} = \frac{n_{22}}{\pi_{22}} \text{ or written as } \frac{\pi_{11}}{\pi_{22}} = \frac{n_{11}}{n_{22}}$$

Similarly

$$\frac{n_{11}}{\pi_{11}} = \frac{n_{12}}{\pi_{12}} = \frac{n_{21}}{\pi_{21}} = \frac{n_{22}}{\pi_{22}}.$$

Therefore

$$\pi_{11} = n_{11}\lambda, \;\; \pi_{12} = n_{12}\lambda, \;\; \pi_{21} = n_{21}\lambda, \;\; \pi_{22} = n_{22}\lambda,$$

Summing up all the terms

$$1 = \pi_{11} + \pi_{12} + \pi_{21} + \pi_{22} = [n_{11} + n_{12} + n_{21} + n_{22}]\,\lambda = n\lambda$$

yields $\lambda = \frac{1}{n}$.

The MLE solution is

$$\hat{\pi}_{11} = \frac{n_{11}}{n}, \qquad \hat{\pi}_{12} = \frac{n_{12}}{n}, \qquad \hat{\pi}_{21} = \frac{n_{21}}{n}, \qquad \hat{\pi}_{22} = \frac{n_{22}}{n}.$$

Finding the MLE Solution by Lagrange Multiplier

MLE as a constrained optimization:

$$\underset{\pi_{11},\pi_{12},\pi_{21},\pi_{22}}{\text{argmax}} \quad n_{11}\log\pi_{11} + n_{12}\log\pi_{12} + n_{21}\log\pi_{21} + n_{22}\log\pi_{22}$$

$$\text{subject to : } \pi_{11} + \pi_{12} + \pi_{21} + \pi_{22} = 1$$

The unconstrained optimization problem:

$$\underset{\pi_{11},\pi_{12},\pi_{21},\pi_{22}}{\text{argmax}} \quad L = n_{11}\log\pi_{11} + n_{12}\log\pi_{12} + n_{21}\log\pi_{21} + n_{22}\log\pi_{22}$$

$$- \lambda\left(\pi_{11} + \pi_{12} + \pi_{21} + \pi_{22} - 1\right)$$

Finding the optimum: $\frac{\partial L}{\partial z} = 0, \;\; z \in \{\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22}, \lambda\}$

$$\frac{n_{11}}{\pi_{11}} - \lambda = 0, \qquad \frac{n_{12}}{\pi_{12}} = \frac{n_{21}}{\pi_{21}} = \frac{n_{22}}{\pi_{22}} = \lambda, \quad \pi_{11} + \pi_{12} + \pi_{21} + \pi_{22} = 1$$

Contingency Table with Margin Constraints

Original Contingency Table

N_{11}	N_{12}
N_{21}	N_{22}

Sample Contingency Table

n_{11}	n_{12}
n_{21}	n_{22}

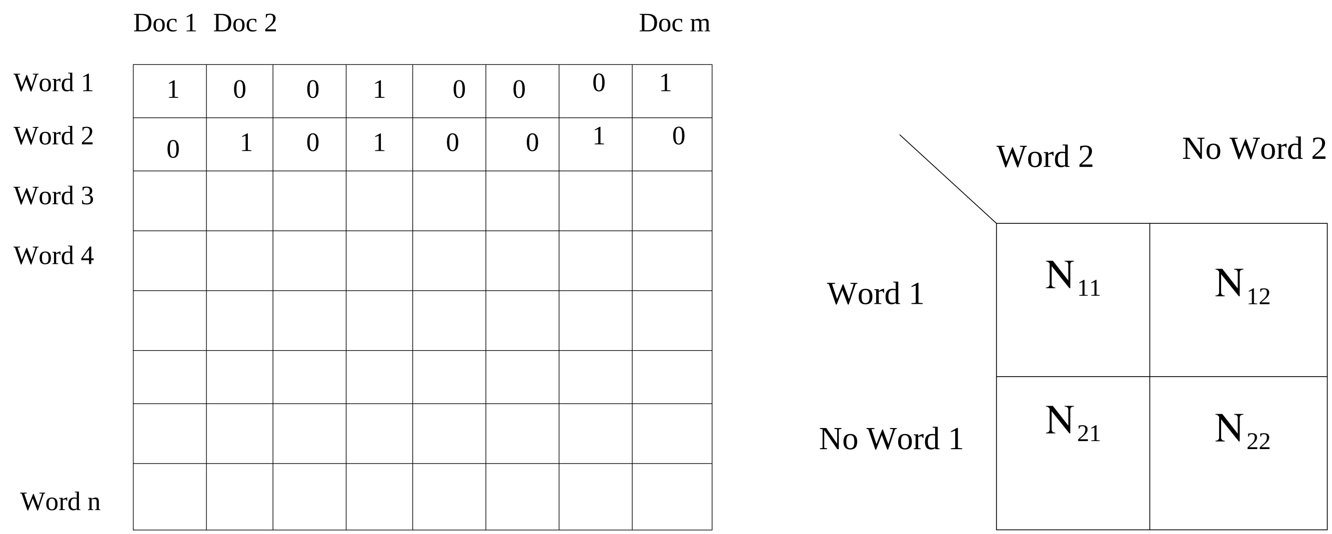
Margins: $M_1 = N_{11} + N_{12}, \qquad M_2 = N_{11} + N_{21}.$

Margins are much easier to be counted exactly than interactions.

An Example of Contingency Tables with Known Margins

Term-by-Document matrix $n = 10^6$ words and $m = 10^{10}$ (Web) documents.

Cell $x_{ij} = 1$ if word i appears in document j. $x_{ij} = 0$ otherwise.



N_{11}: number of documents containing both word 1 and word 2.

N_{22}: number of documents containing neither word 1 nor word 2.

Margins ($M_1 = N_{11} + N_{12}$, $M_2 = N_{11} + N_{21}$) for all rows costs nm, easy!

Interactions (N_{11}, N_{12}, N_{21}, N_{22}) for all pairs costs $n(n-1)m/2$, difficult!.

To avoid storing all pairwise contingency tables ($n(n-1)/2$ pairs in total), one strategy is to sample a fraction (k) of the columns of the original (term-doc) data matrix and and build sample contingency tables on demand, from which one can estimate the original contingency tables.

However, we observe that the margins (the total number of ones in each row) can be easily counted. This naturally leads to the conjecture that one might (often considerably) improves the estimation accuracy by taking advantage of the known margins. The next question is how.

Two approaches:

1. Maximum likelihood estimator (MLE) accurate but fairly complicated.
2. Iterative proportional scaling (IPS).

An Example of IPS for 2 by 2 Tables

n_{11}	n_{12}
n_{21}	n_{22}

The steps of IPS

(1) Modify the counts to satisfy the row margins.

(2) Modify the counts to satisfy the column margins.

(3) Iterate until some stopping criterion is met.

An example: $n_{11} = 30$, $n_{12} = 5$, $n_{21} = 10$, $n_{22} = 10$, $D = 600$.

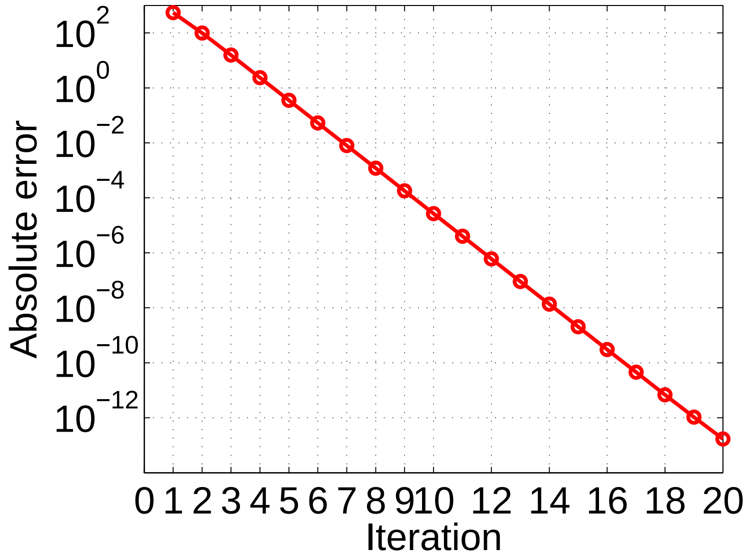
$M_1 = N_{11} + N_{12} = 400$, $\qquad M_2 = N_{11} + N_{21} = 300$.

In the first iteration: $N_{11} \leftarrow \frac{M_1}{n_{11}+n_{12}} n_{11} = \frac{400}{35} 30 = 342.8571$.

```
Iteration 1
  342.8571   57.1429
  100.0000  100.0000

  232.2581  109.0909
   67.7419  190.9091
Iteration 2
  272.1649  127.8351
   52.3810  147.6190

  251.5807  139.2265
   48.4193  160.7735
Iteration 3
  257.4985  142.5015
   46.2916  153.7084

  254.2860  144.3248
   45.7140  155.6752
Iteration 4
```

```
  255.1722  144.8278
   45.3987  154.6013

  254.6875  145.1039
   45.3125  154.8961
Iteration 5
  254.8204  145.1796
   45.2653  154.7347

  254.7477  145.2211
   45.2523  154.7789
Iteration 6
 254.7676  145.2324
   45.2453  154.7547

  254.7567  145.2386
   45.2433  154.7614
```

Error = $\left|\text{current step - previous step counts}\right|$, sum over four cells.



IPS converges fast and it always converges.

But how good are the estimates?: My general observation is that it is very good for 2 by 2 tables and the accuracy decreases (compared to the MLE) as the table size increases.

The MLE for 2 by 2 Table with Known Margins

Total samples : $n = n_{11} + n_{12} + n_{21} + n_{22}$

Total original counts : $N = N_{11} + N_{12} + N_{21} + N_{22}$, i.e., $\pi_{ij} = N_{ij}/N$.

Sample Contingency Table

n_{11}	n_{12}
n_{21}	n_{22}

Original Contingency Table

N_{11}	N_{12}
N_{21}	N_{22}

Margins: $M_1 = N_{11} + N_{12}, \qquad M_2 = N_{11} + N_{21}$.

If margins M_1 and M_2 are known, then only need to estimate N_{11}.

The likelihood

$$\propto \left(\frac{N_{11}}{N}\right)^{n_{11}} \left(\frac{N_{12}}{N}\right)^{n_{12}} \left(\frac{N_{21}}{N}\right)^{n_{21}} \left(\frac{N_{22}}{N}\right)^{n_{22}}$$

The log likelihood

$$\begin{aligned}
&n_{11}\log\left(\frac{N_{11}}{N}\right) + n_{12}\log\left(\frac{N_{12}}{N}\right) + n_{21}\log\left(\frac{N_{21}}{N}\right) + n_{22}\log\left(\frac{N_{22}}{N}\right) \\
=&n_{11}\log\left(\frac{N_{11}}{N}\right) + n_{12}\log\left(\frac{M_1 - N_{11}}{N}\right) + n_{21}\log\left(\frac{M_2 - N_{11}}{N}\right) \\
&+ n_{22}\log\left(\frac{N - M_1 - M_2 + N_{11}}{N}\right)
\end{aligned}$$

The MLE equation

$$g(N_{11}) = \frac{n_{11}}{N_{11}} - \frac{n_{12}}{M_1 - N_{11}} - \frac{n_{21}}{M_2 - N_{11}} + \frac{n_{22}}{N - M_1 - M_2 + N_{11}} = 0.$$

which is a cubic equation and can be solved either analytically or numerically.

Solving the MLE Equation $g(\hat{N}_{11}) = 0$

$$g(N_{11}) = \frac{n_{11}}{N_{11}} - \frac{n_{12}}{M_1 - N_{11}} - \frac{n_{21}}{M_2 - N_{11}} + \frac{n_{22}}{N - M_1 - M_2 + N_{11}}$$

$$g'(N_{11}) = -\frac{n_{11}}{N_{11}^2} - \frac{n_{12}}{(M_1 - N_{11})^2} - \frac{n_{21}}{(M_2 - N_{11})^2} - \frac{n_{22}}{(N - M_1 - M_2 + N_{11})^2}$$

Using Newton's method, at the $(t+1)$-th iteration, we compute

$$N_{11}^{(t+1)} = N_{11}^{(t)} - \frac{g(N_{11}^{(t)})}{g'(N_{11}^{(t)})}$$

Estimation Error

Often, the estimation error is measured by the "Mean Square Error (MSE)":

$$MSE(\hat{N}_{11}) = E\left(\hat{N}_{11} - N_{11}\right)^2$$

Here, the "E" means expectation. Suppose we repeat the sampling and estimation V times, each time we compute $\left(\hat{N}_{11} - N_{11}\right)^2$ and MSE is the average over V measurements.

One can verify by simulations that

$$MSE(\hat{N}_{11}) \approx \frac{N/n}{\frac{1}{N_{11}} + \frac{1}{(M_1 - N_{11})} + \frac{1}{(M_2 - N_{11})} + \frac{1}{(N - M_1 - M_2 + N_{11})}}$$

The "Margin-Free" Estimator and Error

It is more common to estimate N_{11} without using margins:

$$\hat{N}_{11,MF} = \frac{n_{11}}{n}N$$

and the estimation error can be computed exactly to be

$$MSE(\hat{N}_{11,MF}) = \frac{N/n}{\frac{1}{N_{11}} + \frac{1}{(N-N_{11})}}$$

Aren't these interesting?

Multiple Roots

In the previous problem, we have

$$g(N_{11}) = \frac{n_{11}}{N_{11}} - \frac{n_{12}}{M_1 - N_{11}} - \frac{n_{21}}{M_2 - N_{11}} + \frac{n_{22}}{N - M_1 - M_2 + N_{11}}$$

$$g'(N_{11}) = -\frac{n_{11}}{N_{11}^2} - \frac{n_{12}}{(M_1 - N_{11})^2} - \frac{n_{21}}{(M_2 - N_{11})^2} - \frac{n_{22}}{(N - M_1 - M_2 + N_{11})^2} \leq 0$$

Because g is non-increasing, the solution of $g(N_{11}) = 0$ is unique (on the real-line).

However in general $g(x) = 0$ can have multiple roots.

Another Cubic Equation

In general, a cubic equation

$$g(x) = x^3 + ax^2 + bx + c = 0$$

will have multiple real-valued roots, as the first derivative

$$g'(x) = 3x^2 + 2ax + b$$

is not necessarily monotonic. Here, we study a special form of cubic equation:

$$g(x) = x^3 - x^2\sum_{i=1}^{n} u_i v_i + x\left(-1 + \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2\right) - \sum_{i=1}^{n} u_i v_i$$

Special cases? For example, $u_i = v_i$, $\sum_{i=1}^{n} u_i v_i = 0$, etc.

In general, u_i and v_i are observations from data.

When $u_i = v_i$ for all i's, we have

$$\begin{aligned}
g(x) =& x^3 - x^2\sum_{i=1}^n u_i^2 + x\left(-1 + \sum_{i=1}^n u_i^2 + \sum_{i=1}^n u_i^2\right) - \sum_{i=1}^n u_i^2 \\
=& x^3 - x - \left(x^2 - 2x + 1\right)\sum_{i=1}^n u_i^2 \\
=& (x-1)(x^2+x) - (x-1)^2\sum_{i=1}^n u_i^2 \\
=& (x-1)\left(x^2 + x - (x-1)\sum_{i=1}^n u_i^2\right)
\end{aligned}$$

Therefore, one solution to $g(x) = 0$ will be $x = 1$.

In other words, when the two data vectors are identical, one solution will always be $x = 1$.

When $\sum_{i=1}^{n} u_i v_i = 0$ for all i's, we have

$$\begin{aligned} g(x) =& x^3 + x\left(-1 + \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2\right) \\ =& x\left(x^2 + \left(-1 + \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2\right)\right) \end{aligned}$$

Therefore, one solution will be $g(x) = 0$ will be $x = 0$.

In other words, when the two data vectors are orthogonal, one solution will always be $x = 0$.

Indeed, $g(x) = 0$ is the MLE equation for estimating the covariance when the data $(u_i,\ v_i)$ are sampled from a bivariate normal distribution with unit variances.

MLE Equation for Estimating Bivariate Normal Covariance

When data (u_i, v_i), $i = 1, ..., n$, are sampled from a bivariate normal distribution with unit variances and covariance ρ, the MLE equation for estimating ρ is the solution:

$$g(x) = x^3 - x^2 \sum_{i=1}^{n} u_i v_i + x\left(-1 + \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2\right) - \sum_{i=1}^{n} u_i v_i = 0$$

```
r=0.9;n=5;
Z=[1 r; r 1]^0.5*randn(2,n); u=Z(1,:); v=Z(2,:);
u =
    0.4300   -0.4534    0.7924    0.1123   -0.5033
v =
    0.2592   -0.6948    0.4281    0.7656   -0.6185
```

We would like to find the solution and study if there exist multiple roots. This is a good matlab exercise and a practice for the homework.

```
function p = TestRoots(r,n)

%matlab coding excercise for
%finding multiple roots to a cubic equation
%by exhaustive search in a finite range.

x = -1:1e-6:1;
m = 0;
N = 100;
for i = 1:N
    if(mod(i,10)==0)
        i
    end
    Z=[1 r; r 1]^0.5*randn(2,n); u=Z(1,:); v=Z(2,:);
    g = x.^3 - x.^2*(u*v') + x.*(-1+u*u'+v*v')-u*v';
    [~,ind] = sort(abs(g));
    cand = x(ind(1:10));
    cand = unique(round(cand*10^4)/10^4);
    if(length(cand)>1)
        m = m + 1;
    end
end
p = m/N;
```

Linear Interpolation

Given two data points: (x_0, y_0) and (x_1, y_1), we would like to find a linear function

$$P_1(x) = a_0 + a_1 x$$

which passes through both points. This can be solved easily from

$$a_0 + a_1 x_0 = y_0$$
$$a_0 + a_1 x_1 = y_1$$

to be

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0}$$
$$a_0 = y_1 - a_1 x_1 = y_1 - \frac{y_1 - y_0}{x_1 - x_0} x_1 = \frac{y_0 x_1 - y_1 x_0}{x_1 - x_0}$$
$$P_1(x) = a_0 + a_1 x = \frac{y_0 x_1 - y_1 x_0}{x_1 - x_0} + \frac{y_1 - y_0}{x_1 - x_0} x$$

For convenience, we can re-write the solution as linear combinations of y_0 and y_1:

$$
\begin{aligned}
P_1(x) =& a_0 + a_1 x \\
=& \frac{y_0 x_1 - y_1 x_0}{x_1 - x_0} + \frac{y_1 - y_0}{x_1 - x_0} x \\
=& y_0 \frac{x_1 - x}{x_1 - x_0} + y_1 \frac{x - x_0}{x_1 - x_0} \\
=& y_0 L_0(x) + y_1 L_1(x)
\end{aligned}
$$

where $L_0(x)$ and $L_1(x)$ are the “basis functions”.

Quadratic Interpolation

Given three data points: (x_0, y_0), (x_1, y_1), and (x_2, y_2), we would like to find a quadratic function

$$P_2(x) = a_0 + a_1 x + a_2 x^2$$

which passes through all three points. Again, this can be solved from a system of equations:

$$a_0 + a_1 x_0 + a_2 x_0^2 = y_0$$
$$a_0 + a_1 x_1 + a_2 x_1^2 = y_1$$
$$a_0 + a_1 x_2 + a_2 x_2^2 = y_2$$

which can be reduced to two equations:

$$a_1(x_1 - x_0) + a_2(x_1^2 - x_0^2) = (y_1 - y_0)$$
$$a_1(x_2 - x_1) + a_2(x_2^2 - x_1^2) = (y_2 - y_1)$$

We apply the usual trick:

$$a_1 + a_2(x_1 + x_0) = \frac{(y_1 - y_0)}{(x_1 - x_0)}$$
$$a_1 + a_2(x_2 + x_1) = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

to solve for a_2:

$$a_2 = \frac{\frac{(y_2-y_1)}{(x_2-x_1)} - \frac{(y_1-y_0)}{(x_1-x_0)}}{x_2 - x_0} = \frac{(y_2 - y_1)(x_1 - x_0) - (y_1 - y_0)(x_2 - x_1)}{(x_1 - x_0)(x_2 - x_1)(x_2 - x_0)}$$

then a_1 and a_0:

$$a_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)} - a_2(x_2 + x_1)$$
$$a_0 = y_2 - a_1 x_2 - a_2 x_2^2$$

At this point, we can use matlab to verify the correctness.

Once the formulas are verified to be correct, we can try to express $P_2(x)$ as

$$P_2(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x)$$

for which we just need to group all the terms which are multiplicative of y_0, y_1, y_2.

For y_0, we have

$$a_2 = \frac{(y_0)(x_2 - x_1)}{(x_1 - x_0)(x_2 - x_1)(x_2 - x_0)} + ... = \frac{y_0}{(x_1 - x_0)(x_2 - x_0)} + ... = \frac{y_0}{B} + ...$$

where we use $B = (x_1 - x_0)(x_2 - x_0)$ to simplify the expression. For now, we don't need to worry about other terms which we skip because they don't involve y_0.

$$a_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)} - a_2(x_2 + x_1) = -a_2(x_2 + x_1) + ... = -\frac{y_0}{B}(x_2 + x_1) + ...$$

$$a_0 = y_2 - a_1 x_2 - a_2 x_2^2 = \frac{y_0}{B}(x_2 + x_1)x_2 - \frac{y_0}{B}x_2^2 + ... = \frac{y_0}{B}(x_1 x_2) + ...$$

At this point, we can express $P_2(x)$ as

$$
\begin{aligned}
P_2(x) =& a_0 + a_1 x^2 + a_2 x^2 \\
=& \frac{y_0}{B}\left(x_1 x_2 - (x_2 + x_1)x + x^2\right) + \dots \\
=& y_0 \frac{(x - x_1)(x - x_2)}{(x_1 - x_0)(x_2 - x_0)} + \dots
\end{aligned}
$$

Therefore, we know $L_0(x)$. By symmetry, we also know $L_1(x)$ and $L_2(x)$.

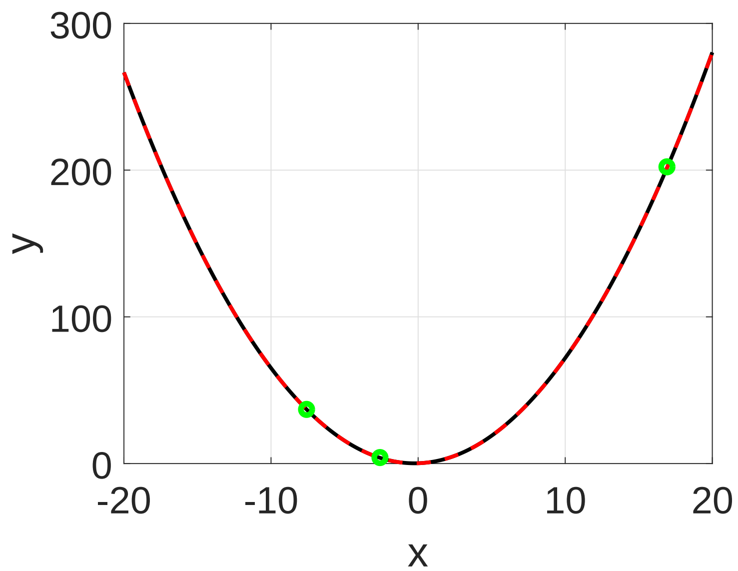
$$
\begin{aligned}
L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_1 - x_0)(x_2 - x_0)} \\
L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_0 - x_1)(x_2 - x_1)} \\
L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_1 - x_2)(x_0 - x_2)}
\end{aligned}
$$

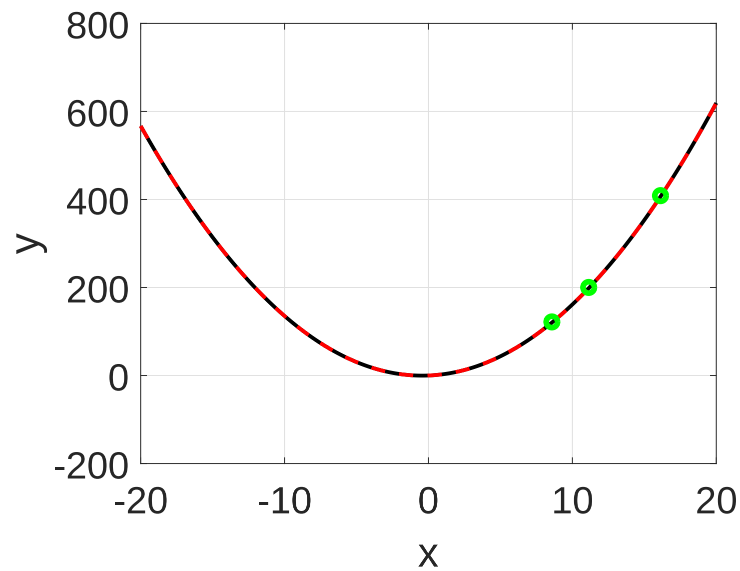
Again, we can verify the formulas by matlab.

```
function TestP2

A0 = rand; A1 = rand*2; A2 = rand*3;
x = -20:0.01:20;
y = A0 + A1*x + A2.*x.^2;

ind = randsample(length(x),3);
x0 = x(ind(1)); y0 = y(ind(1));
x1 = x(ind(2)); y1 = y(ind(2));
x2 = x(ind(3)); y2 = y(ind(3));

a2 = ((y2-y1)*(x1-x0)-(y1-y0)*(x2-x1));
a2 = a2/((x2-x1)*(x1-x0)*(x2-x0));

a1 = (y2-y1)./(x2-x1) - a2*(x2+x1);
a0 = y2 - a1*x2 - a2.*x2.^2;
[A0 A1 A2;a0 a1 a2]

L0 = (x-x1).*(x-x2)./((x1-x0)*(x2-x0));
L1 = (x-x0).*(x-x2)./((x2-x1)*(x0-x1));
L2 = (x-x1).*(x-x0)./((x1-x2)*(x0-x2));

Y = y0*L0 + y1*L1 + y2*L2;
figure;
plot(x,y,'k-','linewidth',2);hold on; grid on;
plot(x,Y,'r--','linewidth',2);
```

```
set(gca,'fontsize',20);xlabel('x');ylabel('y');
plot([x0 x1 x2],[y0 y1 y2],'go','linewidth',3);
```

See the plots for two runs. Note the effect of randomness.





Third-Order Basis Functions

Second-order basis functions:

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_1 - x_0)(x_2 - x_0)}, \qquad L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_0 - x_1)(x_2 - x_1)}$$

Can we guess the third-order basis functions (interpolation through 4 points)?

$$L_0(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_2 - x_0)(x_3 - x_0)},$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_2 - x_1)(x_3 - x_1)},$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_0 - x_2)(x_1 - x_2)(x_3 - x_2)},$$

$$L_3(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_0 - x_3)(x_2 - x_3)(x_2 - x_3)}$$

$$P_3(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) + y_3 L_3(x)$$

Unfortunately, if we modify the TestP2.m code, we will find the formulas are incorrect.

Revisit second-order basis functions:

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_1 - x_0)(x_2 - x_0)} = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$$

We can try another guess: $P_3(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) + y_3 L_3(x)$

$$L_0(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)},$$
$$L_1(x) = \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_2)},$$
$$L_2(x) = \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)},$$
$$L_3(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}$$

They are correct according to TestP3.m

```
function TestP3

A0 = (2*rand-1); A1 = (2*rand-1)*2;
A2 = (2*rand-1)*3; A3 = (2*rand-1)*4;
x = -20:0.01:20;
y = A0 + A1*x + A2.*x.^2 + A3.*x.^3;

ind = randsample(length(x),4);
x0 = x(ind(1)); y0 = y(ind(1));
x1 = x(ind(2)); y1 = y(ind(2));
x2 = x(ind(3)); y2 = y(ind(3));
x3 = x(ind(4)); y3 = y(ind(4));

L0 = (x-x1).*(x-x2).*(x-x3)./((x0-x1)*(x0-x2)*(x0-x3));
L1 = (x-x0).*(x-x2).*(x-x3)/((x1-x0)*(x1-x2)*(x1-x3));
L2 = (x-x0).*(x-x1).*(x-x3)/((x2-x0)*(x2-x1)*(x2-x3));
L3 = (x-x0).*(x-x1).*(x-x2)/((x3-x0)*(x3-x2)*(x3-x1));
Y = y0*L0 + y1*L1 + y2*L2+y3*L3;

figure;
plot(x,y,'k-','linewidth',2);hold on; grid on;
plot(x,Y,'r--','linewidth',2);
set(gca,'fontsize',20);xlabel('x');ylabel('y');
plot([x0 x1 x2 x3],[y0 y1 y2 y3],'go','linewidth',3);
```

General-Order Basis Function

Given $n+1$ data points: $(x_i,\ y_i)$, $i = 0, 1, 2, ..., n$, we hope to use an n-th order polynomial which will pass through all these $n+1$ points.

$$P_n(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) + y_3 L_3(x) + ... + y_n L_n(x)$$

$$\begin{aligned} L_k(x) &= \frac{(x-x_1)(x-x_2)...(x-x_{k-1})(x-x_{k+1})...(x-x_n)}{(x_k-x_1)(x_k-x_2)...(x_k-x_{k-1})(x_k-x_{k+1})...(x_k-x_n)}, \\ &= \frac{\prod_{i\neq k}(x-x_i)}{\prod_{i\neq k}(x_k-x_i)} = \prod_{i\neq k} \frac{x-x_i}{x_k-x_i} \end{aligned}$$

If $x = x_k$, then $L_k = 1$ and $L_{i\neq k} = 0$ for all $i \neq k$, i.e., $P_n(x_k) = y_k$.

These are called “LAGRANGE” basis functions. What about the approximation errors?

First-Order Divided Difference

Define the first-order divided difference as

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

By mean-value theorem, we know that

$$f(x_1) - f(x_0) = f'(c)(x_1 - x_0)$$

for some c between x_0 and x_1. Thus,

$$f[x_0, x_1] = \frac{f'(c)(x_1 - x_0)}{x_1 - x_0} = f'(c)$$

The divided difference is pretty much like derivative, especially as x_0 and x_1 are close.

Second-Order Divided Difference

Define the second-order divided difference as

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

By some argument (skipped in the textbook lecture notes), we have

$$f[x_0, x_1, x_2] = \frac{1}{2} f''(c)$$

for some c intermediate to x_0, x_1 and x_2.

Second-Order Divided Difference

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{f'(c_{12}) - f'(c_{01})}{x_2 - x_0} = f''(c)\frac{c_{12} - c_{01}}{x_2 - x_0}$$

for some c_{01} between x_0 and x_1, some c_{12} between x_1 and x_2, and some c between c_{01} and c_{12}. We are close but we can not argue that $\frac{c_{12}-c_{01}}{x_2-x_0} = \frac{1}{2}$. Therefore, we need a different approach to complete the proof.

General-Order Divided Difference

Define the n-order divided difference as

$$f[x_0, x_1, x_2, ..., x_n] = \frac{f[x_1, x_2, ..., x_n] - f[x_0, x_1, ..., x_{n-1}]}{x_n - x_0}$$

Again, we have

$$f[x_0, x_1, x_2, ..., x_n] = \frac{1}{n!} f^{(n)}(c)$$

for some c intermediate to x_0, x_1, ..., x_n.

When there is $n!$ involved, often the proof also involves the n-th derivatives.

P_n in terms of Divided Difference

Using definition of first-order divided difference:

$$f(x_1) = f[x_0, x_1](x_1 - x_0) + f(x_0)$$

we can express $P_1(x)$ as

$$\begin{aligned} P_1(x) =& f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0} \\ =& f(x_0)\frac{x - x_1}{x_0 - x_1} + (f[x_0, x_1](x_1 - x_0) + f(x_0))\ \frac{x - x_0}{x_1 - x_0} \\ =& f(x_0) + f[x_0, x_1](x - x_0) = f_0 + f[x_0, x_1](x - x_0) \end{aligned}$$

where we use f_0 to simplify $f(x_0)$.

We can re-write

$$
\begin{aligned}
f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} \\
&= \frac{\frac{f_2(x_1 - x_0) + f_0(x_2 - x_1) + f_1(x_0 - x_2)}{(x_2 - x_1)(x_1 - x_0)}}{x_2 - x_0} \\
&= \frac{f_0}{(x_0 - x_1)(x_0 - x_2)} + \frac{f_1}{(x_1 - x_2)(x_1 - x_0)} + \frac{f_2}{(x_2 - x_1)(x_2 - x_0)}
\end{aligned}
$$

and derive

$$
\begin{aligned}
P_2(x) =& f_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_1)(x_2-x_0)} \\
=& f_0 \frac{(x-x_1)}{(x_0-x_1)} + f_1 \frac{(x-x_0)}{(x_1-x_0)} \\
& + f_0 \frac{(x-x_1)(x-x_0)}{(x_0-x_1)(x_0-x_2)} + f_1 \frac{(x-x_0)(x-x_1)}{(x_1-x_0)(x_1-x_2)} + f_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_1)(x_2-x_0)} \\
=& f_0 \frac{(x-x_1)}{(x_0-x_1)} + f_1 \frac{(x-x_0)}{(x_1-x_0)} + (x-x_0)(x-x_1) \times \\
& \left(\frac{f_0}{(x_0-x_1)(x_0-x_2)} + \frac{f_1}{(x_1-x_0)(x_1-x_2)} + \frac{f_2}{(x_2-x_1)(x_2-x_0)} \right) \\
=& P_1(x) + (x-x_0)(x-x_1) f[x_0,x_1,x_2] \\
=& f_0 + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1)
\end{aligned}
$$

In general, we have

$$P_n(x) = P_{n-1}(x) + f[x_0, x_1, ..., x_n](x - x_0)(x - x_1)...(x - x_{n-1})$$

Now we study the error:

$$h(x) = f(x) - P_n(x)$$

At $x_0, x_1, ..., x_n$, we always have $h(x_i) = 0$, $i = 0$, ..., n.

Note that if $h(a) = h(b)$ and h is continuous and differentiable, then there must be a point c between a and b so that $h'(c) = 0$. This is essentially the mean-value theorem.

Therefore, there must be n points such that $h'(x) = 0$. Using this argument on h', we can eventually claim that there must be a point z intermediate to x_0, ..., x_n, so that $h^{(n)} = 0$, i.e.,

$$0 = f^{(n)}(z) - P_n^{(n)}(z)$$

Because $P_n(x)$ is an n-th order polynomial, it can be written as

$$\begin{aligned} P_n(x) =& P_{n-1}(x) + f[x_0, x_1, ..., x_n](x - x_0)(x - x_1)...(x - x_{n-1}) \\ =& x^n f[x_0, x_1, ..., x_n] + x^{n+1}(....) + ... \end{aligned}$$

In other words, $P_n^{(n)}(x) = n! f[x_0, x_1, ..., x_n]$. Therefore, we have proved that

$$f[x_0, x_1, ..., x_n] = \frac{1}{n!} f^{(n)}(c)$$

for some point c intermediate to all $x_0, x_1, ..., x_n$.

Approximation Error of Polynomial Interpolation

Given $n+1$ data points $(x_i,\ y_i)$, $i = 0, 1, ..., n$. If we fit an n-th order polynomial these $n+1$ points, to approximate the original function $f(x)$.

$$f(x) = P_n(x) + R_n(x)$$

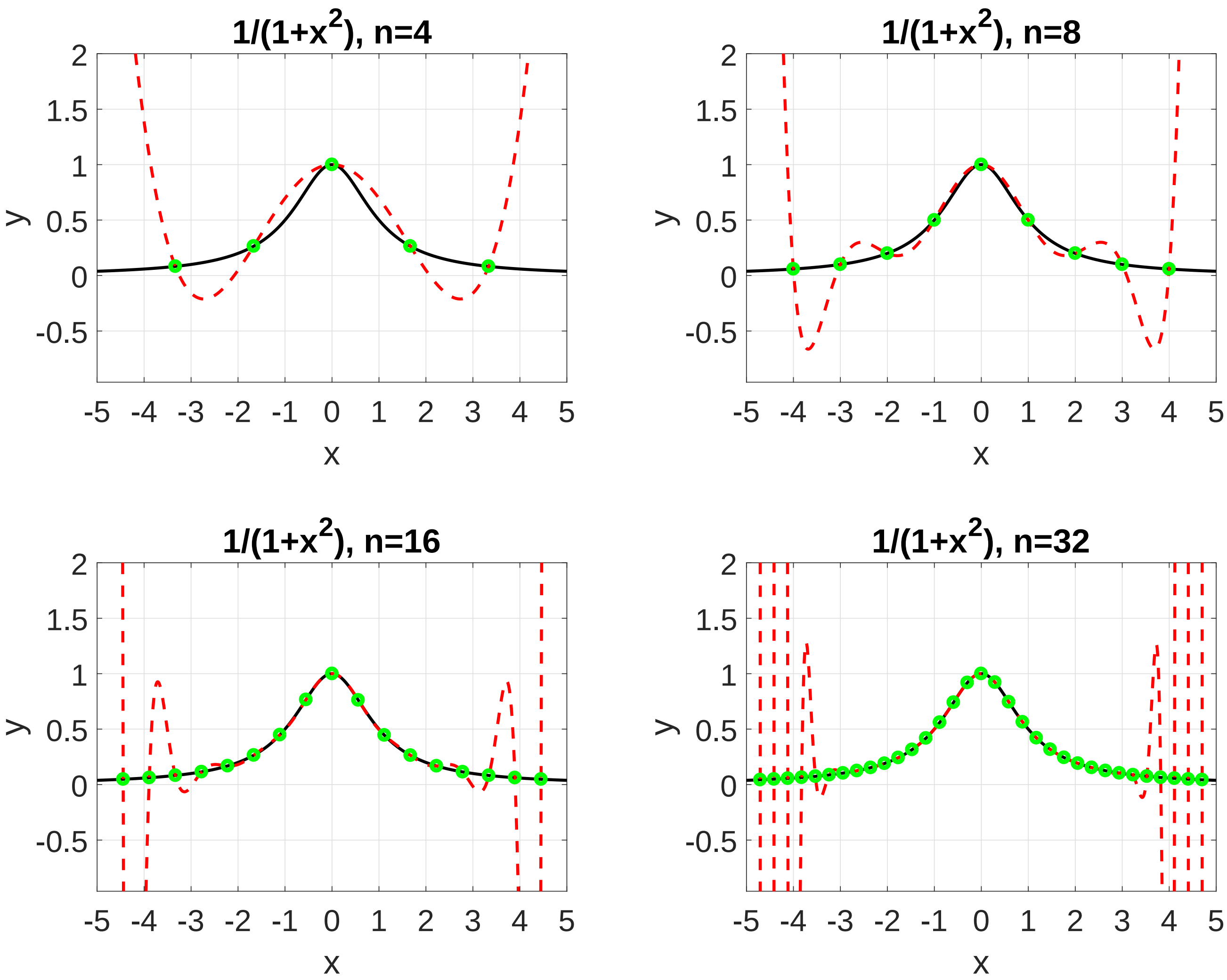
The error for x inside the range of $x_0, x_1, ..., x_n$ would be

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!}(x - x_0)(x - x_1)...(x - x_{n-1})(x - x_n)$$

for some c intermediate to $x_0, x_1, ..., x_n$.

However, if x is far outside the range of the points, then the errors can be very large. For example, if $|x| \gg |x_i|$, for all i, then roughly speaking $|R_n(x)| \propto |x|^n$.

Example: $y = 1/(1+x^2)$, Equally Spaced



Splines

Given n data points (x_1, y_1), ..., (x_n, y_n), where x_i's are in sorted order, the goal is to fit a curve $s(x)$ so that

$$1: \ s(x_i) = y_i, \quad i = 1, 2, ..., n$$

$$2: \ s(x), \ s'(x), \ s''(x), \qquad \text{are continuous on } \ x_i, \ \ i = 1, 2, ..., n$$

Among the many choices which satisfy the above two conditions, we choose the one which minimizes

$$\int_{x_i}^{x_{i+1}} |s''(x)|^2 dx$$

It turns out this has a unique solution.

Cubic Splines

Given n data points (x_1, y_1), ..., (x_n, y_n) within an interval $[a,\ b]$, where x_i's are in sorted order, i.e.,

$$a < x_1 < x_2 < ... < x_n < b$$

a cubic spline $s(x)$ is that

$$1:\ s(x_i) = y_i, \quad i = 1, 2, ..., n$$

$$2:\ s(x),\ s'(x),\ s''(x), \qquad \text{are continuous on } \ x_i,\ \ i = 1, 2, ..., n$$

$$3:\ \text{within each interval } \ s(x) \text{ is a polynomial of degree } \ \leq 3$$

An example of cubic spline (with a single “knot”) is

$$(x - \alpha)_+^3 = \begin{cases} (x-\alpha)^3 & \text{if } x \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

We can combine several simple splines to form a more complicated spline.

Define

$$s(x) = p_3(x) + \sum_{i=1}^{n} a_i (x - x_i)_+^3$$

where $p_3(x)$ is a 3rd-order polynomial. Then $s(x)$ is a cubic spline on $[-\infty,\ \infty]$ with knots $x_1, x_2, ..., x_n$.

Let's first look at some concrete examples.

An Example of (Natural) Cubic Splines

Given (x_1, y_1), (x_2, y_2), and (x_3, y_3), with $x_1 < x_2 < x_3$, we connect (x_1, y_1) and (x_2, y_2) by a cubic function

$$y = ax^3 + bx^2 + cx + d, \qquad y' = 3ax^2 + 2bx + c, \qquad y'' = 6ax + 2b$$

and (x_2, y_2) and (x_3, y_3) by another cubic function

$$y = ex^3 + fx^2 + gx + h, \qquad y' = 3ex^2 + 2fx + g, \qquad y'' = 6ex + 2f$$

We need to determine all the 8 coefficients: a,..., h. So technically we will need 8 equations.

We name this piecewise cubic curve as $s(x)$. We already have 4 constraints:

$$s(x_1) = y_1, \;\; s(x_3) = y_3, \;\; s(x_{2-}) = y_2, \;\; s(x_{2+}) = y_2$$

so we just need four more equations.

We would like the curve looks smooth. A good strategy is to enforce the first and second derivatives to be equal at x_2.

$$s'(x_{2-}) = s'(x_{2+}), \qquad s''(x_{2-}) = s''(x_{2+})$$

which give two additional equations. So we just need 2 more.

So far, we have not looked at the regions $< x_1$ and $> x_3$. One reasonable (and perhaps safe) strategy is to force the curve to be linear outside the range of data:

$$s''(x_1) = 0, \qquad s''(x_3) = 0$$

Therefore, in total we have 8 equations for 8 unknowns!!

$$ax_1^3 + bx_1^2 + cx_1 + d = y_1$$
$$ax_2^3 + bx_2^2 + cx_2 + d = y_2$$
$$ex_2^3 + fx_2^2 + gx_2 + h = y_2$$
$$ex_3^3 + fx_3^2 + gx_3 + h = y_3$$

$$3ax_2^2 + 2bx_2 + c = 3ex_2^2 + 2fx_2 + g$$
$$6ax_2 + 2b = 6ex_2 + 2f$$

$$6ax_1 + 2b = 0$$
$$6ex_3 + 2f = 0$$

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 & 0 & 0 & 0 & 0 \\ x_2^3 & x_2^2 & x_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_2^3 & x_2^2 & x_2 & 1 \\ 0 & 0 & 0 & 0 & x_3^3 & x_2^3 & x_3 & 1 \\ 3x_2^2 & 2x_2 & 1 & 0 & -3x_2^2 & -2x_2 & -1 & 0 \\ 6x_2 & 2 & 0 & 0 & -6x_2 & -2 & 0 & 0 \\ 6x_1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6x_3 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_2 \\ y_3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

If $AB = Z$, then $B = A^{-1}Z$. This can be solved very efficiently with matlab.

General Natural Cubic Splines

Given (x_i, y_i), $i = 1, 2, ..., n$, we would like to fit natural cubic splines $s(x)$.

There are in total $(n-1)4$ unknowns.

For two end points, we have $s(x_1) = y_1$ and $s(x_n) = y_n$.
For $n-2$ interior points, we have in total $(n-2)2$ constraints:

$$s(x_{i-}) = y_i, \;\; s(x_{i+}) = y_i, \;\; i = 2, 3, ..., n-1$$

For $n-2$ interior points, we have another in total $(n-2)2$ constraints:

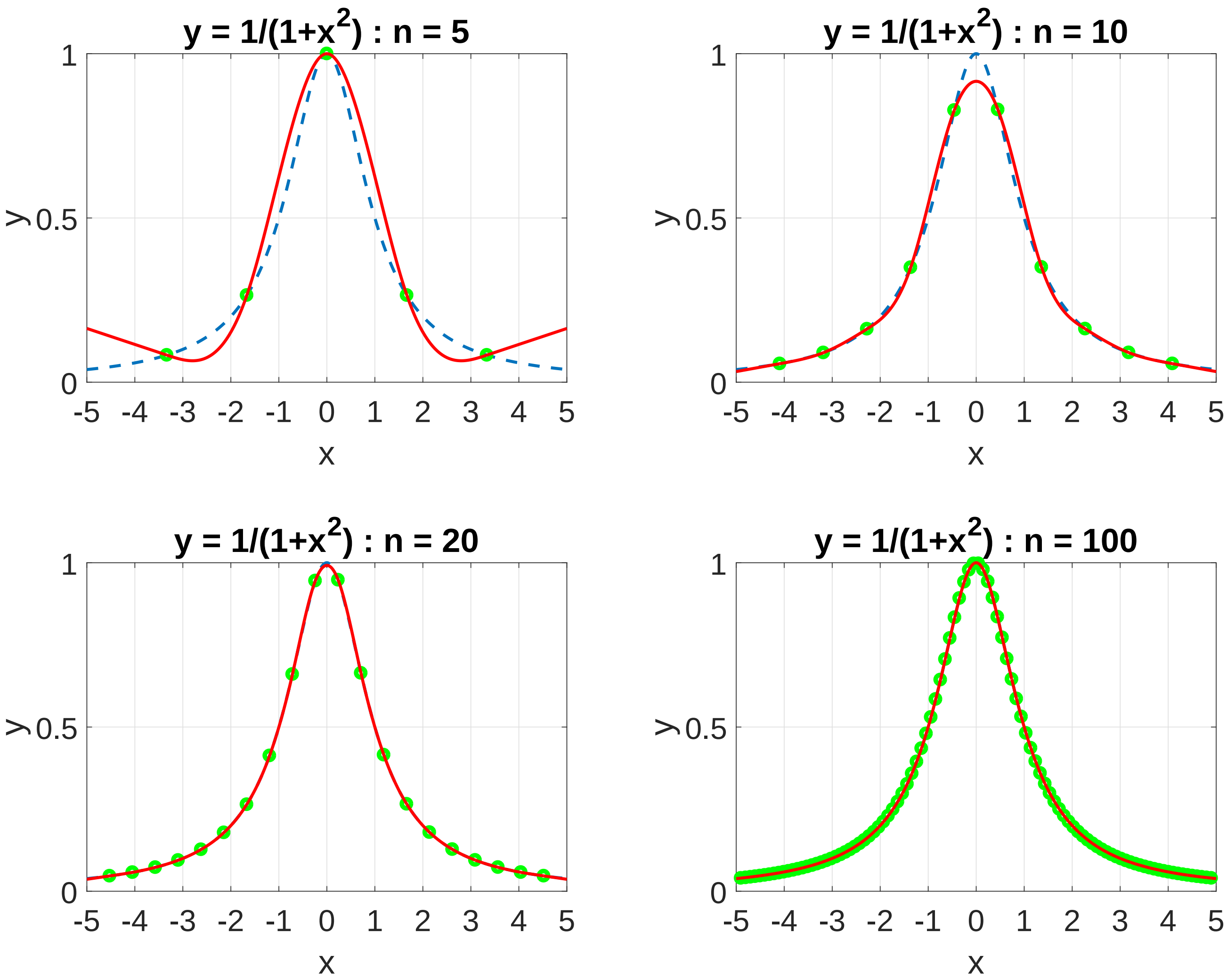
$$s'(x_{i-}) = s'(x_{i+}), \;\; s''(x_{i-}) = s''(x_{i+}), \;\; i = 2, 3, ..., n-1$$

Lastly, for two end points, there are two more constraints:

$$s''(x_1) = 0, \qquad s''(x_n) = 0$$

In total we have $2 + (n-2)2 + (n-2)2 + 2 = (n-1)4$ constraints (equations).

Example: $y = 1/(1+x^2)$, Equally Spaced



Not-A-Knot Cubic Spline

This is the version implemented in matlab. Basically, given (x_i, y_i), $i = 1$ to n with x_i's in sorted order, we try to fit cubic splines after deleting x_2 and x_{n-1}.

There will be $(n-3)4$ unknowns. Without using the natural cubic spline boundary conditions, we have $(n-3)4-2$ constraints. So we need 2 more equations.

We can then add the two additional constraints:

$$s(x_2) = y_2, \qquad s(x_{n-1}) = y_{n-1}$$

Thus, we will have enough constraints to solve for the $(n-3)4$ coefficients.

Theoretically, this method enjoys a better convergence property when n becomes large.

Least Square Approximation

Given a function $f(x)$ in the interval $[a,\ b]$, we seek to find a polynomial $p(x)$ of degree n that minimizes

$$RMSE = \sqrt{\frac{1}{b-a}\int_a^b \left[f(x) - p(x)\right]^2 dx}$$

where RMSE denotes “root-mean-square-error”. It is equivalent to minimizing

$$\int_a^b \left[f(x) - p(x)\right]^2 dx$$

or

$$\int_a^b -2f(x)p(x) + p^2(x)dx$$

Because $p(x)$ is a n-th order polynomial, we can always write $p(x)$ in the form

$$p(x) = \sum_{j=0}^{n} \beta_j P_j(x)$$

where $P_j(x)$ is a j-th order polynomial in a special form, to be identified. Without loss of generality, we consider the interval of $[-1,\ 1]$. We need to find P_j and β_j to minimize

$$\int_{-1}^{1} -2f(x)p(x) + p^2(x)dx = \int_{-1}^{1} -2f \sum_{j=0}^{n} \beta_j P_j + \left(\sum_{j=0}^{n} \beta_j P_j\right)^2 dx$$

$$= \int_{-1}^{1} \left[-2f \sum_{j=0}^{n} \beta_j P_j + \sum_{j=0}^{n} \beta_j^2 P_j^2 + \sum_{i \neq j} \beta_i \beta_j P_i P_j\right] dx$$

$$= -2 \sum_{j=0}^{n} \beta_j \int_{-1}^{1} f(x) P_j(x) dx + \sum_{j=0}^{n} \beta_j^2 \int_{-1}^{1} P_j^2(x) dx + \sum_{i \neq j} \beta_i \beta_j \int_{-1}^{1} P_i(x) P_j(x) dx$$

Suppose we require P_i to be "orthogonal" in that

$$\int_{-1}^{1} P_i(x)P_j(x)dx = 0, \qquad \text{if } i \neq j$$

then, we just need to find β_j and P_j to minimize

$$-2\sum_{j=0}^{n} \beta_j \int_{-1}^{1} f(x)P_j(x)dx + \sum_{j=0}^{n} \beta_j^2 \int_{-1}^{1} P_j^2(x)dx$$

whose first derivative with respect to β_i is

$$-2\int_{-1}^{1} f(x)P_i(x)dx + 2\beta_i \int_{-1}^{1} P_i^2(x)dx$$

setting which to zero yields the solution

$$\beta_i = \frac{\int_{-1}^{1} f(x)P_i(x)dx}{\int_{-1}^{1} P_i^2(x)dx}, \quad i = 0, 1, 2, ..., n$$

Legendre polynomials

Define the Legendre polynomials as

$$P_0(x) = 1$$

$$P_n(x) = \frac{1}{n!2^n} \frac{d^n}{dx^n} \left[(x^2 - 1)^n \right]$$

For example

$$P_1(x) = x$$

$$P_2(x) = \frac{1}{2}(3x^2 - 1)$$

$$P_3(x) = \frac{1}{2}(5x^3 - 3x)$$

$$P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3)$$

Properties

Introduce the special notation

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

- Degree and normalization

$$\text{deg}\, P_n = n, \qquad P_n(1) = 1$$

- Triple recursion relation

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x)$$

- Orthogonality

$$(P_i,\ P_j) = \begin{cases} 0, & i \neq j \\ \frac{2}{2j+1}, & i = j \end{cases}$$

- Legendre polynomials are symmetric or antisymmetric, that is

$$P_n(-x) = (-1)^n P_n(x)$$

- $$p_n(x) = \sum_{j=0}^{n} \frac{(f, P_j)}{(P_j, P_j)} P_j(x)$$

  is the least square approximation of degree n to $f(x)$ on $[-1, 1]$. Because $(P_j, P_j) = \frac{2}{2j+1}$, we have

$$p_n(x) = \sum_{j=0}^{n} (j + 1/2)(f, P_j) P_j(x)$$

  In general, we have to resort to numerical integration to find (f, P_j).

Approximation Errors

$$\int_{-1}^{1} \left[f(x) - p_n(x)\right]^2 dx$$

$$= \int_{-1}^{1} f^2(x)dx - 2\sum_{j=0}^{n} \beta_j \int_{-1}^{1} f(x)P_j(x)dx + \sum_{j=0}^{n} \beta_j^2 \int_{-1}^{1} P_j^2(x)dx$$

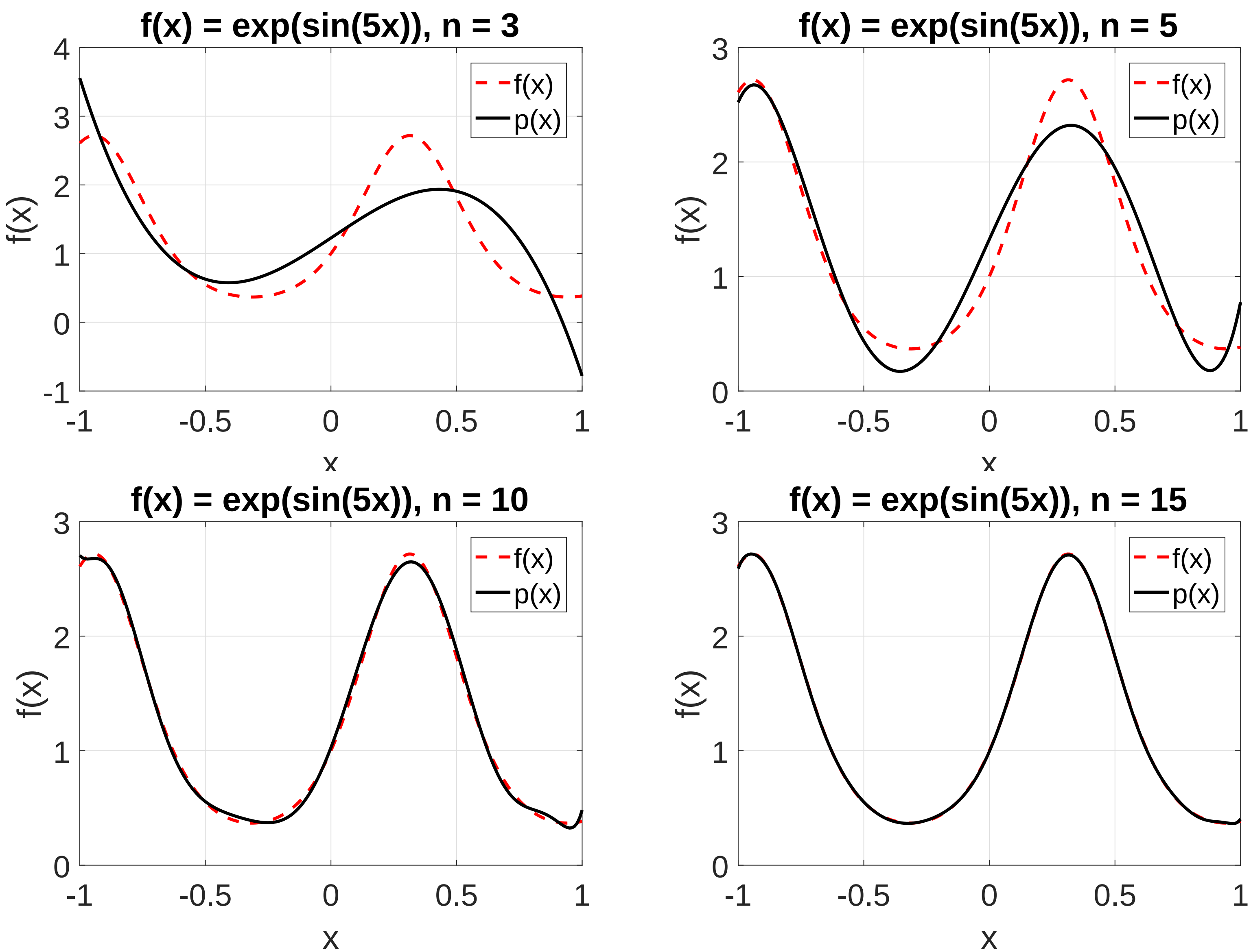
$$= \int_{-1}^{1} f^2(x)dx - 2\sum_{j=0}^{n} \beta_j (f, P_j) + \sum_{j=0}^{n} \frac{2\beta_j^2}{2j+1}$$

$$= \int_{-1}^{1} f^2(x)dx - 2\sum_{j=0}^{n} (j+1/2)(f,P_j)(f,P_j) + \sum_{j=0}^{n} (j+1/2)(f,P_j)(f,P_j)$$

$$= \int_{-1}^{1} f^2 dx - \sum_{j=0}^{n} (j+1/2)(f,P_j)^2$$

$$RMSE = \sqrt{\frac{1}{2}\int_{-1}^{1} \left[f(x) - p(x)\right]^2 dx} = \sqrt{\frac{1}{2}\int_{-1}^{1} f^2 dx - \frac{1}{2}\sum_{j=0}^{n} (j+1/2)(f,P_j)^2}$$

An Example



f(x) = exp(sin(5x)), n = 3
f(x) = exp(sin(5x)), n = 5
f(x) = exp(sin(5x)), n = 10
f(x) = exp(sin(5x)), n = 15
f(x)
p(x)
x

General Intervals

Consider approximating $f(x)$ on the finite interval $[a,\ b]$. Introduce the linear change of variables

$$t = \frac{2}{b-a}\left(x - \frac{a+b}{2}\right)$$

When $x = a$, we have $t = -1$. When $x = b$, we have $t = 1$. Equivalently, we have

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

We can then define another function $g(t)$ such that

$$g(t) = f\left(\frac{1}{2}\left[(1-t)a + (1+t)b\right]\right)$$

Once we approximate $g(t)$ on the interval $[-1,\ 1]$, we have equivalently approximated $f(x)$ on $[a,\ b]$.

Chebyshev Polynomials

For an integer $n \geq 0$, define the function

$$T_n(x) = \cos\left(n \cos^{-1}(x)\right), \qquad -1 \leq x \leq 1$$

which turns out to be polynomials for any $n \geq 0$:

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_2(x) = 2x^2 - 1$$

It might be easier to see by letting $\cos^{-1} x = \theta$ and using trigonometric relations

$$T_{n+1}(x) = \cos((n+1)\theta) = \cos n\theta \cos\theta - \sin n\theta \sin\theta$$
$$T_{n-1}(x) = \cos((n-1)\theta) = \cos n\theta \cos\theta + \sin n\theta \sin\theta$$
$$T_{n+1}(x) + T_{n-1}(x) = 2\cos n\theta \cos\theta = 2xT_n(x)$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Triple recursion relation

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \qquad n \geq 1$$
$$T_3(x) = 2x(2x^2 - 1) - x = 4x^3 - 3x$$
$$T_4(x) = 2x(4x^3 - 3x) - (2x^2 - 1) = 8x^4 - 8x^2 + 1$$

Note that the coefficient of the leading term is always 2^{n-1}.

Orthogonality

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

What is the use of it?

$\boxed{\text{MiniMax Theorem}}$

Let $\tilde{T}_n(x) = \frac{T_n(x)}{2^{n-1}}$. Then $\tilde{T}_n(x)$ is a **monic** polynomial in that coefficient of x^n term is 1.

Theorem: Let $n \geq 1$ be an integer, and consider all possible monic polynomials of degree n. Then the degree n monic polynomial with the smallest maximum on $[-1,\ 1]$ is the modified Chebyshev polynomial $T_n(x)$, and its maximum value on $[-1,\ 1]$ is $\frac{1}{2^{n-1}}$.

Near-MiniMax Approximation

Approximating $f(x)$ in the least square sense is common. However, in some situations, it is more desirable to minimize the maximum errors.

Consider $f(x)$ in $[-1,\ 1]$ and we would like to approximate it by a 3rd-order polynomial using four points (knots): x_0, x_1, x_2, x_3. How do we choose these points in the (at least near) minimax sense?

From previous results, we know that, suppose $c_3(x)$ is the resulting polynomial, the error is

$$f(x) - c_3(x) = \frac{\omega(w)}{4!} f^{(4)}(z), \qquad z \in [-1, 1]$$
$$\omega(x) = (x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

Thus, by ignoring $f^{(4)}(z)$, we hope to minimize the maximum error

$$\max_{-1 \leq x \leq 1} |\omega(x)|$$

Note that the $\omega(x)$ is monic with degree 4. From the previous “MiniMax Theorem”, we know that we need to make $\omega(x) = \tilde{T}_4(x)$. That is

$$\omega(x) = \tilde{T}_4(x) = x^4 - x^2 + \frac{1}{8}$$

In other words, the values x_0, x_1, x_2, and x_3 are the roots of $x^4 - x^2 + \frac{1}{8} = 0$.

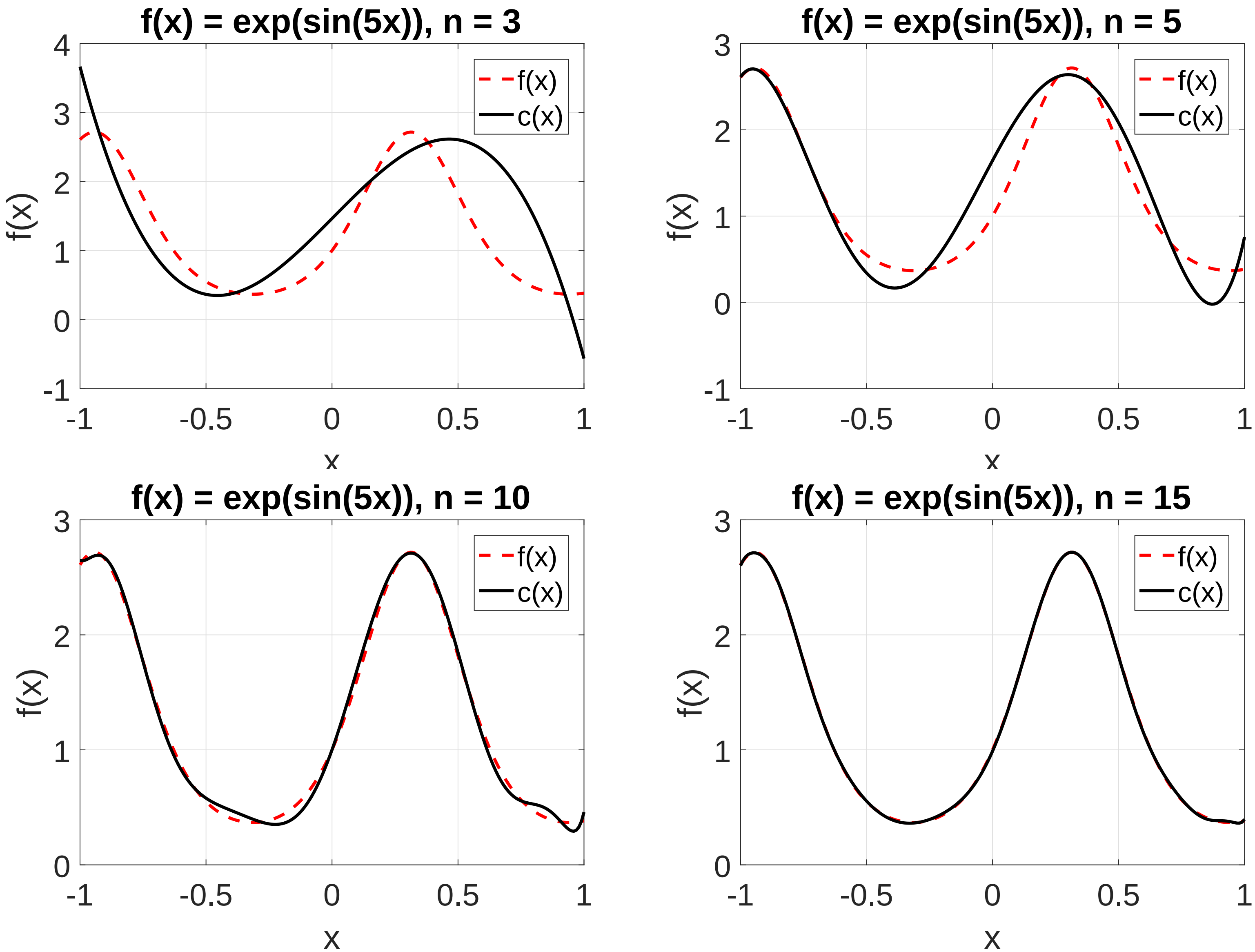
It is more convenient to find the roots in terms of θ. Because $T_4(x) = \cos 4\theta$, the solution to $T_4(x) = 0$ occurs at $4\theta = \pm\frac{\pi}{2},\ \pm\frac{3\pi}{2},\ \pm\frac{5\pi}{2},\ \pm\frac{7\pi}{2}$, ...

Therefore, the solutions are $x = \cos\frac{\pi}{8},\ \cos\frac{3\pi}{8},\ \cos\frac{5\pi}{8},\ \cos\frac{7\pi}{8}$.

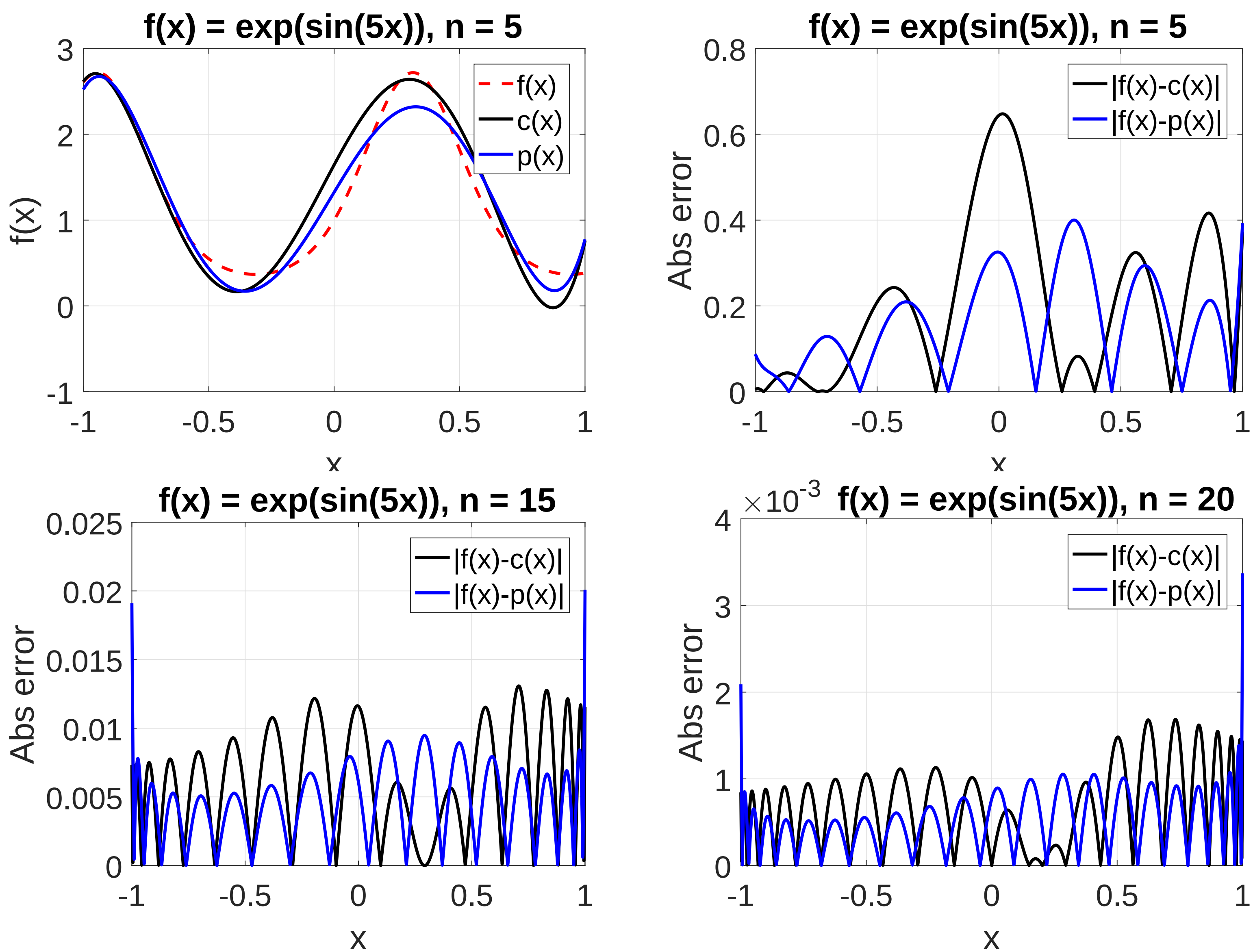
For general n, the unique solutions will be computed from $\cos n\theta = 0$ to be $\theta = \frac{\pi}{2n},\ \frac{3\pi}{2n},\ \frac{5\pi}{2n},\ \ldots\frac{2(n-1)\pi}{2n}$.

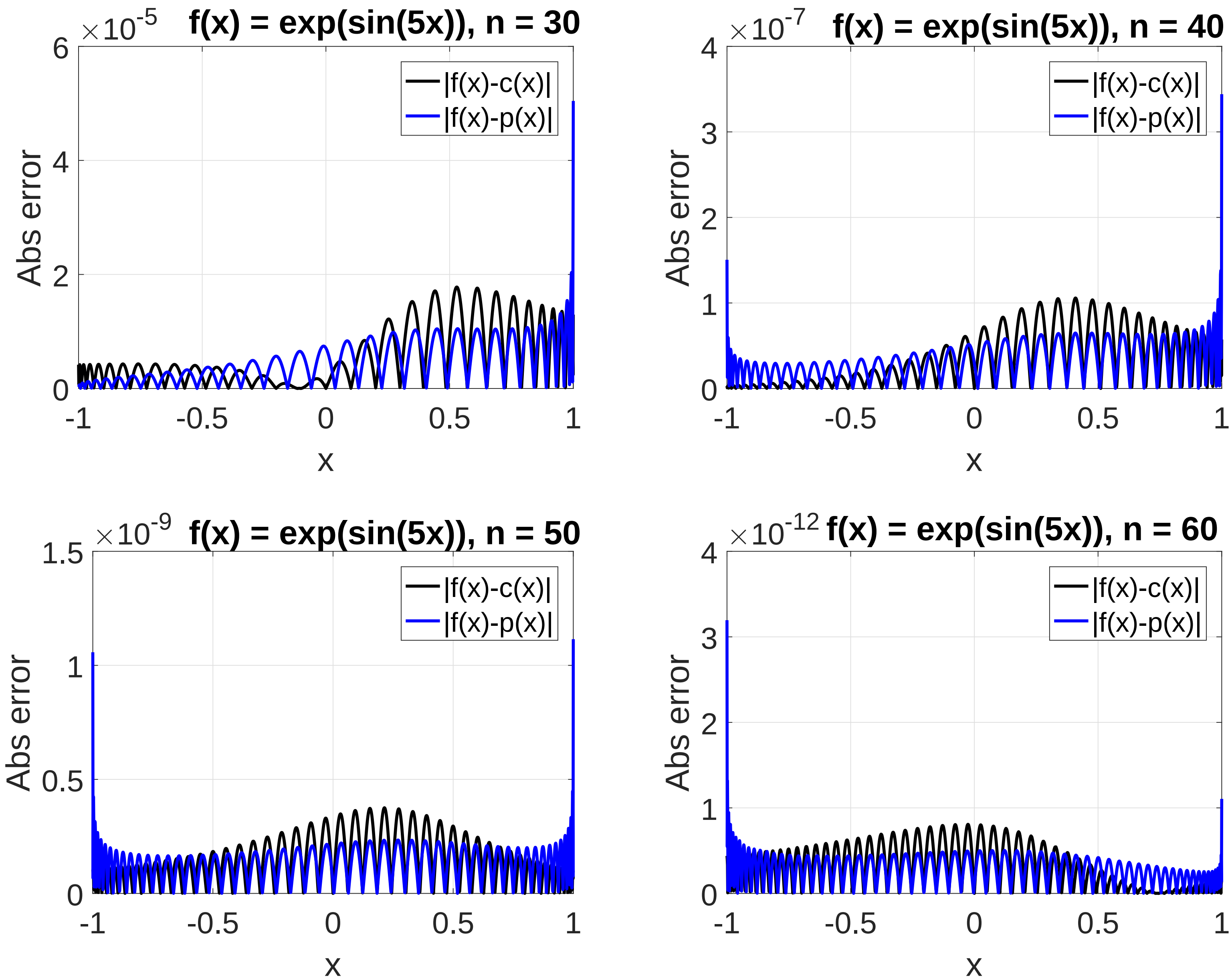
Note that fitting a n-th order polynomial requires using $(n+1)$ data points which are solutions to a $(n+1)$-th order Chebyshev polynomial.

An Example


f(x) = exp(sin(5x)), n = 3
f(x) = exp(sin(5x)), n = 5
f(x) = exp(sin(5x)), n = 10
f(x) = exp(sin(5x)), n = 15
f(x)
c(x)
x
-1
-0.5
0
0.5
1
2
3
4

Chebyshev Approximation Is Only Near-MiniMax




f(x) = exp(sin(5x)), n = 30
×10^-5
|f(x)-c(x)|
|f(x)-p(x)|
Abs error
x
f(x) = exp(sin(5x)), n = 40
×10^-7
|f(x)-c(x)|
|f(x)-p(x)|
Abs error
x
f(x) = exp(sin(5x)), n = 50
×10^-9
|f(x)-c(x)|
|f(x)-p(x)|
Abs error
x
f(x) = exp(sin(5x)), n = 60
×10^-12
|f(x)-c(x)|
|f(x)-p(x)|
Abs error
x

Numerical Integration

The task is to compute the definite integral:

$$I = \int_a^b f(x)dx = F(b) - F(a), \qquad \text{where } F'(x) = f(x)$$

when it is not possible (or inconvenient) to find $F(x)$.

Can we resort to Taylor expansion of $f(x)$? Yes, but often it is less convenient and we have to worry about convergence issue.

It is usually convenient to first approximate $f(x)$ by interpolatory polynomials and then integrate these piece-wise polynomials.

Taylor Approximation

For example, $f(x) = e^x$. Then

$$\int_a^b e^x dx \approx \int_a^b \sum_{j=0}^{n} \frac{x^n}{n!} dx = \sum_{j=0}^{n} \int_a^b \frac{x^j}{j!} dx = \sum_{j=0}^{n} \left. \frac{x^{j+1}}{(j+1)!} \right|_a^b$$

$$= \sum_{j=0}^{n} \frac{b^{j+1}}{(j+1)!} - \sum_{j=0}^{n} \frac{a^{j+1}}{(j+1)!}$$

This is a “perfect” example. In general, however

- We need to make sure Taylor series convergence in the range of $[a,\ b]$. Otherwise, it would not work, for example, one can not exchange summation with integration.
- We need to choose a good point (or several points) around which to expand the Taylor series, otherwise n will be very large in order to achieve sufficient accuracy.

Linear Interpolation and Trapezoidal Rule

For $f(x)$ in $[a,\ b]$, we can approximate it by a straight line

$$P_1(x) = \frac{(b-x)f(a) + (x-a)f(b)}{b-a}$$

then we can approximate the integral

$$\begin{aligned}
I = \int_a^b f(x)dx &\approx \int_a^b P_1(x)dx \\
&= \frac{bf(a) - af(b)}{b-a}x + \frac{f(b)-f(a)}{2(b-a)}x^2\Big|_a^b \\
&= bf(a) - af(b) + \frac{f(b)-f(a)}{2}(b+a) \\
&= \frac{b-a}{2}\left(f(a)+f(b)\right) = T_1(f)
\end{aligned}$$

Of course, we can improve the accuracy by adding points, for example, $c = \frac{a+b}{2}$, and integral each straight line separately. That is

$$
\begin{aligned}
I = \int_a^b f(x)dx &= \int_a^c f(x)dx + \int_c^b f(x)dx \\
&\approx \frac{c-a}{2}\left(f(a) + f(c)\right) + \frac{b-c}{2}\left(f(b) + f(c)\right) \\
&= \frac{b-a}{4}\left(f(a) + f(c)\right) + \frac{b-a}{4}\left(f(b) + f(c)\right) \\
&= \frac{b-a}{4}\left(f(a) + 2f(c) + f(b)\right) \\
&= \frac{h}{2}\left(f(a) + 2f(c) + f(b)\right) = T_2(f)
\end{aligned}
$$

where $h = \frac{b-a}{2}$ is the step size.

In general, we can choose a step size h to break the interval evenly into n sub-intervals. We name the $n-1$ interior points in sorted order as

$$a < x_1 < x_2 < ... < x_{n-1} < b$$

where $x_{i+1} - x_i = h = \frac{b-a}{n}$. The Trapezoidal Rule becomes

$$\begin{aligned}
I = &\int_a^b f(x)dx \\
\approx &h\left[f(a)/2 + f(x_1) + f(x_2) + ... + f(x_{n-1}) + f(b)/2\right] \\
= &T_n(f)
\end{aligned}$$

Error of Trapezoidal Rule

Theorem: Let $f(x)$ have two continuous derivatives on the interval $a \le x \le b$. Then

$$E_n^T(f) = \int_a^b f(x)dx - T_n(f) = -\frac{h^2}{12}(b-a)f''(z) = -\frac{(b-a)^3}{12n^2}f''(z)$$

for some z in the interval $[a,\ b]$.

Think about the negative sign, h^2, and f''

A (Heuristic) Explanation of Error Formula

By Taylor expansion, we have

$$f(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2}f''$$

$$f(b) = f(a) + (b-a)f'(a) + \frac{(b-a)^2}{2}f''$$

Then, we can show the error in one interval:

$$\begin{aligned}
\int_a^b f(x)dx =& (b-a)f(a) + \frac{(b-a)^2}{2}f'(a) + \frac{(b-a)^3}{6}f'' \\
=& (b-a)f(a) + (f(b) - f(a))\frac{b-a}{2} - \frac{(b-a)^3}{4}f'' + \frac{(b-a)^3}{6}f'' \\
=& \frac{b-a}{2}(f(a)+f(b)) - \frac{(b-a)^3}{12}f'' \\
=& T_1(f) - \frac{(b-a)^3}{12}f''
\end{aligned}$$

With n intervals (i.e., $h = \frac{b-a}{n}$), we have

$$
\begin{aligned}
E_n^T(f) &= \int_a^b f(x)dx - T_n(f) \\
&= -\frac{h^3}{12} f''(z_1) - \frac{h^3}{12} f''(z_2) - ... - \frac{h^3}{12} f''(z_n) \\
&= -\frac{h^3}{12} n \frac{f''(z_1) + f''(z_2) + ... + f''(z_n)}{n} \\
&= -\frac{h^3}{12} n f''(z) \\
&= -\frac{h^2}{12} (b-a) f''(z)
\end{aligned}
$$

An Example

$$I(f) = \int_0^\pi e^x \cos x dx = \int_0^\pi \cos x de^x = - e^\pi - 1 + \int_0^\pi e^x \sin x dx$$
$$= - e^\pi - 1 - \int_0^\pi e^x \cos x dx$$

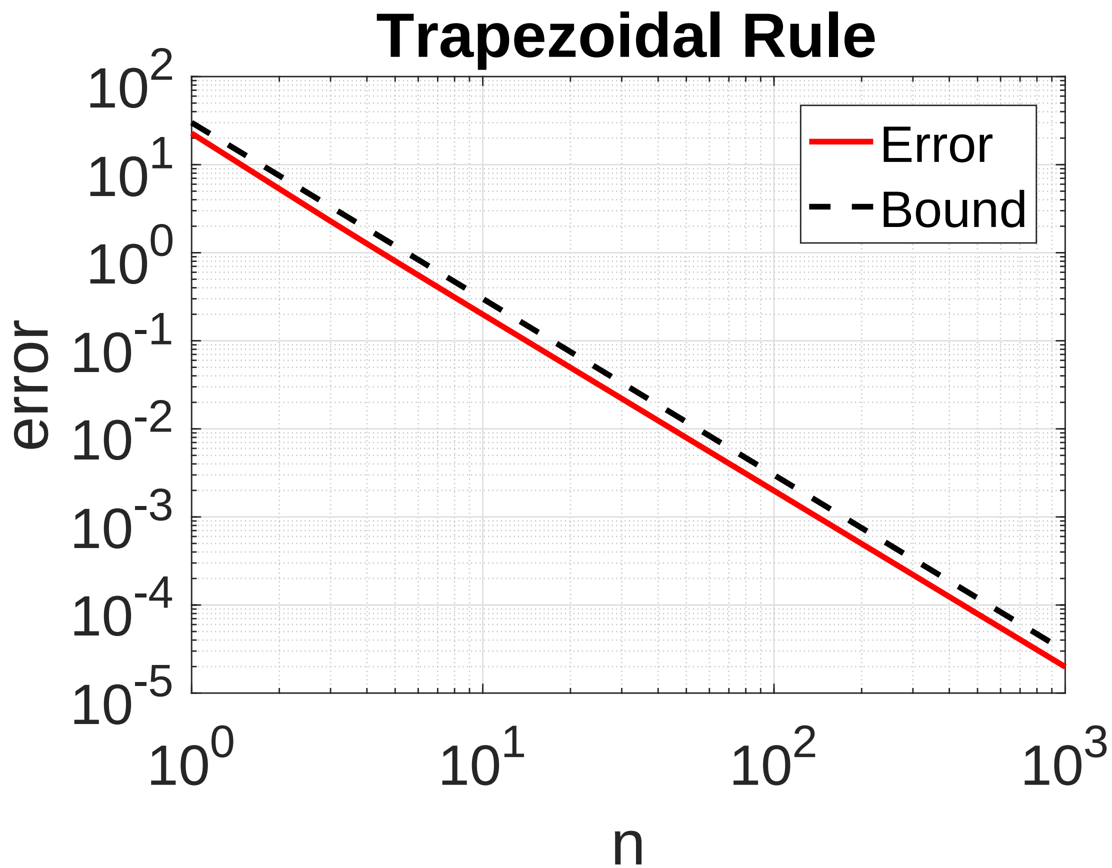
Thus

$$I(f) = -\frac{e^\pi + 1}{2}$$

$$f'(x) = e^x \cos x - e^x \sin x, \qquad f''(x) = -2e^x \sin x, \qquad \max |f''(x)| < 15$$

Therefore,

$$|E_n^T(f)| = |I(f) - T_n(f)| < \frac{\pi^3}{12n^2} 15 < \frac{39}{n^2}$$


Trapezoidal Rule
error
n
Error
Bound
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^0
10^1
10^2
10^3

Quadratic Interpolation

Suppose $c = \frac{a+b}{2}$. Using quadratic interpolation, we will obtain

$$I = \int_a^b f(x)dx \approx \frac{h}{3}\left(f(a) + 4f(c) + f(b)\right) = S_2(f)$$

where $h = \frac{b-a}{2}$ is the step size.

Given three points, a, c, b, with $c - a = b - c = h$, the quadratic interpolation polynomial is

$$P_2(x) = \frac{(x-c)(x-b)}{2h^2} f(a) + \frac{(x-a)(x-c)}{2h^2} f(b) - \frac{(x-a)(x-b)}{h^2} f(c)$$

$$S_2(f) = \int_a^b P_2(x)dx = \frac{h}{3}\left(f(a) + 4f(c) + f(b)\right)$$

Let n be an even number, and

$$h = \frac{b-a}{n}, \qquad x_j = a + jh, \;\; j = 0, 1, 2, ..., n$$

$$S_n(f) = \frac{h}{3} \times$$
$$(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + ...2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n))$$

This is called Simpson's Rule.

Error of Simpson's Rule

Theorem: Let $f(x)$ have two continuous derivatives on the interval $a \leq x \leq b$. Then

$$E_n^S(f) = \int_a^b f(x)dx - S_n(f) = -\frac{h^4}{180}(b-a)f^{(4)}(z) = -\frac{(b-a)^5}{180n^4}f^{(4)}(z)$$

for some z in the interval $[a,\ b]$.

An Example

$$I(f) = \int_0^\pi e^x \cos x dx = -\frac{e^\pi + 1}{2}$$

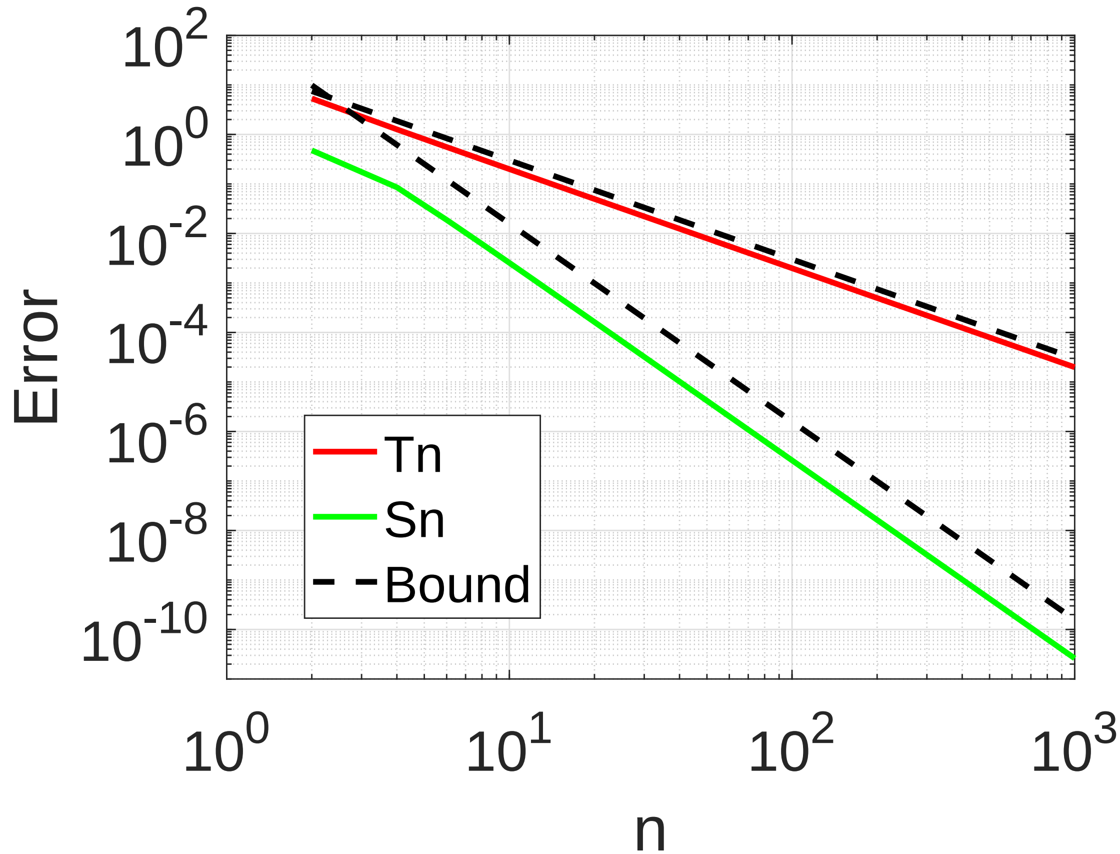
$$f'(x) = e^x \cos x - e^x \sin x, \qquad f''(x) = -2e^x \sin x,$$

$$f^{(3)}(x) = -2e^x \sin x - 2e^x \cos x \qquad f^{(4)}(x) = -4e^x \cos x,$$

$$\max |f^{(4)}(x)| < 93$$

Therefore,

$$|E_n^S(f)| = |I(f) - S_n(f)| < \frac{\pi^5}{180n^4} 93 < \frac{158}{n^4}$$



10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
Error
Tn
Sn
Bound
10^{0}
10^{1}
10^{2}
10^{3}
n

Another Example

$$I(f) = \int_0^1 \frac{1}{1+x^2} dx = \tan^{-1} x\Big|_0^1 = \frac{\pi}{4}$$

$$f'(x) = \frac{-2x}{(1+x^2)^2}$$

$$f''(x) = \frac{-2(1+x^2)^2 + 8x^2(1+x^2)}{(1+x^2)^4} = \frac{-2+6x^2}{(1+x^2)^3}$$

$$f^{(3)}(x) = \frac{12x(1+x^2)^3 - (-2+6x^2)3(1+x^2)^2 2x}{(1+x^2)^6} = \frac{24x - 24x^3}{(1+x^2)^4}$$

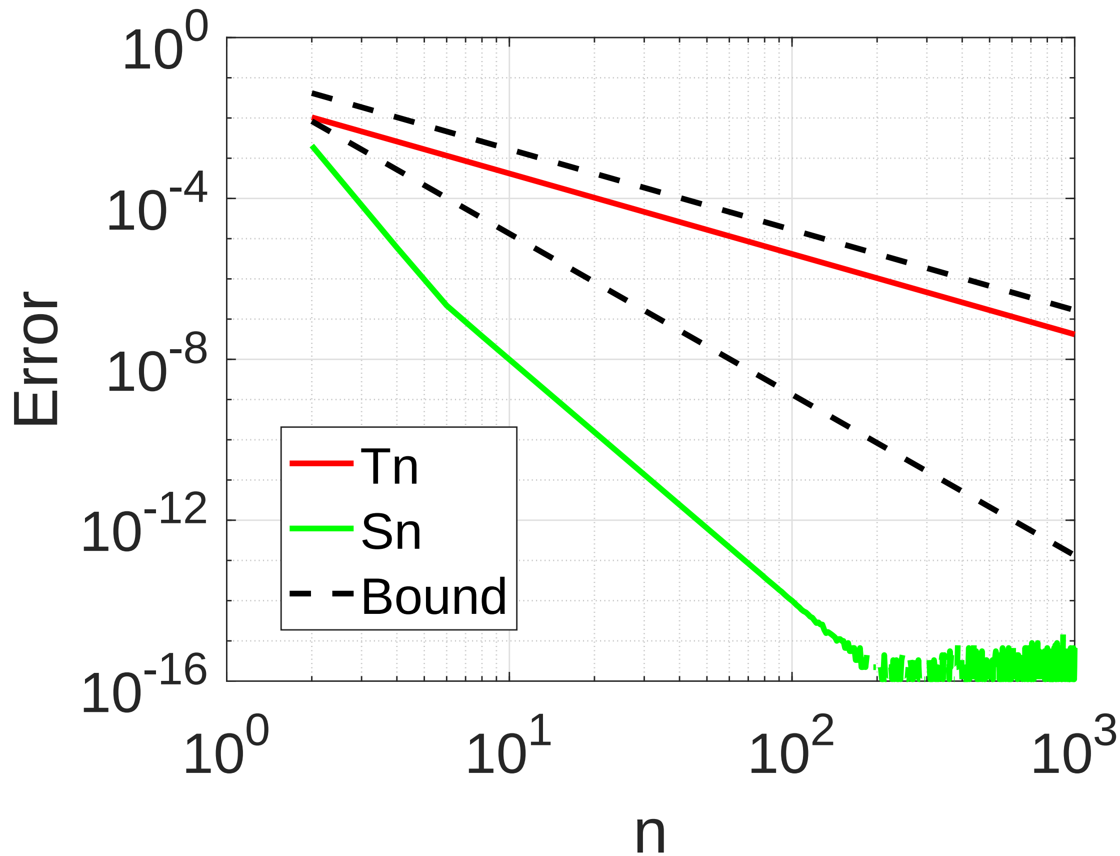
$$f^{(4)}(x) = \frac{(24-72x^2)(1+x^2)^4 - (24x-24x^3)4(1+x^2)^3 2x}{(1+x^2)^8}$$

$$= \frac{24(1-10x^2+5x^4)}{(1+x^2)^5}$$

$$\max|f''(x)| < 2, \qquad \max|f^{(4)}(x)| < 24$$

$$|E_n^T(f)| = |I(f) - T_n(f)| < \frac{1}{12n^2}2 = \frac{1}{6n^2}$$

$$|E_n^S(f)| = |I(f) - S_n(f)| < \frac{1}{180n^4}24 = \frac{2}{15n^4}$$



Note the additional errors caused by machine precision (matlab eps = 2.2204e-16).

Gaussian Quadrature: Another Approach for Numerical Integration

We would like to find data points x_i, $i = 1$ to n and weights w_i, $i = 1$ to n, to directly approximate the integral of $f(x)$ in the interval of $[-1,\ 1]$. That is

$$\int_{-1}^{1} f(x) = \sum_{i=1}^{n} w_i f(x_i)$$

- We need $2n$ equations and hence $2n$ constraints. This can be done by requiring the quadrature formula to be exact when $f(x) = x^j$, $j = 0, 1, 2, ..., 2n - 1$.
- The general interval $[a,\ b]$ can be converted to $[-1,\ 1]$ by a change of variable.
- Gaussian quadrature: is a very good idea when $f(x)$ is close to be a polynomial.

$\boxed{\textbf{The Case of } n = 1}$

To determine $\int_{-1}^{1} f(x) = w_1 f(x_1)$, it requires 2 constraints: $f(x) = x^j$, $j = 0$, 1.

$$\int_{-1}^{1} x^0 dx = 2 = w_1 x_1^0 \Longrightarrow w_1 = 2$$

$$\int_{-1}^{1} x^1 dx = 0 = w_1 x_1 \Longrightarrow x_1 = 0$$

Therefore, Gaussian quadrature for $n = 1$ becomes

$$\int_{-1}^{1} f(x) = 2f(0)$$

This is the mid-point rule, which understandably would not be so accurate in general.

The Case of $n = 2$

To determine $\int_{-1}^{1} f(x) = w_1 f(x_1) + w_2 f(x_2)$, it requires 4 constraints: $f(x) = x^j$, $i = 0, 1, 2, 3$.

$$(1): \; w_1 x_1^0 + w_2 x_2^0 = \int_{-1}^{1} x^0 dx = 2$$

$$(2): \; w_1 x_1^1 + w_2 x_2^1 = \int_{-1}^{1} x^1 dx = 0$$

$$(3): \; w_1 x_1^2 + w_2 x_2^2 = \int_{-1}^{1} x^2 dx = \frac{2}{3}$$

$$(4): \; w_1 x_1^3 + w_2 x_2^3 = \int_{-1}^{1} x^3 dx = 0$$

Comparing (2) with (4) yields $x_1^2 = x_2^2$, because $w_1 x_1^3 + w_2 x_2 x_1^2 = 0$.

If $x_1 = x_2$, then combining (1) and (2) yields $x_1 = x_2 = 0$, which would violate (3).

Therefore, the solution must be $x_1 = -x_2$, which leads to $w_1 = w_2 = 1$, by (1) and (2).

Then from (3), we know $-x_1 = x_2 = \frac{1}{\sqrt{3}}$.

Therefore, the solution for Gaussian quadrature with $n = 2$ turns out to be

$$w_1 = 1,\ w_2 = 1$$

$$x_1 = -\frac{1}{\sqrt{3}},\ x_2 = \frac{1}{\sqrt{3}}$$

$$\int_{-1}^{1} f(x) \approx f(-1/\sqrt{3}) + f(1/\sqrt{3})$$

For example,

$$\int_{-1}^{1} \frac{dx}{3+x} = \log 4 - \log 2 = \log 2 = 0.6931...$$

$$f(-1/\sqrt{3}) + f(1/\sqrt{3}) = \frac{1}{3 - 1/\sqrt{3}} + \frac{1}{3 + 1/\sqrt{3}} = 0.6923...$$

The Case of General n

$$w_1 x_1^j + w_2 x_2^j + ... + w_n x_n^j = \int_{-1}^{1} x^j = \begin{cases} \frac{2}{j+1}, & j = 0, 2, ..., 2n-2 \\ 0, & j = 1, 3, ..., 2n-1 \end{cases}$$

Properties:

- $x_i = -x_{n-i}, i = 1, 2, ..., n$
- $w_i = w_{n-i}, i = 1, 2, ..., n$
- $w_i > 0, i = 1, 2, ..., n$

Review of Legendre Polynomials

Define the Legendre Polynomials as

$$P_0(x) = 1$$

$$P_n(x) = \frac{1}{n!2^n} \frac{d^n}{dx^n} \left[(x^2 - 1)^n\right]$$

For example

$$P_1(x) = x$$

$$P_2(x) = \frac{1}{2}(3x^2 - 1)$$

$$P_3(x) = \frac{1}{2}(5x^3 - 3x)$$

$$P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3)$$

Let $P_1(x) = 0$, then $x = 0$. Let $P_2(x) = 0$, then $x = \pm\frac{1}{\sqrt{3}}$.

Gauss - Legendre Quadrature

It turns out that for general n, the solution can be written in terms of Legendre Polynomials $P_n(x)$:

$$x_i = i\text{-th root of } P_n$$

$$w_i = \frac{2}{(1-x_i^2)P_n'(x_i)}$$

For example, when $n = 3$, solving for $P_3(x) = 0$ yields

$$x_1 = 0,\ x_2 = -\sqrt{\frac{3}{5}},\ x_3 = \sqrt{\frac{3}{5}}$$

which allows to obtain w_i easily

$$w_1 = \frac{8}{9},\ w_2 = \frac{5}{9},\ w_3 = \frac{5}{9}$$

This way, obtain the approximation

$$\int_{-1}^{1} f(x) \approx \frac{8}{9} f(0) + \frac{5}{9} f(-\sqrt{3/5}) + \frac{5}{9} f(\sqrt{3/5})$$

For example,

$$\int_{-1}^{1} \frac{dx}{3+x} = \log 4 - \log 2 = \log 2 = 0.69314718...$$

$$\frac{8}{9}\frac{1}{3} + \frac{5}{9}\frac{1}{3-\sqrt{3/5}} + \frac{5}{9}\frac{1}{3+\sqrt{3/5}} = 0.69312169...$$

General Interval

For a general integral $\int_a^b f(x)dx$, using a change of variable

$$x = \frac{b+a+t(b-a)}{2}, \qquad -1 \leq t \leq 1$$

we have

$$\int_a^b f(x)dx = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b+a+t(b-a)}{2}\right) dt$$

For example, when $n = 1$, the mid-point rule of Gauss - Legendre Quadrature becomes

$$\begin{aligned}
\int_a^b f(x)dx =& \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b+a+t(b-a)}{2}\right) dt \\
\approx& \frac{b-a}{2} 2f\left(\frac{b+a+0(b-a)}{2}\right) \\
=& (b-a)f\left(\frac{b+a}{2}\right)
\end{aligned}$$

When $n = 2$, the formula becomes

$$
\begin{aligned}
\int_a^b f(x)dx =& \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b+a+t(b-a)}{2}\right) dt \\
\approx& \frac{b-a}{2} f\left(\frac{b+a-(b-a)/\sqrt{3}}{2}\right) + \frac{b-a}{2} f\left(\frac{b+a+(b-a)/\sqrt{3}}{2}\right)
\end{aligned}
$$

An interesting idea: can we divide the interval to n sub-intervals and apply Gauss - Legendre Quadrature in each sub-interval?

Let $h = \frac{b-a}{n}$ (assuming n is even) and denote

$$x_i = a + h \times i, \quad i = 0, 1, 2, ..., n$$

Then Trapzoidal rule becomes

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

$$S_n(f) = \frac{h}{3}\times$$
$$(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + ...2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n))$$

Both T_n and S_n involve $n+1$ function evaluations. Therefore, we can divide $[a,\ b]$ into $\frac{n}{2}$ sub-intervals and apply the "$n = 2$" Gauss - Legendre Quadrature formula for each sub-interval, with a total number of function evaluation being n.

Let $h' = \frac{b-a}{n/2}$ (assuming n is even) and denote

$$x_i = a + h' \times i, \quad i = 0, 1, 2, ..., n/2$$

Also, we denote $h = h'/2$ (to compare with T_n and S_n). Then

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx hf\left(\frac{x_i + x_{i+1} - 2h/\sqrt{3}}{2}\right) + hf\left(\frac{x_i + x_{i+1} + 2h/\sqrt{3}}{2}\right)$$

This way, we obtain an approximation formula for $\int_a^b f(x)$:

$$\begin{aligned}\int_a^b f(x) \approx & GL_{2,n}(f) \\ = & h \sum_{i=0}^{n/2-1} \left[f\left(\frac{x_i + x_{i+1} - 2h/\sqrt{3}}{2}\right) + f\left(\frac{x_i + x_{i+1} + 2h/\sqrt{3}}{2}\right)\right]\end{aligned}$$

Suppose we hope to further improve the accuracy, we can use "$n = 3$" Gauss - Legendre Quadrature on each sub-interval. Let $h' = \frac{b-a}{n/3}$ (assuming n is divisible by 3) and denote

$$x_i = a + h' \times i, \quad i = 0, 1, 2, ..., n/3$$

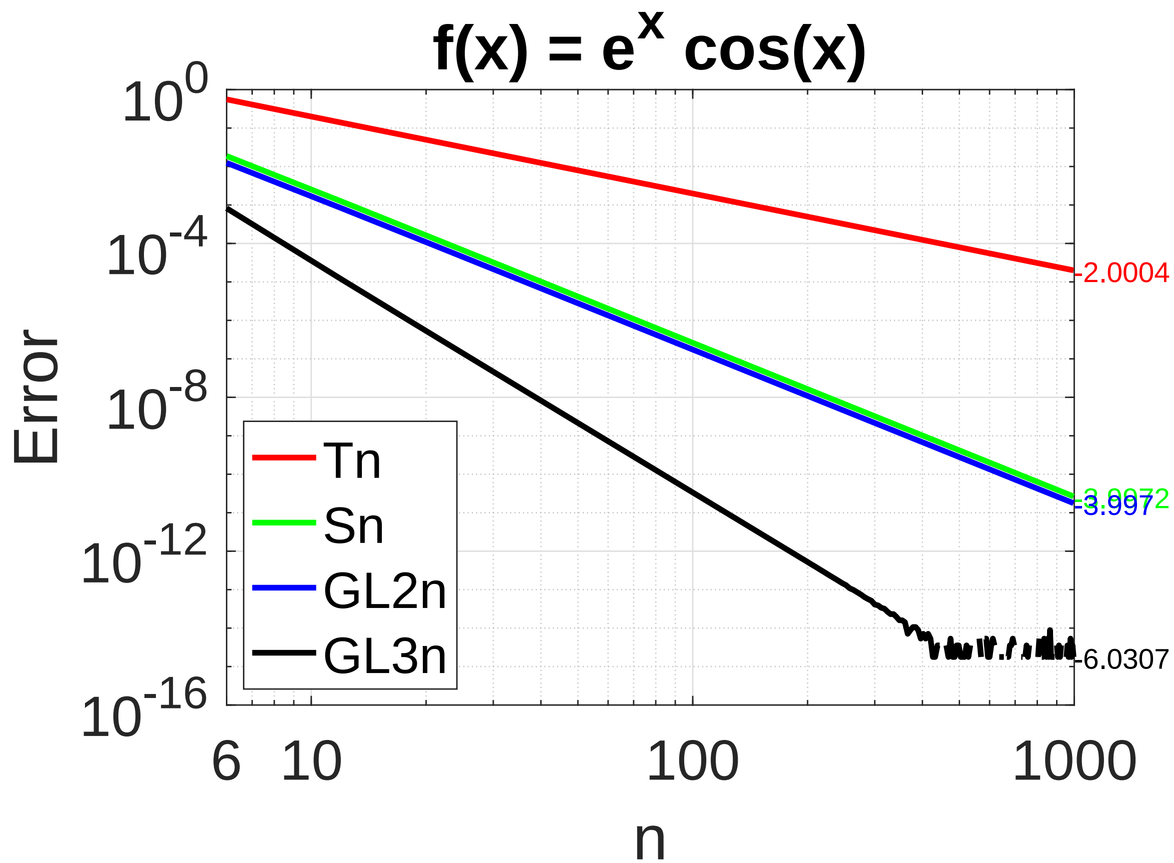
Also, we denote $h = h'/3$ (to compare with T_n and S_n). Then

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \frac{3h}{2} \times$$

$$\left(\frac{5}{9} f\left(\frac{x_i + x_{i+1} - 3h\sqrt{3/5}}{2} \right) + \frac{8}{9} f\left(\frac{x_i + x_{i+1}}{2} \right) + \frac{5}{9} f\left(\frac{x_i + x_{i+1} + 3h\sqrt{3/5}}{2} \right) \right)$$

$$\int_a^b f(x) \approx GL_{3,n}(f) = \frac{3h}{2} \sum_{i=0}^{n/3-1}$$

$$\left[\frac{5}{9} f\left(\frac{x_i + x_{i+1} - 3h\sqrt{3/5}}{2} \right) + \frac{8}{9} f\left(\frac{x_i + x_{i+1}}{2} \right) + \frac{5}{9} f\left(\frac{x_i + x_{i+1} + 3h\sqrt{3/5}}{2} \right) \right]$$



In this example, it is expected that $GL_{2,n}$ will be roughly similar to Simpson's rule because both use quadratic approximation. $GL_{3,n}$ is substantially more accurate. We also print out the slopes (in the log scale) to show that the errors are, respectively, $\frac{1}{n^2}$, $\frac{1}{n^4}$, $\frac{1}{n^4}$, and $\frac{1}{n^6}$. The numbers on the figure are determined by least square (using points with errors $> 10^{-14}$).

One More Approach: Numerical Integration by Monte Carlo

$$I = \int_a^b f(x)dx = (b-a)\int_a^b f(x)\frac{1}{b-a}dx = (b-a)\int_a^b f(x)g(x)dx$$

where

$$g(x) = \frac{1}{b-a}, \text{ if } x \in [a,\ b], \qquad g(x) = 0, \text{ otherwise}$$

In other words, $g(x)$ is the density function of a uniform distribution on $[a,\ b]$. Therefore, we can write the integral as an expectation

$$I = \int_a^b f(x)dx = (b-a)\int_a^b f(x)g(x)dx = (b-a)E(f(x))$$

Expectation can be understood as average with an infinite number of samples. If we sample n x's from a uniform distribution on $[a,\ b]$, then the average is approximately the expectation:

$$I = (b-a)E(f(x)) \approx \frac{(b-a)}{n}\sum_{i=1}^{n} f(x_i) = M_n(f)$$

Error analysis: using statistics terminology,

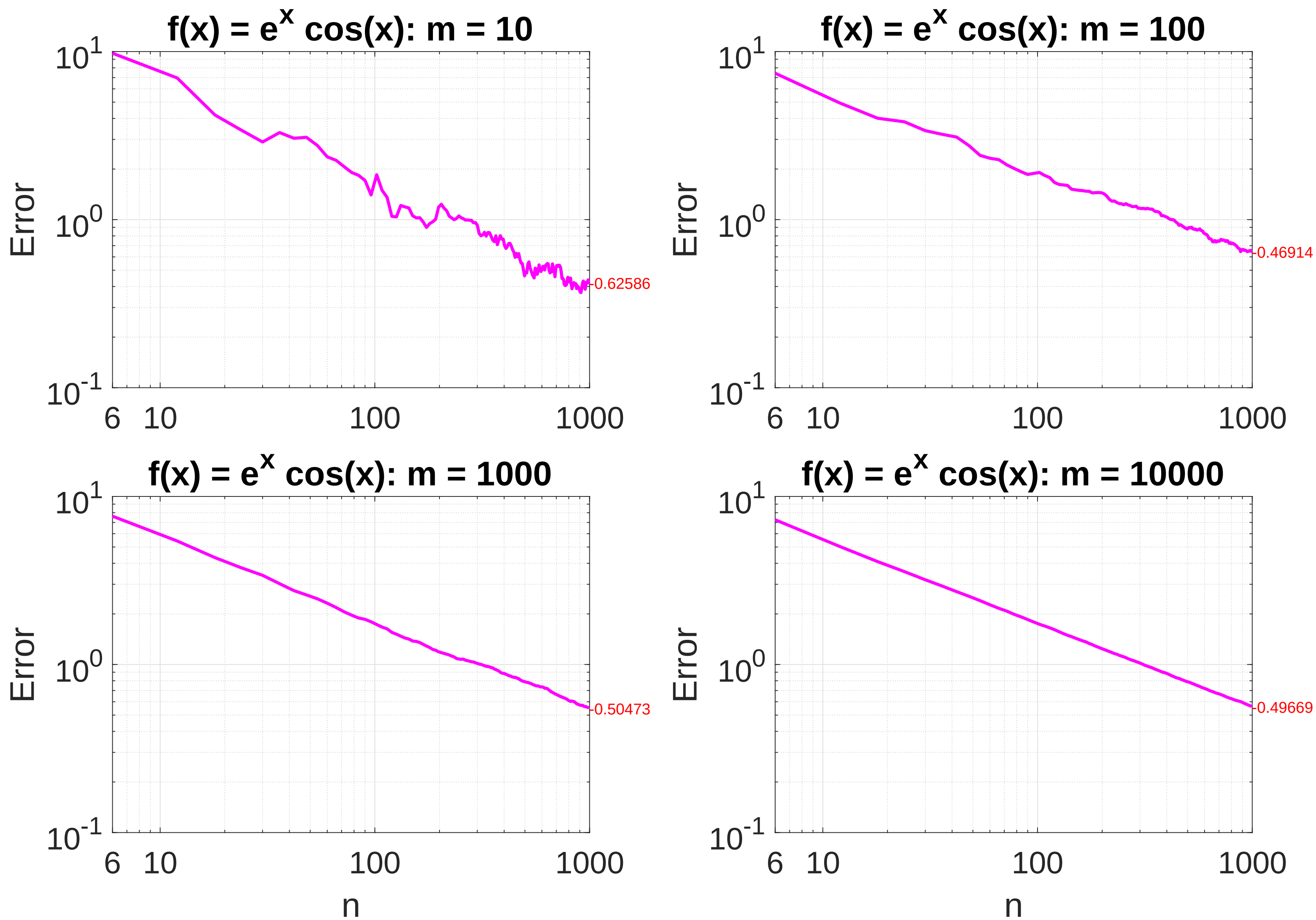
$$E(M_n(f)) = \frac{(b-a)}{n}\sum_{i=1}^{n} E(f(x_i)) = (b-a)E(f(x)) = I$$

$$E((M_n(f)-I)^2) = Var\left(M_n(f)\right) = \frac{1}{n}Var\left(f(x)\right) = O\left(\frac{1}{n}\right)$$

One can show, as verified by simulations in the next slide:

$$E(|M_n(f)-I|) = O\left(\frac{1}{\sqrt{n}}\right)$$

Note that, due to the randomness, for each n, we need to simulate m times in order to approximate $E(|M_n(f)-I|)$.



While it appears Monte-Carlo is a poor method for numerical integration, it does have many advantages, especially in high-dimensions. All examples we have seen are 1-dimensional.

Linear Regression

Given data $\{x_i, y_i\}_{i=1}^n$, where x_i is a p-dimensional vector and y_i is a scalar.

We construct the data matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,p} \\ \ldots & & & & \\ 1 & x_{n,1} & x_{n,2} & \ldots & x_{n,p} \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}$$

The data model is

$$\mathbf{y} = \mathbf{X} \times \beta$$

β (a vector of length $p+1$) is obtained by minimizing the square error (equivalent to maximizing the joint likelihood under the normal distribution model).

Linear Regression Estimation by Least Square

The idea is to minimize the square error

$$SE(\beta) = \sum_{i=1}^{n} |y_i - x_i\beta|^2 = (\mathbf{Y} - \mathbf{X}\beta)^{\mathsf{T}} (\mathbf{Y} - \mathbf{X}\beta)$$

We can find the optimal β by setting the first derivative to be zero

$$\begin{aligned} \frac{\partial SE(\beta)}{\beta} &= \mathbf{X}^{\mathsf{T}} (\mathbf{Y} - \mathbf{X}\beta) = 0 \\ \Longrightarrow & \mathbf{X}^{\mathsf{T}}\mathbf{Y} = \mathbf{X}^{\mathsf{T}}\mathbf{X}\beta \\ \Longrightarrow & \beta = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathsf{T}}\mathbf{Y} \end{aligned}$$

Don't worry much about how to do matrix derivatives. The trick is to view this simply as a scalar derivative but we need to manipulate the order (and add transposes) to get the dimensions correct.

Linear Regression with Only One Variable

Given data $\{x_i, y_i\}_{i=1}^{n}$, where both x_i and y_i are scalar. The goal is to fit a straight line

$$y = \beta_0 + \beta_1 x$$

Again, we minimize the square error

$$SE(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

We can find the optimal β_0, β_1 by setting their first derivative to be zero

$$\frac{\partial SE}{\beta_0} = -\sum_{i=1}^{n} 2\left(y_i - \beta_0 - \beta_1 x_i\right) = 0$$

$$\frac{\partial SE}{\beta_1} = -\sum_{i=1}^{n} 2\left(y_i - \beta_0 - \beta_1 x_i\right) x_i = 0$$

The solution turns out to be

$$\beta_1 = \frac{n\sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2}$$

$$\beta_0 = \frac{\sum_{i=1}^n y_i}{n} - \beta_1 \frac{\sum_{i=1} x_i}{n}$$

β_0 is called “intercept” and β_1 is called “slope”.

The solution is of course the same as the matrix formula by letting $p = 1$.

Ridge Regression

Similar to l_2-regularized logistic regression, we can add a regularization parameter

$$\beta = (\mathbf{X}^{\intercal}\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}}\mathbf{X}^{\intercal}\mathbf{Y}$$

which is known as **ridge regression**.

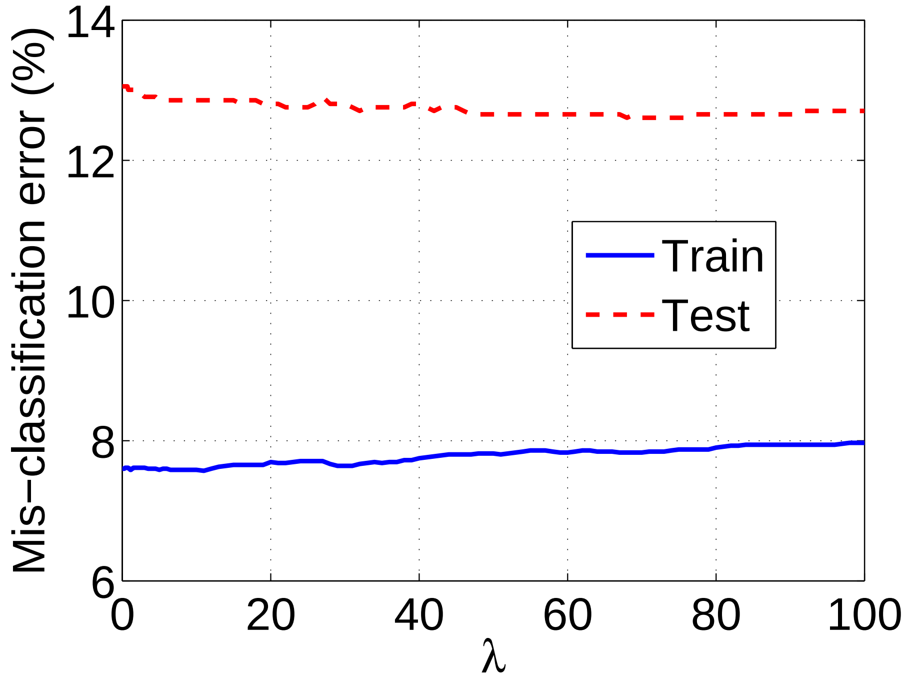
Adding regularization not only improves the numerical stability but also often increases the test accuracy.

Linear Regression for Classification

For binary classification, i.e., $y_i \in \{0, 1\}$, we can simply treat y_i as numerical response and fit a linear regression. To obtain the classification result, we can simply use $\hat{y} = 0.5$ as the classification threshold.

Multi-class classification (with K classes) is more interesting. We can use exactly the same trick as in multi-class logistic regression by first expanding the y_i into a vector of length K with only one entry being 1 and then fitting K binary linear regressions simultaneously and using the location of the maximum fitted value as the class label prediction. Since you have completed the homework in multi-class logistic regression, this idea should be straightforward.

Mis-Classification Errors on Zipcode Data



- This is essentially the first iteration of multi-class logistic regression. Clearly, the results are not as good as logistic regression with many iterations.
- Adding regularization (λ) slightly increases the training errors but decreases the testing errors at certain range.

Beyond Linear Regression: Basis Expansions

Consider one-dimensional case. We model the function $f(x)$ as

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x)$$

For example, if $M = 2$, $h_1(x) = 1$, $h_2(x) = x$, then under least square criterion, it becomes linear regression.

Here, we will consider more flexible modeling, by letting $h_m(x)$ be splines.

Cubic Splines with Two Knots

Suppose there are n observations: (x_i, y_i), $i = 1$ to n. We model the data using cubic splines. Suppose we choose two "knots": ξ_1 and ξ_2 ($\xi_1 < x_2$). That is, there are three regions $x \leq \xi_1$, $\xi_1 < x \leq \xi_2$, $x > \xi_2$, and there is a cubic curve in each region. These three cubic curves should satisfy the usual continuity and smoothness constraints.

Without giving a proof, we can show

$$
\begin{aligned}
&f(x) = \sum_{m=1}^{6} \beta_m h_m(x) \\
&h_1(x) = 1,\ h_2(x) = x,\ h_3(x) = x^2, \\
&h_4(x) = x^3,\ h_5(x) = (x - \xi_1)_+^3,\ h_6(x) = (x - \xi_2)_+^3
\end{aligned}
$$

satisfy the constraints. Here $t_+ = t$ if $t > 0$, otherwise $t_+ = 0$.

$$f(\xi_{1-}) = \beta_1 + \beta_2\xi_1 + \beta_3\xi_1^2 + \beta_4\xi_1^3$$

$$f'(\xi_{1-}) = \beta_2 + 2\beta_3\xi_1 + 3\beta_4\xi_1^2$$

$$f''(\xi_{1-}) = 2\beta_3 + 6\beta_4\xi_1$$

$$f(\xi_{1+}) = \beta_1 + \beta_2\xi_1 + \beta_3\xi_1^2 + \beta_4\xi_1^3$$

$$f'(\xi_{1+}) = \beta_2 + 2\beta_3\xi_1 + 3\beta_4\xi_1^2$$

$$f''(\xi_{1+}) = 2\beta_3 + 6\beta_4\xi_1$$

$$f(\xi_{2-}) = \beta_1 + \beta_2\xi_2 + \beta_3\xi_2 + \beta_4\xi_2^3 + \beta_5(\xi_2 - \xi_1)^3$$

$$f'(\xi_{2-}) = \beta_2 + 2\beta_3\xi_2 + 3\beta_4\xi_2^2 + 3\beta_5(\xi_2 - \xi_1)^2$$

$$f''(\xi_{2-}) = 2\beta_3 + 6\beta_4\xi_2 + 6\beta_5(\xi_2 - \xi_1)$$

$$f(\xi_{2+}) = \beta_1 + \beta_2\xi_2 + \beta_3\xi_2 + \beta_4\xi_2^3 + \beta_5(\xi_2 - \xi_1)^3$$

$$f'(\xi_{2+}) = \beta_2 + 2\beta_3\xi_2 + 3\beta_4\xi_2^2 + 3\beta_5(\xi_2 - \xi_1)^2$$

$$f''(\xi_{2+}) = 2\beta_3 + 6\beta_4\xi_2 + 6\beta_5(\xi_2 - \xi_1)$$

Note that

$$\frac{\partial^3}{\partial x^3}(x-\xi_1)_+^3 = \begin{cases} 6 & \text{if } x \geq \xi_1 \\ 0 & \text{if } x < \xi_1 \end{cases}$$

In other words, $f'''(x)$ is not continuous at ξ_1.

Similarly,

$$\frac{\partial^2}{\partial x^2}(x-\xi_1)_+^2 = \begin{cases} 2 & \text{if } x \geq \xi_1 \\ 0 & \text{if } x < \xi_1 \end{cases}$$

In other words, if we include terms like $(x-\xi_1)_+^2$ in $f(x)$, then $f''(x)$ is not continuous at ξ_1. This will violate the constraints required by cubic splines.

Therefore, given two knots ξ_1 and ξ_2, the construction

$$f(x) = \sum_{m=1}^{6} \beta_m h_m(x)$$

$$h_1(x) = 1,\ h_2(x) = x,\ h_3(x) = x^2,$$

$$h_4(x) = x^3,\ h_5(x) = (x - \xi_1)_+^3,\ h_6(x) = (x - \xi_2)_+^3$$

is the function which is cubic and satisfies the constraints required by cubic splines. Other choices which we can think of would not work. Thus, even though we do not provide a proof that the construction $f(x)$ is unique, we should be convinced it is the correct choice.

Cubic Splines with K Knots

In general, we have

$$f(x) = \sum_{m=1}^{4+K} \beta_m h_m(x)$$

$$h_1(x) = 1,\ h_2(x) = x,\ h_3(x) = x^2,\ h_4(x) = x^3$$

$$h_5(x) = (x - \xi_1)_+^3,\ h_6(x) = (x - \xi_2)_+^3,\ \ldots,\ h_{4+K}(x) = (x - \xi_K)_+^3$$

$f(x)$ satisfies the constraints: $f(x)$, $f'(x)$, $f''(x)$ are continuous at all the knots.

The coefficients β can then be identified by a standard linear regression procedure.

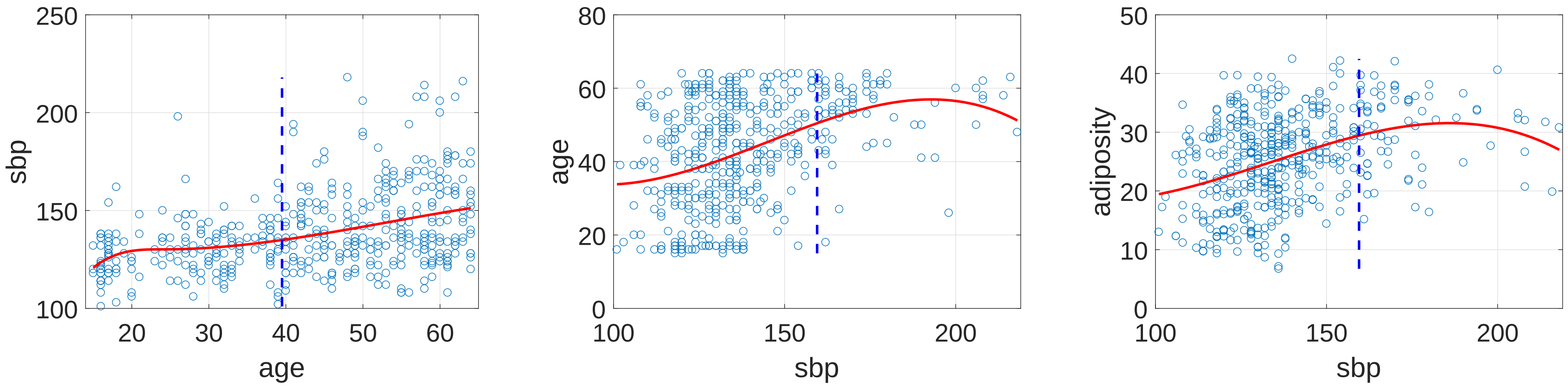
Example: South African Heart Disease

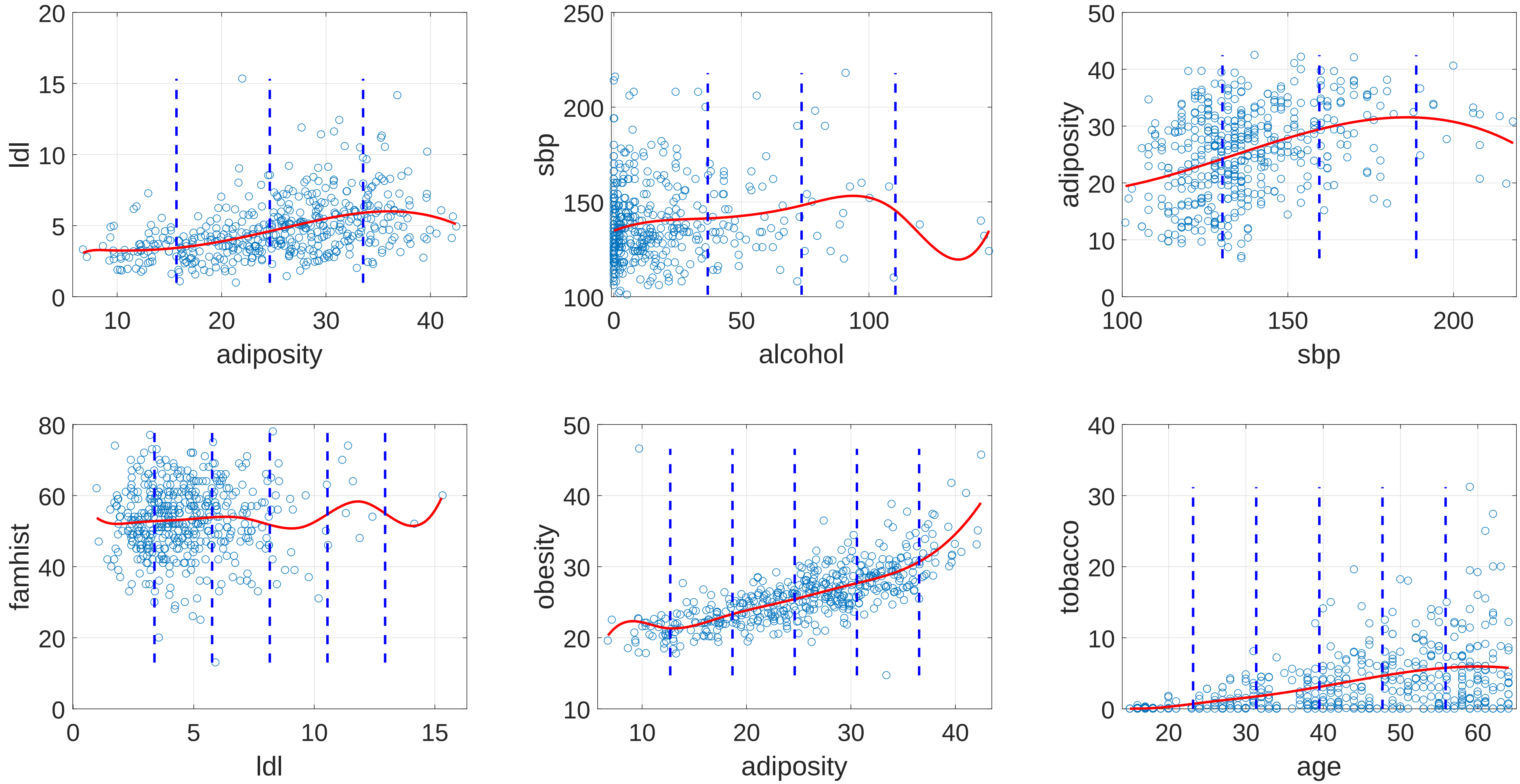
This dataset contains 462 examples. After filtering, there are in total 8 “features”:

```
names = {'sbp','tobacco','ldl','adiposity','famhist','obesity','alcohol','age'};
```

A matlab program is written which randomly selects 2 features (columns) to be “x” and “y” to fit cubic splines. The knots are equally spaced between $\min(x)$ and $\max(x)$.

The next three rows of figures are for $K = 1$, $K = 3$, and $K = 5$, respectively.





20
15
10
5
0
ldl
10
20
30
40
adiposity
250
200
150
100
sbp
0
50
100
alcohol
50
40
30
20
10
0
adiposity
100
150
200
sbp
80
60
40
20
0
famhist
0
5
10
15
ldl
50
40
30
20
10
obesity
10
20
30
40
adiposity
40
30
20
10
0
tobacco
20
30
40
50
60
age

Flexible Modeling

In general,

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x)$$

provides a powerful and flexible framework for modeling concrete data distributions, because we can choose the basis function $h_m(x)$, for example, trees. Note that logistic regression can be made flexible in this framework too, as we can always define a probability model $p(y = 1|x) = \frac{1}{1+e^{-f(x)}}$ (or the multi-class version).

If you wish to learn more about data modeling, or building predictive models, etc, you may consult a classical textbook in machine learning :

The Elements of Statistical Learning: Data Mining, Inference, and Prediction, by Trevor Hastie, Robert Tibshirani, and Jerome Friedman.