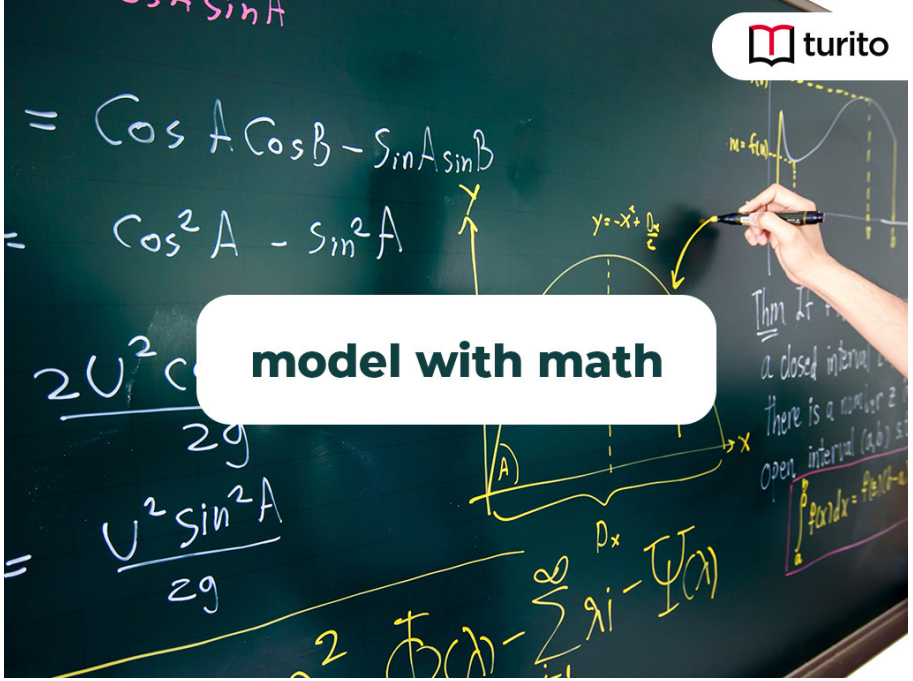


Mathematical Modeling

MAT 3300 Notebook

Youthy WANG

SICK
Sensor Intelligence.

Date: 2021 Sep 9

MAT 3300 Mathematical Modeling

● Introduction

① Process: real-world data —Simplify→ Model —Analyze→ Mathematical conclusions —Interpret→ Predictions / explanations —Verify→ real-world data

② Classification

	Empirical	Mechanistic
Deterministic	regression / relationship	(differential equations)
Stochastic	variance	(probabalistic equations)

③ Construction (process)

(i) Identify the problem

(ii) Make assumptions ☆ { identify & classify the variables; determine interrelationships between variables

(iii) Solve the model

(iv) Verify / Interpret the model { Address problem? Make common sense? Correspond with real-world data?

(v) Implement the model

(vi) Maintain the model (some change)

SICK
Sensor Intelligence.

Date: 2021 Sep 10

④ A modeling example (classic).

☆ "The Secretary Problem" (Socrates' opinion of love)

• Q: a single position / n applicants, the value is known / can be ranked once seen / interview in random order / either accept or reject after each interview / decision only based on previous known rank /

• Aim: highest probability to select the best one

• Strategy chosen (choose a path & go farthest)

↳ reject first r-1 (as ranking sample), accept 1st better than r-1 one. ⇒ choose a good r for highest P_r

$$P(r) = \sum_{i=r}^{n} P(\text{applicant } i \text{ is selected and the best})$$

$$= \sum_{i=r}^{n} P(i \text{ is selected} \mid i \text{ is the best}) P(i \text{ is the best})$$

(the best ... may not have relations / independent)

$$= \sum_{i=r}^{n} P(\text{the best in first } i-1 \text{ is in } r-1 \mid i \text{ the best}) P(i \text{ the best})$$

$$= \sum_{i=r}^{n} \frac{r-1}{i-1} \times \frac{1}{n} = \sum_{r=r}^{n} \frac{r/n - 1/n}{i/n - 1/n} \times \frac{1}{n}$$

let $x = \frac{r}{n}$, when n large $\approx x \int_0^1 \frac{1}{t}\,dt$ (give up $\frac{1}{n}$ in numerator) $= -x \ln x$

$P'(x) = 0 \Rightarrow x = \frac{1}{e} \approx 36.8\%$. $P(\frac{1}{e}) \approx \frac{1}{e} \approx 36.8\%$

SICK
Sensor Intelligence.

Date: 2021 Sep 10

• PART I. Differential Equations (ODE)

① Difference Equation: relates change (n^{th} order difference) to past values in a (discrete) sequence.

$$\begin{cases} \Delta a_n := a_{n+1} - a_n = f(a_0, a_1, \cdots, a_n). \\ \text{another form (dynamic system)} \quad a_{n+1} = g(a_0, a_1, \cdots, a_n) \end{cases}$$

② Differential Equations: relates continous change (approximation to difference equation)

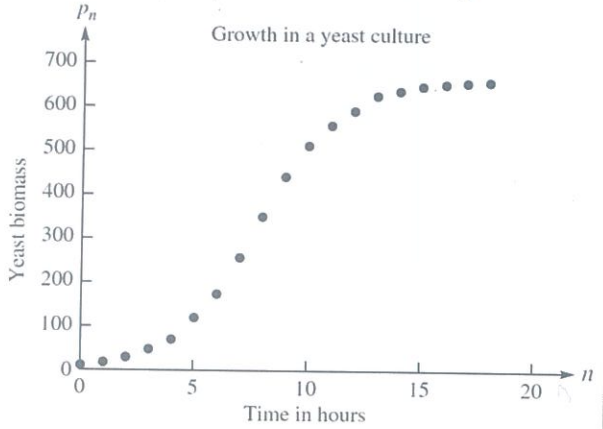
$$a'(t) = f(a(t))$$ (ODE)

Remark: f can be got from either theoretical derivation or model fitting.

eg: (i) Growth of a Yeast Cultivation (Yeasts' "S" shape growth)

Analyze given data (by experiment)
⇒ propose a difference equation

Time in hours n	Yeast biomass p_n	Change/hour $p_{n+1} - p_n$
0	9.6	8.7
1	18.3	10.7
2	29.0	18.2
3	47.2	23.9
4	71.1	48.0
5	119.1	55.5
6	174.6	82.7
7	257.3	93.4
8	350.7	90.3
9	441.0	72.3
10	513.3	46.4
11	559.7	35.1
12	594.8	34.6
13	629.4	11.4
14	640.8	10.3
15	651.1	4.8
16	655.9	3.7
17	659.6	2.2
18	661.8	



Date: 2021 Sep 12

SICK
Sensor Intelligence.

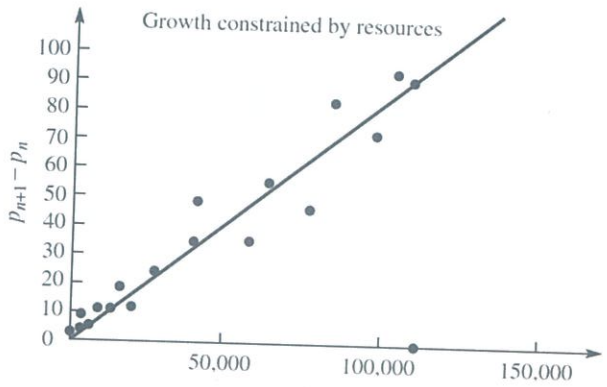
Carrying capacity: 665 due to resource limit

Model: $\Delta p_n := p_{n+1} - p_n = k p_n (665 - p_n)$

difference equation → estimate k using a linear regression (with Δp_n ; p_n, $665 - p_n$)

get $k = 0.00082$ with no analytical sol.

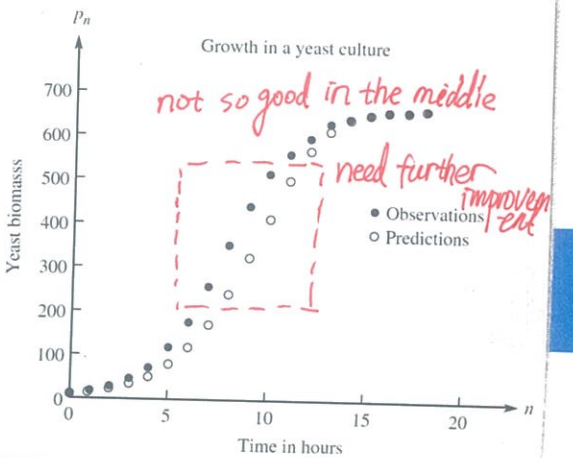
$p_{n+1} - p_n$	$p_n(665 - p_n)$
8.7	6291.84
10.7	11,834.61
18.2	18,444.00
23.9	29,160.16
48.0	42,226.29
55.5	65,016.69
82.7	85,623.84
93.4	104,901.21
90.3	110,225.01
72.3	98,784.00
46.4	77,867.61
35.1	58,936.41
34.6	41,754.96
11.4	22,406.64
10.3	15,507.36
4.8	9050.29
3.7	5968.69
2.2	3561.84



a relatively good model

Test the model – prediction with real data

Time in hours	Observation	Prediction
0	9.6	9.6
1	18.3	14.8
2	29.0	22.6
3	47.2	34.5
4	71.1	52.4
5	119.1	78.7
6	174.6	116.6
7	257.3	169.0
8	350.7	237.8
9	441.0	321.1
10	513.3	411.6
11	559.7	497.1
12	594.8	565.6
13	629.4	611.7
14	640.8	638.4
15	651.1	652.3
16	655.9	659.1
17	659.6	662.3
18	661.8	663.8



SICK
Sensor Intelligence.

Date: 2021 Sep 12

Modification. (estimated by differential equation)

★ Logistic Model. $\frac{dP(t)}{dt} = rP(t)(M-P(t))$, M – maximum possible population

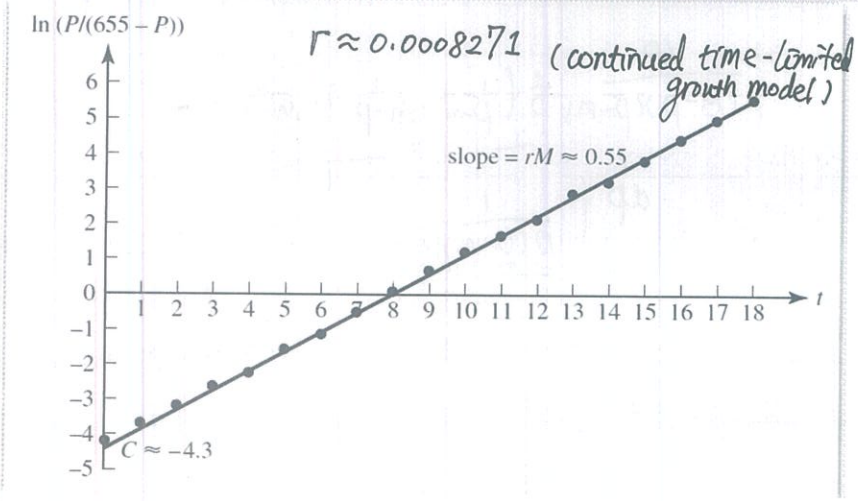
analysis:

$$\frac{dP(t)}{P(t)(M-P(t))} = r\,dt \Rightarrow \frac{1}{M}\left(\frac{dP(t)}{P(t)} - \frac{d(M-P(t))}{M-P(t)}\right) = r\,dt$$

$$\Rightarrow \ln\frac{P(t)}{P(0)} - \ln\frac{M-P(t)}{M-P(0)} = rMt$$

$$\Rightarrow P(t) = \frac{M\,P(0)}{P(0)+(M-P(0))e^{-rMt}} \qquad t\to\infty,\ P(t)\to M$$

getting "r": linear regression $\ln\frac{P(t)}{M-P(t)} = \ln\frac{P(0)}{M-P(0)} + rMt$

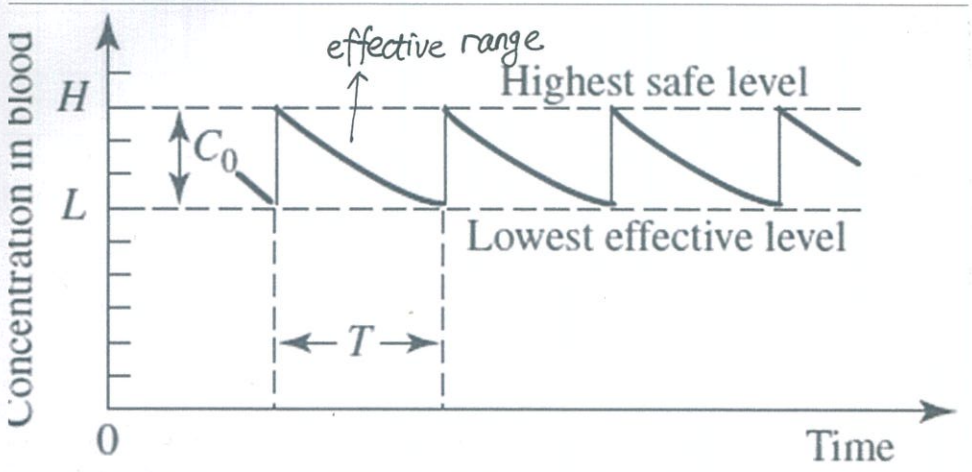


(ii) Prescribing Drug Dosage:

How can doses & time between doses can be adjusted to maintain a safe but effective concentration?

Date: 2021 Sep 12

SICK
Sensor Intelligence.



$C(t) = f$(decay rate, assimilation rate, dosage amount, time)

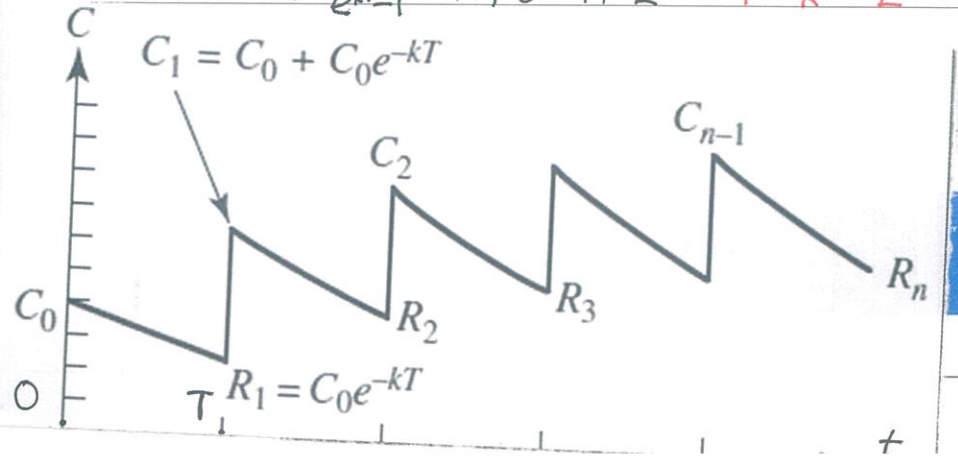
simplify → assume that body weight & blood volume are constants

$\Rightarrow \frac{dC(t)}{dt} = -kC(t)$ (just about total concentrate)

$C(t) = C_0 \cdot e^{-kt}$ after interval T C_0e^{-kT} dose $\to C_0e^{-kT} + C_0 \cdots$

After n^{th} dosage: $R_n = C_0 + C_0e^{-kT} + \cdots + C_0e^{-nkT} \overset{n\to\infty}{=} \frac{C_0}{e^{kT}-1}$

$\Rightarrow \frac{C_0}{e^{kT}-1} = L$, $C_0 = H-L \Rightarrow T = \frac{1}{k}\ln\frac{H}{L}$



SICK
Sensor Intelligence.

Date: 2021 Sep 16

③ Solving some ODEs (methods)

- seperation of variables (Calculus 1)
- first-order linear equation (Calculus 1)
- second-order (or higher) linear equation with constant coefficients (Linear Algebra)

Others with analytical solution: website "Eqworld"

What if not analytically solvable? → Graphically solve / Numerically solve

④ Graphical Solutions:

- (i) phase lines ($\frac{dy}{dx} = g(x,y)$, each tangent line at (x_0, y_0))
- (ii) Autonomous Differential Equations, $\frac{dy}{dx} = g(y)$ (in calculus 1) (equilibrium values / rest points + derivative analyze)
- (iii) ☆ System of ODEs

Some concepts:

- autonomous system of ODEs: $\frac{dx}{dt} = f(x,y)$; $\frac{dy}{dt} = g(x,y)$ (f, g not explicitly (like) contains t)
- Solutions = pairs of function $(x(t), y(t))$ satisfies... in this case
- phase plane = the x-y plane
- trajectory / path / orbit: a curve consists of pts $(x(t), y(t))$ $\forall t \geq 0$
- rest point / equilibrium point: A point s.t. $\frac{dx}{dt} = 0 = \frac{dy}{dt}$

Stable: to describe the rest pts. if any trajectory starting close to it stays close to it for all future time t

Asymptotically stable: ~ (same as stable) if any trajectory starting close to it approaches to it when $t \to \infty$

Unstable (points not stable)

eg 1: (i) a linear autonomous system ($\frac{dx}{dt} = -x+y$; $\frac{dy}{dt} = -x-y$)

(ii) a non-linear autonomous system ($\frac{dx}{dt} = xy$; $\frac{dy}{dt} = x^2$)

↳ $d(y^2 - x^2) = 2y\,dy - 2x\,dx = 0$.

Properties of trajectories (omitted)

egs (i) Competitive Hunter Model

Imagine a small pond, mature enough to support wildlife. Stock the pond with game fish, say trout & bass.

Let $x(t)$ denote the population of the trout at time t, where $y(t)$ denote the bass.

- Q: How sensitive is the final solution of population levels to the initial stockage levels & external perturbations?
- Modeling: Assume 2 species depend on an unlimited common food supply.

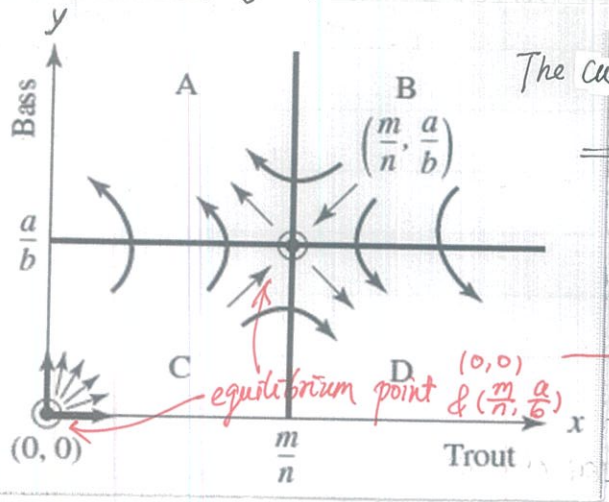
$$\frac{dx}{dt} = (a - by)x \quad ; \quad \frac{dy}{dt} = (m - nx)y$$

compete: each species affects another

SICK
Sensor Intelligence.

Date: 2021 Sep 21

Principle of competitive exclusion: mutual coexistence of the species is highly improbable.



The currency of the curve (shown by graphical sol.) ⇒ principle of competitive exclusion.

equilibrium point $(0,0)$ & $(\frac{m}{n}, \frac{a}{b})$ → properties: both unstable

• Modification: $\begin{cases} \frac{dx}{dt} = (a - \varepsilon x - by)x \\ \frac{dy}{dt} = (m - \eta y - nx)y \end{cases}$ more realistic (if $x \to \infty$, $\frac{dx}{dt} \to -\infty$ decrease)

environment limits

equilibrium points.

$(x,y) = (0,0)$; $(0, \frac{m}{\eta})$; $(\frac{a}{\varepsilon}, 0)$; $\left(\frac{mb - a\eta}{nb - \varepsilon\eta}, \frac{an - m\varepsilon}{nb - \varepsilon\eta}\right)$

all unstable! — it depends whether stable or not

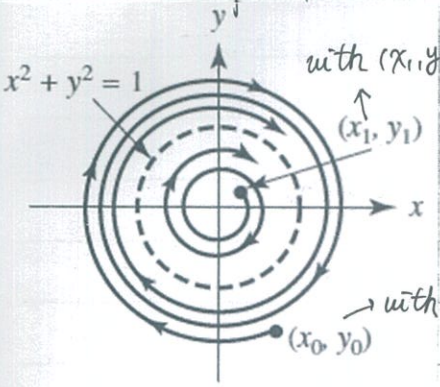
★ Limitations of Graphical Analysis.

Consider $\begin{cases} \frac{dx}{dt} = y + x - x(x^2+y^2) \\ \frac{dy}{dt} = -x + y - y(x^2+y^2) \end{cases}$ (the only equilibrium pt. $(0,0)$)

Analyze: when $x^2+y^2=1$, $2x\frac{dx}{dt} + 2y\frac{dy}{dt} = 0$.

limit circle. $2(x^2+y^2)(1-x^2-y^2)$

Date: 2021 Sep 27



with (x_1, y_1) inside the circle, it will spin outward $(2x\frac{dx}{dt} + 2y\frac{dy}{dt} > 0)$ periodically

It's hard to do graphical analysis!

with (x_0, y_0) outside the circle, it'll spin inward $(2x\frac{dx}{dt} + 2y\frac{dy}{dt} < 0)$

eg(ii) Predator-Prey Model

Whale and krill live in the ocean, with former the predator, latter the prey.

• Q: Does the cycle of whales and krill population increasing or decreasing continue indefinitely or one of them die out?

• Modeling: assume the ocean can support unlimited number of krill, so $x' = ax$, with $a > 0$ with-out whales.

Similarly, without krill, whale decreases, $y' = -my$ with $m > 0$.

By considering their interactions, we get

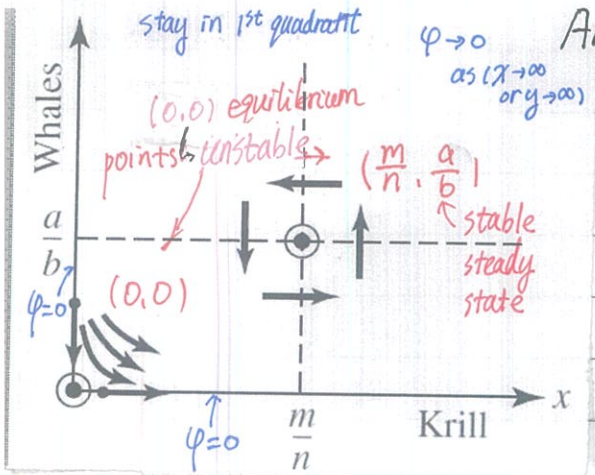
$\frac{dx}{dt} = ax - bxy = x(a - by)$ — decreases in y

$\frac{dy}{dt} = -my + nyx = y(-m + nx)$ — increases in x (correspondence with reality)

25th ANNIVERSARY SICK CHINA

SICK
Sensor Intelligence.

Date: 2021 Oct 1



Analysis: $\frac{dy}{dx} = \frac{(-m+nx)y}{(a-by)x}$

$\Rightarrow \frac{y^a}{e^{by}} = K\frac{e^{nx}}{x^m}$, with $K>0$.

Let $f(y) = \frac{y^a}{e^{by}}$, $g(x) = \frac{x^m}{e^{nx}}$ ($= \varphi(x,y)$)

Equation $f(y)\,g(x) = K$

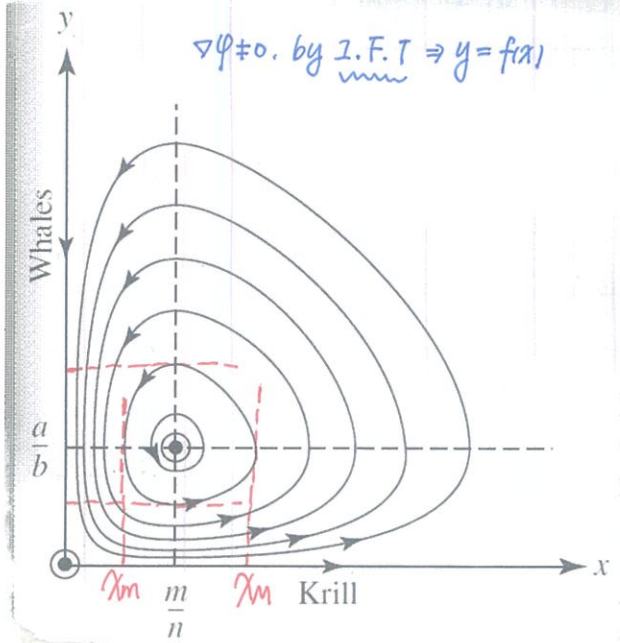
By graphs of f & g.

~~we know they're bdd~~ by M_x, M_y respectively.

$\therefore K \le M_x M_y$. If $K = sM_y$ with $s < M_x$,

The equation $g(x) = s$ has exactly two solutions when $f(y) = M_y$.

↳ indicates that the trajectory is a "circle".



$\nabla\varphi \neq 0$. by I.F.T $\Rightarrow y = f(x)$

i) When $f(y) = M_y$.

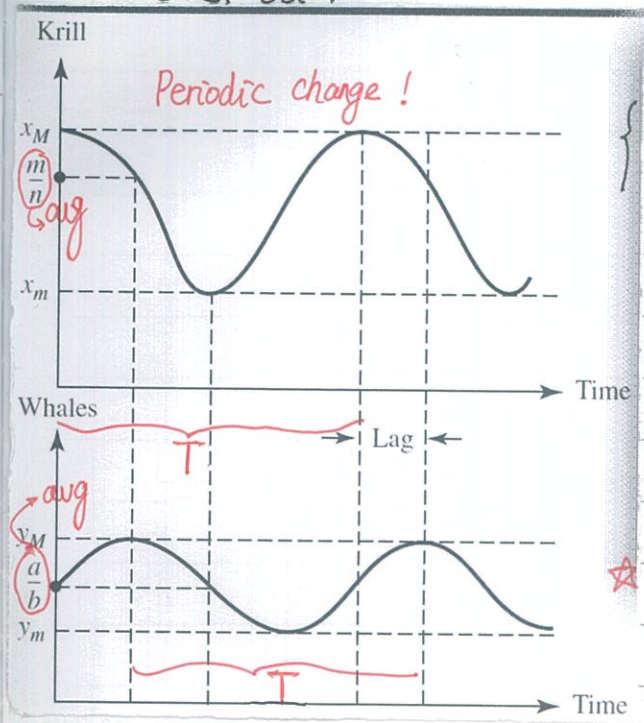
$y = \frac{b}{a}$, we have two side x_m, x_M

2) when $g(x) = M_x$, $x = \frac{m}{n}$.

Similarly.

SICK
Sensor Intelligence.

Date: 2021 Oct 1



Calculate Average:

$\bar{x} = \frac{1}{T}\int_0^T x(t)\,dt$

$\bar{y} = \frac{1}{T}\int_0^T y(t)\,dt$.

We know that $\frac{1}{T}\int_0^T \frac{1}{x}\frac{dx}{dt}\,dt$

$= \frac{1}{T}\int_0^T (a-by)\,dt$ ($\frac{x'}{x} = a-by$)

$LHS = \ln\frac{x(T)}{x(0)} = 0$

$RHS = aT - bT\bar{y}$ (def)

$\bar{y} = \frac{a}{b}$

Similarly, $\bar{x} = \frac{m}{n}$

▲ Modification: (the effect of harvesting)

Harvest krill (make both animals decrease in rate):

$\frac{dx}{dt} = (a-by)x - rx = ((a-r)-by)x$

$\frac{dy}{dt} = (-m+nx)y - ry = (-(m+r)+nx)y$

$\therefore$ The average change to $\bar{y} = \frac{a-r}{b}$, $\bar{x} = \frac{m+r}{n}$ $(r<a)$

Moderate amount of harvesting preys actually increases the average level of preys, while decreasing the avg level of predators. ⇐ Voltera's Principle

Fishing gives the opposite results.

SICK
Sensor Intelligence.

Date: 2021 Oct 1

egs (iii) Two military.

Situation of Combat between two homogeneous forces.

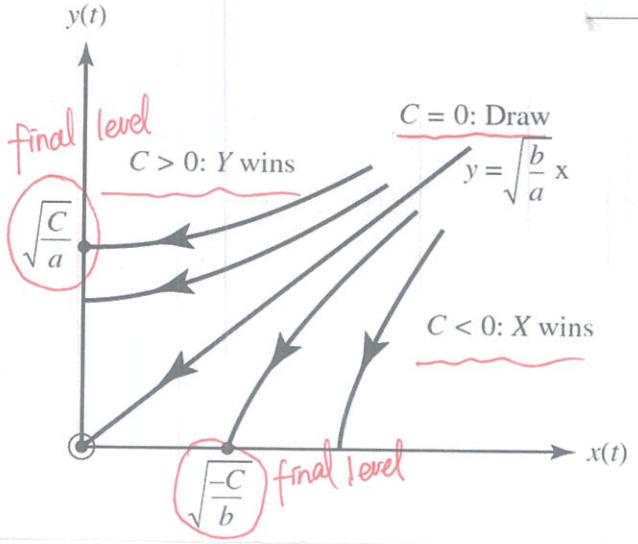
▲ Q: Will one force win / end in draw? How do force levels decrease? How many survivors? How long will it last? How does initial force level affect the outcome?

▲ Modeling: Decreasing rates of X / Y propotional to stength of their opponents. $\begin{cases} \frac{dx}{dt} = -ay, & a>0 \\ \frac{dy}{dt} = -bx, & b>0 \end{cases}$

Analysis: $-\frac{dx}{dt}(+bx) + \frac{dy}{dt}(ay) = 0$.

$\therefore -bx^2 + ay^2 = C$ for some $C \in \mathbb{R}$

Lanchester square law model.



Two cases:

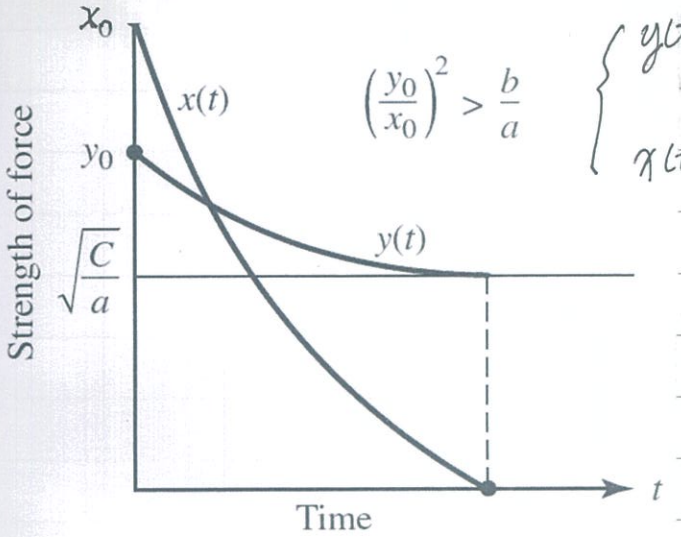
$-bx_0^2 + ay_0^2 > 0$, Y wins.

$-bx_0^2 + ay_0^2 < 0$, X wins.

Date: 2021. Oct. 1

Solve for equations (consider the case Y wins)

$y'' = \frac{d^2y}{dt^2} = aby$ Solving it. ($x'' = abx$, characteristic equation)



$\left(\frac{y_0}{x_0}\right)^2 > \frac{b}{a}$

$\begin{cases} y(t) = y_0 \cosh\sqrt{ab}\,t - x_0\sqrt{\frac{a}{b}}\sinh\sqrt{ab}\,t \\ x(t) = x_0 \cosh\sqrt{ab}\,t - y_0\sqrt{\frac{a}{b}}\sinh\sqrt{ab}\,t \end{cases}$

▲ Modification: Let the Lanchester attrition-rate coefficients depend on # of targets, refine it as $\begin{cases} x' = -gxy \\ y' = -hxy \end{cases}$

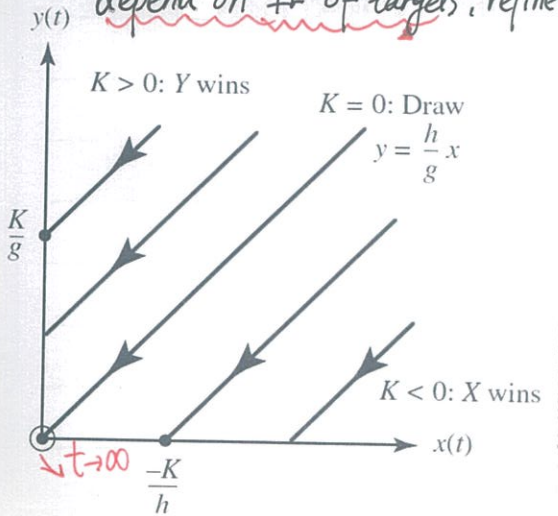
$\therefore \frac{dy}{dx} = \frac{h}{g}$ we have

$gy - hx = K$.

Find x(t) as

solve: $x' = x(-k+hx)$

$x(t) = \begin{cases} x_0\left[\frac{hx_0 - gy_0}{hx_0 - gy_0 e^{-(hx_0 - gy_0)t}}\right], & \text{for } hx_0 \neq gy_0 \\ \frac{x_0}{1 + hx_0 t}, & \text{for } hx_0 = gy_0 \end{cases}$



SICK
Sensor Intelligence.

Date: 2021 Oct. 1

⑤ Numerical Solutions: Euler's Method

(i) Euler's Method:

Consider a system of ODEs $\begin{cases} \frac{dx}{dt} = f(t, x, y) \\ \frac{dy}{dt} = g(t, x, y) \end{cases}$

Step-1 (Take time discretization)

$t_i = t_0 + i\Delta t,\ i \in \mathbb{N}$

Step-2 (Approximation, iteration)

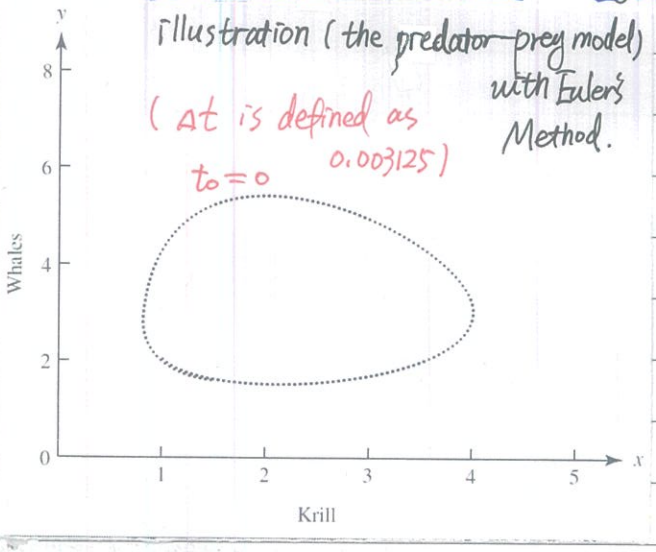
$\frac{x_{i+1} - x_i}{\Delta t} = f(t_i, x_i, y_i)$; $\frac{y_{i+1} - y_i}{\Delta t} = g(t_i, x_i, y_i)$

$$\Rightarrow \begin{cases} x_1 = x_0 + f(t_0, x_0, y_0)\Delta t \\ y_1 = y_0 + g(t_0, x_0, y_0)\Delta t \end{cases} ;\ \cdots ; \begin{cases} x_n = x_{n-1} + f(t_{n-1}, x_{n-1}, y_{n-1})\Delta t \\ y_n = y_{n-1} + g(t_{n-1}, x_{n-1}, y_{n-1})\Delta t \end{cases}$$

Illustration (the predator-prey model) with Euler's Method.

(Δt is defined as 0.003125)

$t_0 = 0$



(ii) Examples.

Date: 2021 Oct. 1

SICK
Sensor Intelligence.

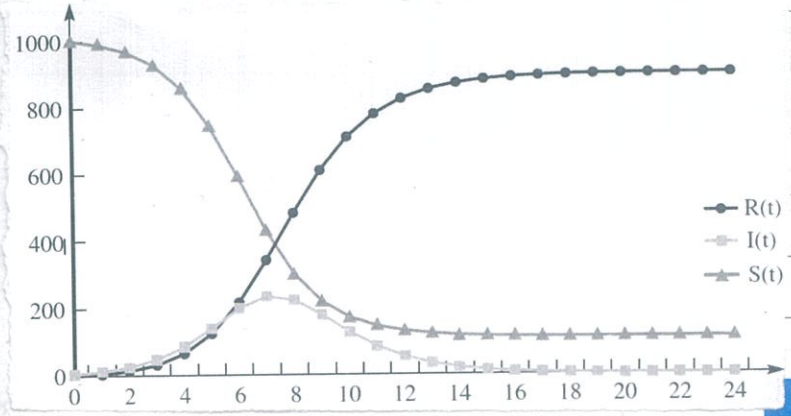
eg: ① Continous SIR Model for epidemics

Consider a disease spreading through out the US. CDC is interested in knowing about & experimenting with a model for it.

▲ Modeling: Divide people into 3 categories.

- every person is either susceptible S / infected I / removed R
- no one leaves / enters, no other contacts with outside community.
- Someone got the flu can never be infected again.
- Time period = per week; Time needed to recover: $\frac{5}{3}$ weeks avg.

$$\begin{cases} \frac{dR}{dt} = 0.6I & \text{per week 0.6 of them (avg) recovered.} \\ \frac{dI}{dt} = -0.6I + 0.001407IS & \text{density-dependent / mass action} \\ \frac{dS}{dt} = -0.001407IS & \text{got by data analyses.} \end{cases}$$



(data in the next page)

SICK
Sensor Intelligence.

Date: 2021. Oct 12

Week	R(t)	I(t)	S(t)
0	0	5	995
1	4.29623	13.0107	982.693
2	13.6301	25.0714	961.299
3	31.503	47.0769	921.42
4	64.6671	84.2088	851.124
5	122.74	138.163	739.097
6	214.755	197.109	588.136
7	339.747	231.908	428.346
8	478.985	221.114	299.902
9	605.748	176.888	217.364
10	704.076	125.809	170.115
11	772.755	83.435	143.81
12	817.839	53.1578	129.003
13	846.397	33.0964	120.506
14	864.118	20.332	115.55
15	874.983	12.3924	112.624
16	881.598	7.51803	110.884
17	885.608	4.54821	109.844
18	888.033	2.74696	109.22
19	889.497	1.6574	108.845
20	890.381	0.99941	108.62
21	890.913	0.60242	108.484
22	891.234	0.36305	108.403
23	891.428	0.21876	108.354
24	891.544	0.13181	108.324

eg② Risk Aversion & Risk Premium

In finance (FIN 2010), we defined utility func & certainty-equivalent (CE). Assume $U(x)$ is the utility func.

$\therefore x_{CE} := U^{-1}(E(U(x)))$ < $E(x)$ (risk-aversion)

$\Rightarrow E(U(x)) < UE(x)$ ($U(x), U'(x) \uparrow$), by Jensen's equality.

$U(x)$ is concave, concavity → extent of risk-aversion

risk-premium $= E(x) - x_{CE}$; (π_A)

relative risk-premium $= \frac{\pi_A}{E(x)} = 1 - \frac{x_{CE}}{E(x)}$; (π_R)

($E(x) \triangleq \bar{x}$, $Var(x) \triangleq \delta_x^2$)

Expand $U(x)$ around $\bar{x}$: $U(x) \approx U(\bar{x}) + \frac{1}{1!}U'(\bar{x})(x-\bar{x}) + \frac{1}{2!}U''(\bar{x})(x-\bar{x})^2$

when $x = x_{CE}$, $U(x_{CE}) \approx U(\bar{x}) + U'(\bar{x})(x_{CE} - \bar{x})$

$E(U(x)) \approx U(\bar{x}) + \frac{1}{2}U''(\bar{x})\delta_x^2$

Date: 2021. Oct.12

SICK
Sensor Intelligence.

Because $E(U(x)) = U(x_{CE})$, we have

$U'(\bar{x})(x_{CE} - \bar{x}) \approx \frac{1}{2}U''(\bar{x})\delta_x^2$

We can approximate $\pi_A = \bar{x} - x_{CE} \approx -\frac{1}{2}\frac{U''(\bar{x})}{U'(\bar{x})}\delta_x^2$

$-\frac{U''(x)}{U'(x)} =: A(x)$ Absolute Risk-aversion

$\therefore \pi_A \approx \frac{1}{2}A(\bar{x})\delta_x^2$;

Similarly, $\pi_R = \frac{\pi_A}{\bar{x}} = -\frac{1}{2}\frac{U''(\bar{x})\cdot\bar{x}}{U'(\bar{x})}\cdot\delta_{\frac{x}{\bar{x}}}^2$

$-\frac{U''(x)\cdot x}{U'(x)} =: R(x)$ Relative Risk-aversion

$\therefore \pi_R = \frac{1}{2}R(\bar{x})\delta_{\frac{x}{\bar{x}}}^2$

▲ Special cases:

Constant Absolute Risk-Aversion (CARA)

$A(x) = -\frac{U''(x)}{U'(x)} = r \Rightarrow U(x) = -ae^{-rx} + b$

Constant Relative Risk-Aversion (CRRA)

$R(x) = -\frac{U''(x)\cdot x}{U'(x)} = r \Rightarrow U(x) = a\frac{x^{1-r}}{r} + b$, $(0<r<1)$

or $a\ln x + b$, $(r=1)$

SICK
Sensor Intelligence.

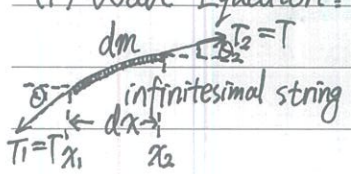
Date: 2021 Oct 13

• PART II Differential Equations (PDE)

① Partial differential equations: often used to model the evolution over time & space.

② Derivation of PDEs.

(i) Wave Equation.



$(y=y(t,x))$

$$F_y = T_2\sin\theta_2 - T_1\sin\theta_1 \approx T\frac{\partial y}{\partial x}(t,x_2) - T\frac{\partial y}{\partial x}(t,x_1) \approx T\frac{\partial^2 y}{\partial x^2}(t,x)(x_2-x_1)$$

($\approx \tan\theta_2$, $\approx \tan\theta_1$; $x_2-x_1 = dx$)

$$\| \quad dm\text{-}a$$

$$dm\cdot y_{tt}(t,x) \qquad (dm=\lambda dx)$$

(λ: linear density)

$$\therefore y_{tt} = \frac{T}{\lambda}y_{xx} \quad \text{(wave equation)}$$

Solution of wave equation: (Characteristics)

Let $c^2=\frac{T}{\lambda}$, $y_{tt}=c^2y_{xx}$.

Let $u=x-ct$, $v=x+ct$, then (changing variables)

$$\begin{cases} \frac{\partial}{\partial t} = \left(\frac{\partial u}{\partial t}\frac{\partial}{\partial u} + \frac{\partial v}{\partial t}\frac{\partial}{\partial v}\right) = -c\frac{\partial}{\partial u} + c\frac{\partial}{\partial v} \\ \frac{\partial}{\partial x} = \left(\frac{\partial u}{\partial x}\frac{\partial}{\partial u} + \frac{\partial v}{\partial x}\frac{\partial}{\partial v}\right) = \frac{\partial}{\partial u} + \frac{\partial}{\partial v} \end{cases}$$

$$\therefore \left(\frac{\partial}{\partial u}+\frac{\partial}{\partial v}\right)^2 y = \frac{1}{c^2}y_{tt} = \frac{1}{c^2}\cdot c^2\left(\frac{\partial}{\partial u}-\frac{\partial}{\partial v}\right)^2 y$$

$$\| \; y_{xx} \quad \Rightarrow \quad y_{uv}=0$$

which means $y(u,v) = f(u)+g(v) = f(x-ct)+g(x+ct)$

$y(x,t)$

Date: 2021 Oct 15

SICK
Sensor Intelligence.

characteristic line: on line $\begin{cases} u=x-ct \\ v=x+ct \end{cases}$ on the u-v plane.

$\frac{\partial^2 y}{\partial u\partial v}=0$.

(ii) One-dim Conservation Law

Space-time variation of quantity density in a very thin tube

☆ $u(t,x)$ – denotes the quantity density at time t, location x

⇒ Total amount of quantity in $[x, x+h]$

$$A\int_x^{x+h} u(t,y)\,dy$$

(A: like volume)

☆ $\Phi(t,x)$ – denotes the flux density at time t, location x

⇒ Net rate the quantity flows into $[x, x+h]$ is

$$A[\Phi(t,x)-\Phi(t,x+h)]$$

☆ $f(t,x,\Phi(t,x))$ – denotes the source func (e.g. chemical reactions)

⇒ Rate of quantity produced in $[x, x+h]$ by sources.

$$A\int_x^{x+h} f(t,y,u(t,y))\,dy$$

The conservation law: rate of change amount = rate flows in + rate produced

$$\Rightarrow \frac{\partial}{\partial t}A\int_x^{x+h} u(t,y)\,dy = A[\Phi(t,x)-\Phi(t,x+h)] + A\int_x^{x+h} f(t,y,u(t,y))\,dy$$

Divide both sides by h. take $h\to 0$, we have

$$u_t(t,x) = -\partial_x\Phi(t,x) + f(t,x,u(t,x))$$

(i.e. $-\Phi_x(t,x)$)

SICK
Sensor Intelligence.

Date: 2021 Oct 15

(iii) Transport Equations

(suppose) The quantity of interest moves together with the surrounding medium with a given velocity, $v(t,x)$. & $f=0$ (no source)

Then $\Phi(t,x) = u(t,x)v(t,x)$ (defn of Φ)

$\partial_x \Phi(t,x) = u_x(t,x)\cdot v(t,x) + v_x(t,x)u(t,x)$

Use (ii) (a special case of 1-dim conservation Law)

$u_t(t,x) = -v u_x(t,x)$

(iv) Diffusion / Heat Equations

(suppose) No sources, the 1-dim conservation Law: $u_t + \Phi_x = 0$

Because heat moves from high temperature to low one (⇕ concentration)

we have $\Phi(t,x) = -D u_x(t,x)$

$\therefore$ 1-dim diffusion equation $u_t = D u_{xx}$

(v) Initial & Boundary Conditions

PDEs need the initial & boundary conditions to be unique.

☆ Initial conditions $u(0,x) = \psi(x)$ form

☆ Boundary conditions.

i) Dirichlet $u(t,x) = g(t,x)$, $\forall x \in \partial\Omega$

SICK
Sensor Intelligence.

Date: 2021 Oct 20

(e.g. heat equation, submerging in large container with melting ice.)

ii) Neumann $\partial_x u(t,x) = g(t,x)$, $\forall x \in \partial\Omega$

(e.g. heat equation, head is confined in the body.)

iii) Robin $\partial_x u(t,x) + a u(t,x) = g(t,x)$, $\forall x \in \partial\Omega$

(e.g. heat equation, body immersed in a reservoir with known temperature)

③ Modeling Examples.

Eg 1. (Traffic Flow Modeling)

Different scales: (microscopic) a specific car & driver's respond

(macroscopic) approx. as gas / liquid (continuum)

▲ Modeling.

i) Continuum limit: suppose at every position x, $\exists$ a car with velocity $v(t,x)$ at time t. $\begin{cases} \frac{dx}{dt} = v(t, x(t)), \\ x(t_0) = x_0 \end{cases}$ (each car)

ii) Traffic density $\rho(t,x)$: avg # vehicles per unit length of road.

Suppose length of car $= L$ & equal distance d.

$\rho(t,x) = \frac{1}{L+d} \le \frac{1}{L}$ $(d \ge 0)$

Traffic flux $q(t,x) = \rho(t,x)v(t,x)$

iii) Fix an arbitrary interval $[A, B]$

By the conservation law $\frac{d}{dt}\int_A^B \rho(t,x)dx = q(t,A) - q(t,B)$

Take $B \to A = x$, we have

SICK Sensor Intelligence.

Date: 2021 Oct 20

$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0$, with $q = \rho v$, we have

$\frac{\partial \rho}{\partial t} + \frac{\partial (v\rho)}{\partial x} = 0$

(Remark: $\dot{x}_1(t)$ $\dot{x}_2(t)$ let $x_1(t)$, $x_2(t)$ be positions of two cars (without any other cars / with some cars)

Then $\frac{d}{dt}\int_{x_1(t)}^{x_2(t)} \rho(t,x)dx = x_2'(t)\rho(t,x_2(t)) - x_1'(t)\rho(t,x_1(t)) + \int_{x_1(t)}^{x_2(t)} \frac{\partial \rho}{\partial t} dx$

no cars move in/out of the interval

$0 = \int_{x_1(t)}^{x_2(t)} \frac{\partial (v\rho)}{\partial x} dx$ ⇐)

$v(t,x)$ can be specified as $v(\rho(t,x))$ (reasonable with reality)

& Let $F(\rho) = \rho v(\rho)$, we have $\frac{\partial \rho}{\partial t} + F'(\rho)\frac{\partial \rho}{\partial x} = 0$

⇒ Applications: one of the possible v-ρ relations

$v(\rho) = v_{max}(1 - \rho/\rho_{max})$

$\Rightarrow F'(\rho) = v_{max}(1 - 2\rho/\rho_{max}) \approx v_{max}(1 - 2\rho_0/\rho_{max}) := v_0$

($\rho \approx \rho_0$ which is a small deviation from a constant density)

Then, above model of PDE — $\frac{\partial \rho}{\partial t} + v_0 \frac{\partial \rho}{\partial x} = 0$

Use method of characteristics $\rho(t,x) = f(x - v_0 t)$

with v_0 — speed of linear traffic wave.

(solve for ρ: parametrize line (characteristic) by $(x(s), t(s))$

Consider $\frac{d\rho}{ds} = \frac{dx}{ds}\frac{\partial \rho}{\partial x} + \frac{dt}{ds}\frac{\partial \rho}{\partial t}$

when $\frac{dx}{ds} = v_0$, $\frac{dt}{ds} = 1$, $\frac{d\rho}{ds} = 0$.

which means if $\begin{cases} x = x_0 + v_0 s \\ t = t_0 + s \end{cases}$, $\rho(t,x) = \rho(t(s), x(s)) = \rho(t(0), x(0))$

choose different parametrize, (x_0, t_0) / $(x_0 - v_0 t_0, 0)$ we can get all pts (x,t)

$= \rho(0, x - v_0 t) \triangleq f(x - v_0 t)$)

Date: 2021 Oct 20

SICK Sensor Intelligence.

Eg 2. (Population Growth Modeling)

Previously, we talk about population growth model without age structure. (e.g. $N'(t) = rN(t)(M - N(t))$ logistic)

▲ Modeling: population growth with age structure

(parametrize $t(s) = t$ (time); $a(s) = a$ (age structure))

$\Rightarrow n(t,a) = n(t(s), a(s)) = n(s)$.

i) Total population at time t: $n(t) = \int_0^\infty n(t,a)da$

Let $\mu(a)$ be the death rate at age a

collected by data analyses

$dn(t,a) = \frac{\partial n}{\partial t}dt + \frac{\partial n}{\partial a}da = -\mu(a)n(t,a)dt$ ↑ real-world regulation

Because $\frac{da}{dt} = 1$ (defn), $\Rightarrow$ $\frac{\partial n}{\partial t} + \frac{\partial n}{\partial a} = -\mu(a)n(t,a)$

Von Foerster Equation

ii) Initial distribution age structure: $n(0,a) = f(a)$, $a \geq 0$

iii) Let $b(a)$ be the birth rate at age a (age a people give birth to age 0)

then, $n(t,0) = \int_0^\infty b(a)n(t,a)da$

($b(a) \to 0$ when $a \to 0/\infty$)

Solve for the model:

By parametrize above, we want

$-\mu(a(s))n(s) = n'(s) = \frac{\partial n}{\partial t}t'(s) + \frac{\partial n}{\partial a}a'(s)$

want — set to 1

25th ANNIVERSARY SICK CHINA

SICK
Sensor Intelligence.

Date: 2021 Oct 21

We then have $n(s) = n_0 \exp\left(-\int_{a_0}^{a_0+s} \mu(\delta)\,d\delta\right)$

(with $t = t_0 + s$, $a = a_0 + s$)

For $a > t$, set $t_0 = 0$, then $n(t,a) = n(0,a_0)\exp\left(-\int_{a_0}^{t+a_0} \mu(\delta)\,d\delta\right)$, $a > t$

In this case, $n_0 = n(0,a_0)$

$$n(t,a) = f(a-t)\exp\left(-\int_{a-t}^{a} \mu(\delta)\,d\delta\right), \quad a > 0$$

For $a < t$, set $a_0 = 0$. $\Rightarrow a = t - t_0 = s$

$$n(t,a) = n(t_0,a_0)\exp\left(-\int_0^a \mu(\delta)\,d\delta\right) = n(t-a,0)\exp\left(-\int_0^a \mu(s)\,ds\right)$$

$$n(t,0) = \int_0^\infty b(a)\,n(t,a)\,da$$

$$= \int_0^t b(a)\,n(t,a)\,da + \int_t^\infty b(a)\,n(t,a)\,da = \int_0^t b(a)\,n(t-a,0)\exp\left(-\int_0^a \mu(s)\,ds\right)da + \int_t^\infty b(a) f(t-a)\exp\left(-\int_{a-t}^{a} \mu(s)\,ds\right)da$$

2 terms

$\Downarrow$ boundary conditions

▲ Long-term asympotic solutions

On the long run, the initial age structure is not important.

Seek for the solution in form $n(t,a) = e^{rt} r(a)$

(satisfying $\frac{n(t,a)}{\int_0^\infty n(t,a)\,da} = \frac{r(a)}{\int_0^\infty r(a)\,da}$)

Then $\frac{\partial n}{\partial t} + \frac{\partial n}{\partial a} = re^{rt} r(a) + e^{rt} r'(a) = -\mu(a) e^{rt} r(a)$

$\Rightarrow r'(a) = -(\mu(a) + r)\,r(a)$

$\therefore r(a) = r(0)\exp\left(-ra - \int_0^a \mu(s)\,ds\right)$

SICK
Sensor Intelligence.

Date: 2021 Oct 21

By the boundary condition, $n(t,0) = e^{rt} r(0) = \int_0^\infty b(a) e^{rt} r(0) \exp\left(-ra - \int_0^a \mu(s)\,ds\right)da$

r determined by $1 = \int_0^\infty b(a)\exp\left(-ra - \int_0^a \mu(s)\,ds\right)da$.

Eg 3. (Financial Derivatives Pricing — Black-Scholes Model)

Option: getting the right but no obligation of buying a stock at "\$100" per year (S_0)

- Aim: give a fair option price (do option pricing)
- Pricing by replication:

Consider the over-simplified case: the value of stock traded at \$100 (S_0), and can become uS_0, dS_0 ($u \neq 1 \neq d$) with probability p, $1-p$. The risk-free asset have the rate of return r. (No friction in trading)

☆ IDEA: If there is a portfolio perfectly replicating the option payoff, its value = the option price.

(Otherwise, arbitrage — e.g. if option price > portfolio value, $\Rightarrow$ sell the option & buy the portfolio)

Δ shares & b risk-free assets:

$$\begin{cases} b(1+r) + \Delta u S_0 = V_1^u = uS_0 - S_0 & \leftarrow \text{option payoff case 1} \\ b(1+r) + \Delta d S_0 = V_1^d = 0 & \leftarrow \text{option payoff case 2} \end{cases}$$

$$\therefore V_0 = b + \Delta S_0 = \frac{1}{1+r}\left(\tilde{p} V_1^u + (1-\tilde{p}) V_1^d\right)$$

$$= \frac{1}{1+r} \tilde{E}^Q(V_1) \quad \text{with } \tilde{p} = \frac{1+r-d}{u-d}$$

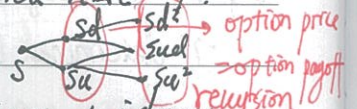
$d < 1+r < u$

SICK Sensor Intelligence.

Date: 2021 Oct 25

From one period, we consider multi-period $n\Delta t = T$. → maturity

with one period interest rate $r\Delta t$.



After k step, $S_k(w) = S_0 d^{k-j} u^j$ with j ups, $k-j$ downs

$d = 1+r\Delta t - \delta\sqrt{\Delta t} < 1+r\Delta t < 1+r\Delta t+\delta\sqrt{\Delta t} = u$, with $\tilde{p} = 1/2$

▲ Modeling:

Suppose the option price is $C(t,S) = C$, where $0 \le S < \infty$

$t \le T$, C is smooth.

Then $C^+ := C(t_0+\Delta t, uS_0) = C(t_0+\Delta t, S_0+(u-1)S_0)$ (like V_1^u)

$C^- := C(t_0+\Delta t, dS_0) = C(t_0+\Delta t, S_0+(d-1)S_0)$ (V_1^d)

By Taylor's theorem, $C^+ = C(t_0,S_0) + \frac{\partial C}{\partial S}(t_0,S_0)(\delta\sqrt{\Delta t}+r\Delta t)S_0$

$+\frac{\partial C}{\partial t}(t_0,S_0)\Delta t + \frac{1}{2}\frac{\partial^2 C}{\partial S^2}(t_0,S_0)(\delta\sqrt{\Delta t}+r\Delta t)^2 S_0^2 + O(\sqrt{\Delta t}^3)$

$C^- = C(t_0,S_0) + \frac{\partial C}{\partial S}(t_0,S_0)(-\delta\sqrt{\Delta t}+r\Delta t)S_0 + \frac{\partial C}{\partial t}(t_0,S_0)\Delta t + \frac{1}{2}\frac{\partial^2 C}{\partial S^2}$

$(-\delta\sqrt{\Delta t}+r\Delta t)^2 S_0^2 + O(\sqrt{\Delta t}^3)$

According to the previous conclusion, we have

$(1+r\Delta t)C(t_0,S_0) = \frac{1}{2}(C^+ + C^-) + O(\Delta t^2)$

↓ recursion (sell option at price)

Up to $O(\sqrt{\Delta t}^3)$, $\Delta t \to 0$,

we get $(1+r\Delta t)C(t_0,S_0) = C(t_0,S_0) + \frac{\partial C}{\partial S}(t_0,S_0)r\Delta t S_0 + \frac{\partial C}{\partial t}(t_0,S_0)\Delta t$

$+\frac{1}{2}\frac{\partial^2 C}{\partial S^2}(t_0,S_0)\cdot S_0^2\cdot\delta^2\Delta t$

$\Rightarrow rC(t_0,S_0) = \frac{\partial C}{\partial S}(t_0,S_0)rS_0 + \frac{\partial C}{\partial t}(t_0,S_0) + \frac{1}{2}\frac{\partial^2 C}{\partial S^2}(t_0,S_0)\delta^2 S_0^2$

Suppose at maturity the stock price is K.

we have initial condition $C(T,S) = (S-K)^+$

SICK Sensor Intelligence.

Date: 2021 Oct 25

(if $S \le K$, get 0, otherwise $S-K$)

$\Rightarrow$ Black-Scholes PDE

☆
$$\begin{cases} C_t + rSC_S + \frac{1}{2}\delta^2 S^2 C_{SS} - rC = 0 \\ C(T,S) = (S-K)^+ \end{cases}$$

Solve it by changing variables $C(t,S) = e^{-r(T-t)}G(S,X)$

$x = \ln S + (r - \delta^2/2)(T-t)$, $s = \delta^2(T-t)/2$

$T-t = 2s/\delta^2$, $S = e^{x-(2r/\delta^2-1)s}$.

$$\Rightarrow \begin{cases} G_s = G_{xx}, \ x \in \mathbb{R}, \ s > 0 \\ G(0,x) = (e^x - K)^+ \end{cases}$$ (Heat Equation)

By Fourier Transform (need more explanations) ($\hat{G}(s,\xi)$)

☆ $\mathcal{F}(\frac{\partial G}{\partial s}) = \frac{\partial \mathcal{F}(G)}{\partial s} = \tilde{G}_s$

$\mathcal{F}(\frac{\partial^2 G}{\partial x^2}) = (i\xi)^2\mathcal{F}(G) = -\xi^2\tilde{G}$ ($\varphi(x) = G(0,x)$)

$\Rightarrow \partial_s\tilde{G} + \xi^2\tilde{G} = 0$. $\tilde{G} = \tilde{\varphi}(\xi)e^{-\xi^2 s}$ (let $\tilde{G}(0,x) = \tilde{\varphi}(\xi)$)

$\therefore G = \mathcal{F}^{-1}(\tilde{G}) = \int_{-\infty}^{\infty}(e^y - K)^+ \frac{1}{2\sqrt{\pi s}} e^{-\frac{(x-y)^2}{4s}} dy$

$= e^{x+s}\mathcal{N}(\frac{x-\ln K+2s}{\sqrt{2s}}) - K\mathcal{N}(\frac{x-\ln K}{\sqrt{2s}})$

where $\mathcal{N}(\cdot) = \int_{-\infty}^{\cdot}\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}dx$.

$$\Rightarrow \boxed{\begin{aligned} C(t,S) &= S\mathcal{N}(d_+) - e^{-r(T-t)}\mathcal{N}(d_-) \\ d_\pm &= \frac{\ln(S/K) + (r \pm \delta^2/2)(T-t)}{\delta\sqrt{T-t}} \end{aligned}}$$

25th ANNIVERSARY SICK CHINA

SICK
Sensor Intelligence.

Date: 2021 Oct 29

• PART III Optimization – Linear Programming

① Theoretical knowledge:

Defn; Geometric solutions; the simplex method; sensitivity analysis (in MAT 3007 course)

② Modeling Examples.

e.g. 1) Foreign-Currency Trading

▲ Assumptions. Big FX market with over $1 trillion daily trading

One day's currency rate as follows

From: \ To:	US Dollar	Pound	FFranc	D-Mark	Yen
US Dollar		0.6390	5.3712	1.5712	98.8901
Pound	1.5648		8.4304	2.4590	154.7733
FFranc	0.1856	0.1186		0.2921	18.4122
D-Mark	0.6361	0.4063	3.4233		62.9400
Yen	0.01011	0.00645	0.05431	0.01588	

▲ Aim = No arbitrage! (currency transaction creates profit without initiating funds called arbitrage)

▲ Modeling: objective – maximize final amount of one currency

constrains – nonnegative final net amount (no investment)

$i = 1, 2, \cdots, 5$ represent the currencies US dollar, British pound, French franc, German D-mark & Japanese Yen, respectively. x_{ij} – amount of currency from i to j $(i \neq j)$

SICK
Sensor Intelligence.

Date: 2021 Oct 29.

Let f_k be final net amount of currency k.

we have $f_1 = 1.5648x_{21} + 0.1856x_{31} + 0.6361x_{41} + 0.01011x_{51} - \sum_{i \neq 1} x_{1i}$

$f_2 = 0.6390x_{12} + 0.1186x_{32} + 0.4063x_{42} + 0.00645x_{52} - \sum_{i \neq 2} x_{2i}$

f_3, f_4, f_5 can be written in the same way.

The Model: $\max_x f_1$ subject to $f_i, x_{ij} \geq 0, \forall i \neq j, i, j = 1, \cdots 5$

(LP model)

$f_1 \leq 1$ → once arbitrage emerges $f_1 \to \infty$ (in case)

$\Rightarrow \max f_1 \leq 1$ normalize

e.g. 2) Option Price in Incomplete Market

▲ Assumptions: n states, m assets, with $m \times n$ payoff matrix $X = (x_{ij})$ means i asset at j state (price)

The asset pricing vector $p = (p_i) \in \mathbb{R}^m$

▲ Modeling: If there is an abritrage opportunity,

$\exists$ amount vector $a \in \mathbb{R}^m$ s.t. $a^T X > 0$ & $a^T p = 0$

or $a^T X = 0$ & $a^T p < 0$

$\Leftrightarrow$ an optimization problem as

$\left(\begin{array}{l} \min a^T p \\ \text{subject to } a^T X \geq 0 \end{array} \right) < 0$

SICK Sensor Intelligence.

Date: 2021 Oct 29

By alternating theorem (Farka's Lemma)
It is equivalent to another expression.
(The First Fundamental Theorem of Asset Pricing - FTAP)
The market admits NO arbitrage exactly when $\exists$ a positive vector $b \in \mathbb{R}^n$ s.t. $p = Xb$
(b works as a probability vector)

▲ Application:

(i) Binomial Model: in Black-Schole PDE part. we can use FTAP to get $E^Q(V) \cdot \frac{1}{1+r}$, by $b = \begin{bmatrix} \frac{1+r-d}{u-d} \\ \frac{u-(1+r)}{u-d} \end{bmatrix}$.

(ii) Incomplete Markets:
Consider the case of n possible values, $S_1 = u_1 S_0, u_2 S_0 \cdots u_n S_0$
No arbitrage price is not unique (underdetermined system)

$V_0 = [\underline{V_0}, \overline{V_0}] = \left[\begin{matrix} \min V_1^T b \\ \text{s.t. } S_1^T b = S_0, \\ \mathbb{1}^T b = 1+r \\ b > 0 \end{matrix} \quad \begin{matrix} \max V_1^T b \\ \text{s.t. } S_1^T b = S_0 \\ \mathbb{1}^T b = 1+r \\ b > 0 \end{matrix} \right]$

Dual problem of $\overline{V_0}$ — super replication ($\min y^T p$ subject to $X^T y \geq V_1$)
Dual problem of $\underline{V_0}$ — sub-replication ($-\max (-y)^T p$ subject to $X^T(-y) \geq V_1$)

e.g. 3) Estimating & Managing Risk

▲ Estimating Value-at-Risk (VaR)
Denote random variable L as the loss of a portfolio

Date: 2021 Oct 30

in the future. The VaR is defined as

$$VaR_\alpha = \sup\{x \mid F_L(x) \leq \alpha\} = F_L^{-1}(\alpha)$$

where $F_L(x)$ can be seen as the cumulative distribution func. (CDF)

$$F_L(x) = P(L \leq x)$$

Estimate VaR (Monte Carlo Simulation)

(i) Generate n i.i.d. (independent, identically distributed) samples of L, sorted as $L_1 \leq L_2 \leq \cdots \leq L_n$

(ii) Approx. F as

$$\star\ F_L^{(n)}(x) = \begin{cases} 0, & x \leq L_1 \\ \frac{i-1}{n-1} + \frac{1}{n-1} \cdot \frac{x - L_i}{L_{i+1} - L_i}, & L_i \leq x \leq L_{i+1},\ i = 1, \cdots, n-1 \\ 1, & x \geq L_n \end{cases}$$

(piece-wise linear)

(iii) If $\alpha \in [\frac{i-1}{n-1}, \frac{i}{n-1}]$, $VaR_\alpha^{(n)}(L) = L_i + (L_{i+1} - L_i)(\alpha(n-1) - (i-1))$

▲ Conditional Value-at-Risk (CVaR)

Defined as $CVaR_\alpha(L) = \mathbb{E}(L \mid L > VaR_\alpha) = \frac{\mathbb{E}(L, L > VaR_\alpha)}{P(L > VaR_\alpha)}$

$= \mathbb{E}(1, L > VaR_\alpha)$

Estimate CVaR

(i) estimate VaR as above shows

(ii) Approx. CVaR as $CVaR_\alpha^{(n)} = \frac{\frac{1}{n}\sum_{i=1}^n L_i \mathbb{1}_{\{L_i > VaR_\alpha^{(n)}\}}}{\frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{L_i > VaR_\alpha^{(n)}\}}}$ ($\mathbb{E}$ use n samples to estimate; $\mathbb{1}$ indicator function)

(By Law of Large Numbers)

▲ A numerical example:

$F_L = 1 - e^{-x}$ (suppose) $CVaR_\alpha = \frac{1}{1-\alpha}\int_{VaR_\alpha}^{\infty} x e^{-x} dx \overset{\alpha = 0.95}{=\!=} 3.996$

25th ANNIVERSARY SICK CHINA

SICK
Sensor Intelligence.

Date: 2021 Oct 30

The result of estimation:

n	$\text{VaR}_\alpha^{(n)}$	$\text{CVaR}_\alpha^{(n)}$
100	2.880	3.766
1,000	2.994	3.912
10,000	3.003	4.016

▲ Stochastic Optimization:

$\min_x \mathbb{E}(f(x,\xi))$
subject to $x \in X$

can be approximize by (Monte Carlo Simulation)

⇒ Take a sample, $\xi_1, \xi_2, \cdots \xi_N$.

$\min_x \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$
subject to $x \in X$

▲ Modeling:

Investment problem (simple)

$\min_{x_i} CVaR_\alpha(-\sum_{i=1}^n x_i R_i)$ → loss ⇒ minimize loss risk
subject to $\mathbb{E}(\sum_{i=1}^n x_i R_i) \geq r_0$ → promise certain profit
$\sum_{i=1}^n x_i = 1$

n assets $x_1, \cdots x_n$ with random rate of return R_i, $i=1,\cdots,n$.

An alternative formulation of CVaR

$$CVaR_\alpha(L) = \min_{\xi \in \mathbb{R}} \left[\xi + \frac{1}{1-\alpha}\mathbb{E}[\max\{L-\xi, 0\}] \right]$$

Reason: $g(\xi) = \xi + \frac{1}{1-\alpha}\mathbb{E}((L-\xi)^+) = \xi + \frac{1}{1-\alpha}\int_\xi^\infty (x-\xi) f_L(x)dx$

$\Rightarrow g'(\xi) = \frac{1}{1-\alpha}(F_L(\xi) - \alpha)$. only candidate ξ is $\xi = VaR_\alpha$

SICK
Sensor Intelligence.

Date: 2021 Oct 30

$\therefore CVaR_\alpha(L) = \min_{\xi \in \mathbb{R}} g(\xi)$

We can write the initial problem as

$\min_{x_i, \xi} \xi + \frac{1}{1-\alpha}\mathbb{E}[\max\{-\sum_{i=1}^n x_i R_i - \xi, 0\}]$
subject to $\mathbb{E}(\sum_{i=1}^n x_i R_i) \geq r_0$
$\sum_{i=1}^n x_i = 1$.

Use simulation to estimate R_i with realizations r_{ij} & probabilities $\frac{1}{S}$

$$\mathbb{E}(\sum_{i=1}^n x_i R_i) = \sum_{i=1}^n x_i \mathbb{E}(R_i) = \sum_{i=1}^n x_i \frac{1}{S}\sum_{j=1}^S r_{ij} \triangleq \sum_{i=1}^n x_i \bar{R}_i$$

$$\mathbb{E}[\max\{-\sum_{i=1}^n x_i R_i - \xi, 0\}] = \mathbb{E}[\max\{\sum_{i=1}^n \sum_{j=1}^S x_i r_{ij} - \xi, 0\}]$$

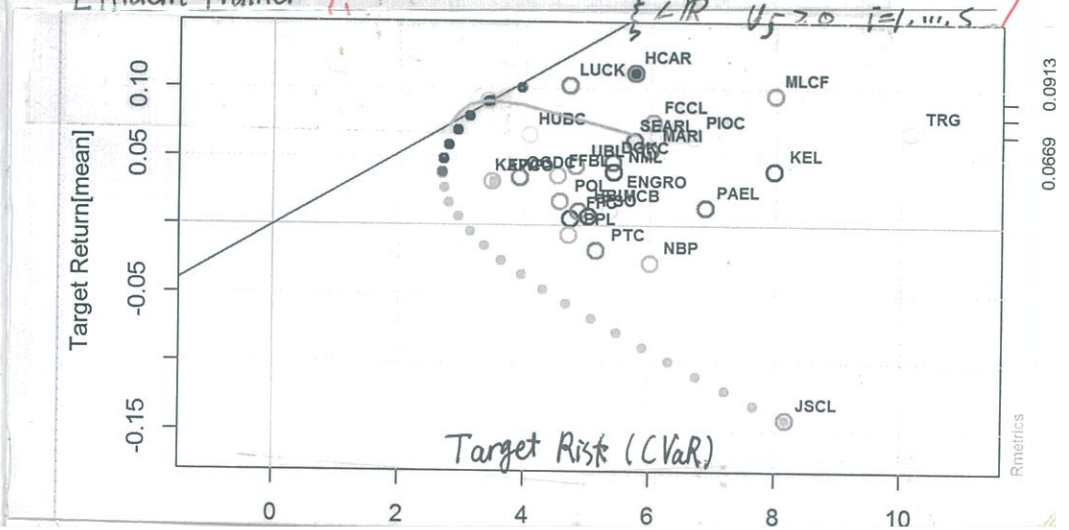
$$= \mathbb{E}[\max\{-\sum_{j=1}^S \sum_{i=1}^n x_i r_{ij}, 0\}] = \frac{1}{S}\sum_{j=1}^S u_j$$

denoted as u_j, $\sum_{j=1}^S u_j$

⇒ the estimation LP:

$\min_{\xi, u_j, x_i} \xi + \frac{1}{(1-\alpha)S}\sum_{j=1}^S u_j$
subject to $u_j \geq -\sum_{i=1}^n x_i r_{ij} - \xi$, $j=1,\cdots,S$
$\sum_{i=1}^n x_i \bar{R}_i \geq r_0$, $\sum_{i=1}^n x_i = 1$
$\xi \in \mathbb{R}$ $u_j \geq 0$ $i=1,\cdots,S$

Efficient Frontier



SICK
Sensor Intelligence.

Date: 2021 Nov 6

• PART IV – Optimization – Non-linear Programming

① Theoretical (part 1)

(i) Existence of solutions; solving NLP by software (in MAT3007)

(ii) ★Examples using NLP modeling

e.g. 1 (Regression)

Given a sequence of data points $\{(x_1, y_1), \dots (x_n, y_n)\}$ with $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$. want to find a mapping $f(x;\theta)$ with θ as the parameter. (e.g. $f(x) = a + bx$ with $\theta = (a, b)$)

formulated as $\min_\theta \frac{1}{n} \sum_{i=1}^n |y_i - f(x_i;\theta)|^r$ (or L_r norm $\|y - f(x;\theta)\|_r$)

e.g.2 (Likelihood maximization Estimation)

Suppose $\exists$ a sequence of independent random observations $\{x_1, x_2, \dots x_n\}$ of a RV X. Want to infer a parametric distribution $f(x;\theta)$ of X from the observations. (e.g. Gaussian $f(x;\theta) = \frac{1}{\sqrt{(2\pi)^n \det\Sigma}} \exp\{-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}\}$, where Σ is the covarance matrix)

formulated as $\max_\theta \prod_{i=1}^n f(x_i;\theta)$ (Because $P(X = x_1 \text{ or } x_2 \text{ or} \dots x_n) \sim$ (density) $\prod_{i=1}^n f(x_i;\theta)$)

e.g.3 (Portfolio Optimization)

An investor has n available assets where she allocate her money. RV R_i represents the (random) return of asset i

Date: 2021 Nov 6

Naive formulation $\max_x \sum_{i=1}^n R_i x_i$, s.t. $\sum_{i=1}^n x_i = 1$

(NOT well-defined)

Further improvement $\max_x \sum_{i=1}^n \mu_i x_i$, s.t. $\sum_{i=1}^n x_i = 1$

where $\mu_i = \mathbb{E}[R_i]$ (NOT well-posed ⇒ only all μ_i equals not unbdd)

☆Mean-Variance Framework (Harry Markowitz)

$$\max_x \mu^T x - \frac{1}{2}\gamma x^T \Sigma x \quad \text{s.t. } e^T x = 1$$

where $\mu = \mathbb{E}[R]$, $\Sigma = \text{Var}[R]$ (covariance matrix) and γ – risk-aversion coeffic (penalty)

(*)

($x^T \Sigma x = \text{Var}[x^T R]$ ($\text{Var}(\sum_{i=1}^n R_i x_i) = x^T \text{Var}[R] x$) the property of Var

Alternative Formulations

(same Lagrangian) $\max_x \mu^T x$, s.t. $x^T \Sigma x \le \sigma^2$, $e^T x = 1$

$\min_x x^T \Sigma x$, s.t. $\mu^T x \ge z$, $e^T x = 1$ ("efficient frontier" (Given Return ⇒ minimum risk))

(Estimations of μ, Σ: factor analysis; Black-Litterman shrinked estimation, ...)

By KKT conditions, we can find analytical solution for (*)

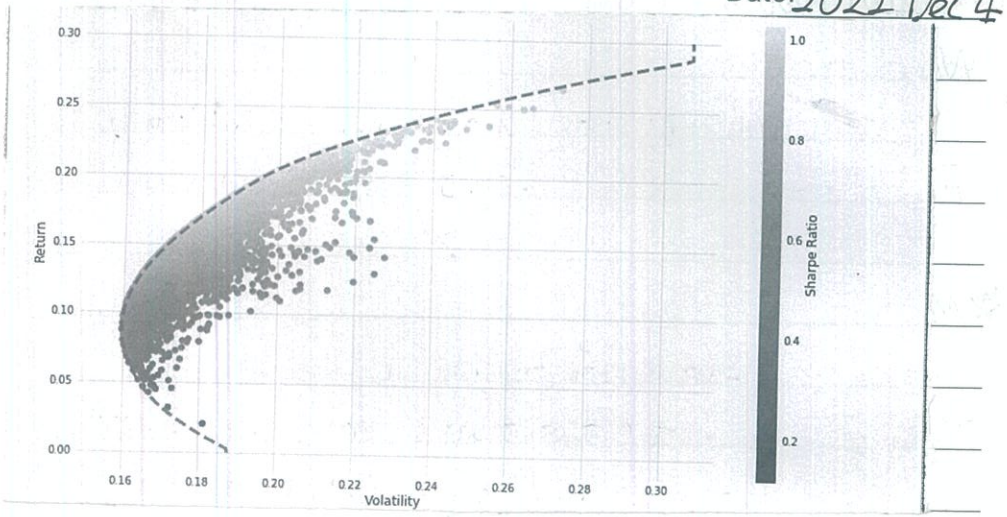
$x = \frac{1}{\gamma}\Sigma^{-1}\mu - \frac{\lambda_\gamma}{\gamma}\Sigma^{-1}e$, where $\lambda_\gamma = \frac{\mu^T\Sigma^{-1}e - \gamma}{e^T\Sigma^{-1}e}$

The corresponding efficient frontier.

(next page)

SICK
Sensor Intelligence.

Date: 2021 Dec 4



② Theoretical (Part II)

(i) Optimality conditions, convex optimizations, KKT conditions (in MAT3007 Notes)

(ii) Non-differentiable Convex Optimization & sub-gradient

The subgradient of f, denoted as ∂f, is given by

$$\partial f(x) := \{g_x \mid f(y) \geq f(x) + g_x^T(y-x)\}, \forall y$$

Given f is convex, x^* is a global minimizer iff $0 \in \partial f(x^*)$.

e.g. $f = \|x\|_1$ (Euclidean norm), $\partial f(0) = \{x \mid \|x\|_\infty \leq 1\}$

(However, f is not differentiable at $x=0$)

★ Application: LASSO (l_1 regularized linear regression)

For the model $y = X\beta$, LASSO estimator is defined as

$$\min_\beta \frac{1}{2}\|X\beta - y\|_2^2 + \rho\|\beta\|_1$$

(Generalized LASSO $\frac{1}{2}\|X\beta - y\|_2^2 + \rho\|F\beta\|_1$, where F is

SICK
Sensor Intelligence.

Date: 2021 Dec 4

an arbitrary linear transformation.)

(sol.) ADMM form: $\begin{cases} \beta^{k+1} := (X^TX + \lambda I)^{-1}(X^Ty + \lambda(\beta^k - u^k)) \\ z^{k+1} := S_{\rho/\lambda}(\beta^{k+1} + u^k) \\ u^{k+1} := u^k + \beta^{k+1} - z^{k+1} \end{cases}$

(Add constrain: $\beta = z / \beta - z = 0$)

→ ridge regression

(iii) Traveling Salesman Problem (TSP) & Simulated Annealing (SA)

TSP: find the shortest tour towards set of given locations.

s.t. 1° each location is passed once; 2° end at the starting point.

★ Simulated Annealing Algorithm (famous together with NN, GA)

1° Choose a random state s, initial Temperature T (large) & the cooling rate β. ($\beta \in (0,1)$)

2° Create a new state s'.

3° Compute the cost function of 2 states & Compare:

If $\begin{cases} C(s') < C(s) \ (\delta = \frac{C(s')-C(s)}{C(s)} < 0) \text{ Accept } s' \\ \text{Otherwise } (\delta > 0) \text{ Accept } s' \text{ with probability } e^{-\delta/T} \end{cases}$

& cool T as βT ($T = \beta T$)

4° Stopping criterion:

small temperature / reject enough times new s'

Traits of SA: $\begin{cases} \text{heuristic (not necessary get optimal)} \\ \text{Have probability go out of local optimal} \end{cases}$

SICK
Sensor Intelligence.

Date: 2021 Dec 4

③ Dynamic Programming

(i) Core IDEA: break complex problem into simple ones
(Divide & conquer, recursion)

(ii) Examples.

e.g.1 Chance = draw a die at most 3 times. Earnings = face value of a die. Option = Stop after each row.

Strategy: (DP) consider the third row $\mathbb{E}(3^{rd}) = \frac{\sum_{i=1}^{6} i}{6} = 3.5$

⇒ consider the second row { Stop if $X_2 > 3.5$ or proceed to 3rd

$$\mathbb{E}(2^{nd}) = \frac{3}{6} \times \left(\frac{4+5+6}{3}\right) + \frac{3}{6} \times 3.5 = 4.25$$

⇒ consider the first row { Stop if $X_1 > 4.25$ or proceed to 2nd

$$\mathbb{E} = \mathbb{E}(1^{st}) = \frac{2}{6} \times \left(\frac{5+6}{2}\right) + \frac{4}{6} \times 4.25 = \frac{14}{3}$$

e.g.2 (The Knapsack Problem)

A set of n items, each with weight w_i & value v_i, find maximum total value, given the capacity of knapsack W.

$$\underset{x}{\text{maximize}} \sum_{i=1}^{n} v_i x_i$$
$$\text{subject to} \sum_{i=1}^{n} w_i x_i \le W$$
$$x_i \in \{0,1\},\ i = 1, \dots, n$$

Integer Programming

Date: 2021 Dec 5

Steps: split to multi-stages; identify state & action; identify the opt. cost-to-go; recursive relationship

Sol. $m(i,w)$ denotes maximum total value can be obtained using the first i items with capacity limit w.

☆
$$\begin{cases} m(0,w) = 0, & \forall\ 0 \le w \le W \\ m(i,w) = \max\{m(i-1, w-w_i) + v_i + m(i-1,w)\}, & \forall\ \substack{1 \le i \le n \\ 0 \le w \le W} \end{cases}$$

(The complexity: $O(nW)$ given that $w_i \in \mathbb{Z}$)

⇒ Use form to solve:

$V(i,w)$	0	1	2	3	…
0					
1					
2					
…					

e.g.3 (Matrix Product Parenthesization)

Suppose we need to multiply a series of matrices: $A_1, \dots, A_n$

Given the dimensions, find the best way to add parenthesis.

$A_i \in R^{p_{i-1} \times p_i}$.

Sol. Let $m(i,j)$ be the minimum number of operations to compute $A_i \cdots A_j$. then

☆
$$m(i,j) = \begin{cases} 0, & \text{if } i = j \\ \min\limits_{i \le k < j} \{m(i,k) + m(k+1,j) + p_{i-1} p_k p_j\} \end{cases}$$

(step 1 $m(1,2), m(3,4), \dots, m(n-1,n)$ ⇒ step II $m(1,3), m(2,4), \dots$ ⇒ step n-1 $m(1,n)$ = target)

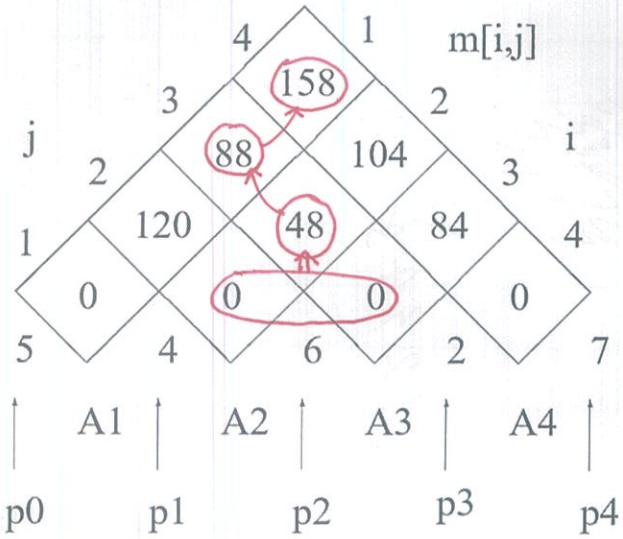
Complexity: $O(n^2)$

An example: $A_1 A_2 A_3 A_4$ with $p_0 = 5$, $p_1 = 4$, $p_2 = 6$, $p_3 = 2$, $p_4 = 7$

25th ANNIVERSARY SICK CHINA

SICK
Sensor Intelligence.

Date: 2021 Dec 5



The optimal steps & all m(i,j) in the graph.

e.g.4. (Gene/DNA sequence alignment)

Given two strings: $x = x_1 x_2 \cdots x_M$, $y = y_1 y_2 \cdots y_N$. Alignment - assignment of gaps to positions $0, \cdots M$ in x & $0, \cdots, N$ in y, so as to line up each letter in one sequence with either a letter or a gap in another sequence. (like AGGCTAGTT / AGCGAAGTTT ⇒ AGGC-TAGTT / AG-CGAAGTT)

Sol. Define scoring function (match +m, gap -d, mismatch -s)

scoring func ← $F(i,j) = \max\{ F(i-1,j-1) + S(x_i, y_j),\ F(i-1,j) - d,\ F(i,j-1) - d \}$

Recursive DP procedure (consider the last letter of the 2)

Complexity: $O(MN)$

Date: 2021 Dec 5

SICK
Sensor Intelligence.

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

☆ 2 assumptions for DP:

i) Memoryless: Decisions made "sequentially" in stages, depending on (only) states.

ii) Additive Cost/reward function (backward seperable & forward monotonic)

e.g.5 (Expected Utility Maximization / EUM)

Discrete time market model: $R_f = 1 + r$ (risk-free)

(Stock) R_t i.i.d overtime. Goal: maximize $\mathbb{E}(U(W_{T1}))$

Date: 2021 Dec 5

where $W_{t+1} = W_t(\phi_t R_{t+1} + (1-\phi_t)R_f)$, $t = 0, 1, \cdots, T-1$ (wealth process)

(ϕ_t — decisions to make)

Sol. $J_0 = \mathbb{E}(U(W_T))$, $J_t = \mathbb{E}(J_{t+1}|W_t)$, $J_T = U(W_T)$

☆ Let $J_t^*(W_t) = \underset{\phi_t, \cdots \phi_{T-1}}{\text{maximize}}\ \mathbb{E}(U(W_T)|W_t)$

Bellman Equation $J_t^*(W_t) = \max_{\phi_t} \mathbb{E}[J_{t+1}^*(W_t(\phi_t R_{t+1} + (1-\phi_t)R_f))|W_t]$ (decide)

(Reason: $\mathbb{E}(J_{t+1}|W_{t+1})|W_t) = \mathbb{E}(\mathbb{E}(J_{t+2}|W_{t+2})|W_t) = \mathbb{E}(J_{t+2}|W_t)$
$= \cdots = \mathbb{E}(U(W_T)|W_t)$)

(APP) Consider $S_{t+1} = \begin{cases} uS_t, & \text{with } p \\ dS_t, & \text{with } 1-p \end{cases}$ & CRRA $(U(W_T) = \frac{1}{1-r}W_T^{1-r})$

$\Rightarrow$ (backward induction. $J_t^* = A_t W_t^{1-r}$) $\phi_t = \frac{\mu - r}{r\sigma^2}$

with $u = 1+\mu\Delta t + \sigma\sqrt{\Delta t}$, $d = 1+\mu\Delta t - \sigma\Delta t$, $R_f = 1+r\Delta t$, $p = q = \frac{1}{2}$.

④ Graph Model.

(i) Vertex, Edge, adjacent, degree... (see in CSC3100 course)

(ii) Examples.

e.g.1 (Maximum-Flow Problems)

Descriptions & optimization formulation (see in MAT3007)

$$\text{maximize } z = \sum_j x_{sj}$$
$$\text{subject to } \sum_i x_{ij} = \sum_k x_{jk},\ \forall j = V(G)\setminus\{s,t\}$$
$$0 \le x_{ij} \le u_{ij},\ \forall ij \in A(G) \text{ (arc set)}$$

Sol. ☆ The Ford-Fulkerson Algorithm:

Step 1: Initidize — $f_c = 0$

Step 2: Find a directed path from s to t, if no such path, stop. f_{max} = current f_c

Step 3: compute u_{min} = minimum capacity of all arcs in the current path.

Step 4: update the residual capacity. $u_{ij} = u_{ij} - u_{min}$, $u_{ji} = u_{ji} + u_{min}$
(If $u_{ij} \to 0$, remove u_{ij})

Step 5: Update $f_c = f_c + u_{min}$ ← iterations

e.g.2 (The shortest path problem)

Given a graph $G = (V(G), E(G))$ & vertex s, t. length c_{ij}, if $ij \in E(G)$. Find the shortest path from s to t along the edges

Sol. ☆ Dijkstra's Algorithm: (idea of DP)

Step 1: Initialize — $L(s) = 0$, $L(i) = \infty$, $\forall i \in V(G)\setminus\{s\}$

Step 2: Find the vertex with smallest temporary label (randomly pick when tie/draw) Make it permanent.

Step 3: Update the temporary label, based on permanent ones as $L(j)^{temp} = \min_i \{L(i)^{perm} + c_{ij}\}$ ☆ ← iterations

SICK
Sensor Intelligence.

Date: 2021 Dec 5

e.g.3 (Vertex Cover Problem)

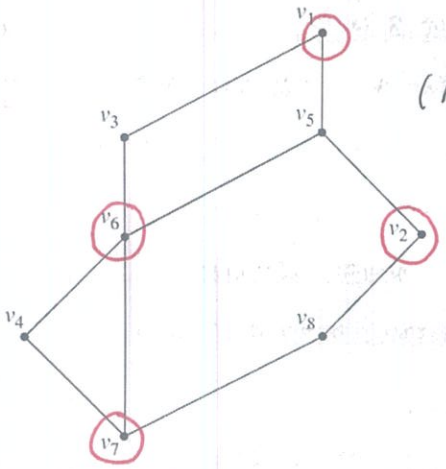
Given $G=(V(G), E(G))$. Find $S \subseteq V(G)$ as the smallest size s.t. every edge in $E(G)$ can find its at least 1 endpoint in S.

minimize $\Xi = \sum_{i \in V(G)} x_i$

subject to $x_i + x_j \geq 1, \ \forall ij \in E(G)$

$x_i \in \{0,1\}, \ \forall i \in V(G)$

(NP-complete problem!)



e.g.4 (Euler's seven bridges Problem)

Can a graph $(V(G), E(G))$ be drawn without repetition in one draw? (Or find a closed walk tranversing every edge exactly once.)

e.g.5 (Four color problem)

Using only 4 colors to color the vertex of a graph s.t. no 2 vertexs with an edge connected have the same color.

SICK
Sensor Intelligence.

Date: 2021 Dec 14

• PART V — Statistics Models

① Empirical Model

(i) Defn: infer a function relationship $y = f(x;\theta)$ based on observations (x_i, y_i), $i = 1, 2, \cdots, n$

traits:
- NOT care about how x affects y
- NOT imply causal effect of x on y
- provide information of y for x in the data set

(ii) Error sources: model error (misspecification, missing factors) & data error (truncation, round-off, measurement)

(iii) Model Fitting Criteria: solving minimize distances

$\min_{\theta \in \Theta} d(\{y_i\}_{i=1}^m, \{f(x_i;\theta)\}_{i=1}^m)$ (distances between 2 sets)

i) Chebyshev Approximation (L_∞ norm)

$$d(\{y_i\}_{i=1}^m, \{f(x_i;\theta)\}_{i=1}^m) = \max_{1 \leq i \leq m} |y_i - f(x_i;\theta)|$$

ii) Sum of Absolute Deviation (L_1 norm)

$$d(\{y_i\}_{i=1}^m, \{f(x_i;\theta)\}_{i=1}^m) = \sum_{i=1}^m |y_i - f(x_i;\theta)|$$

iii) Least square criterion (L_2 norm)

$$d(\{y_i\}_{i=1}^m, \{f(x_i;\theta)\}_{i=1}^m) = \sum_{i=1}^m (y_i - f(x_i;\theta))^2$$

Relating criteria:
- $c_i = |y_i - f_1(x_i)|$, $\xrightarrow{\text{by chebyshev}}$ $c_{max} = \max_{1 \leq i \leq m} c_i$
- $d_i = |y_i - f_2(x_i)|$, $\xrightarrow{\text{by least square}}$ $d_{max} = \max_{1 \leq i \leq m} d_i$

$$D = \sqrt{\frac{d_1^2 + \cdots + d_m^2}{m}} \leq \sqrt{\frac{c_1^2 + \cdots + c_m^2}{m}} \leq c_{max} \leq d_{max}$$

If $|D - c_m| < |c_m - d_m|$ Chebyshev!

SICK
Sensor Intelligence.

Date: 2021 Dec 14

Linear & generalized linear model ($\hat{\theta} = (G^TG)^{-1}G^Ty$)

Transformed Least-square fit e.g. $f(x;\theta) = ae^{bx} \Rightarrow \sum_{i=1}^{m} (\ln y_i - \ln a - bx_i)^2$

(the transformed minimizer is not necessarily the original one)

(iv) Choose the best model:

☆ Quantitive indicators: max deviation; $\sum$|deviation|; $\sum$ deviation^2

☆ Qualitative indicators: # parameters (less is better); obvious characteristics of the data (trend); patterns of deviations ($e(x_i)$)
↘ good model: no explicit patterns on $e(x)$.

(v) High-order polynomial models

Lagrangian form of the polynomial $P(x) = \sum_{i=1}^{m} y_i L_i(x)$

where $L_k(x) = \prod_{i \neq k} \frac{x - x_i}{x_k - x_i}$. (Unique polynomial $P(x)$ of at most degree $m-1$)

Advantages: perfect fitting, easily integrated & differentiated.

Disadvantages: too oscillatory
- possible wrong trends → in reality, not used
- sensitive to measurement errors

Smoothing – the procedure to use lower order polynomials to mitigate the disadvantages.

Divided Difference Table

Data		First divided difference	Second divided difference
x_1	y_1		
x_2	y_2	$\frac{y_2 - y_1}{x_2 - x_1}$	$\frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$
...		.....	

SICK
Sensor Intelligence.

Date: 2021 Dec 14

Choose: constant, if 0; linear, if first $\neq 0$, second $= 0$
& n-order if $0, \cdots, n \neq 0, n+1 = 0$.

(If all not zero, polynomial model may not fit.)

Interpolation (linear, cubic spline)

(Natual spline, $S_1''(x_1) = S_2''(x_3) = 0$; Clamped spline $S_1'(x_1) = f'(x_1)$, $S_2'(x_3) = f'(x_3)$)

② Big Data Analysis

(i) Revisiting generalized regression models:

Infer function f in $Y = f(X) + \varepsilon$ → with 0 mean

$\Rightarrow f(x) = \mathbb{E}(Y|X)$, $Y \in \mathbb{R}$, $X \in \mathbb{R}^p$

(Linear) regression assumes: $f(x) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$

(sources of X: quantitative inputs, basis expansions ($X_2 = X_1^2$, $X_3 = X_1^3$, dummy variables (e.g. (1,0), (0,1)), interactions ($X_3 = X_1 X$

☆

(ii) The Gauss-Markov Theorem: the least square estimates of parameter β have the smallest variance among all linear unbiased estimates (BLUE property – best linear unbiased estimate)

i.e. $\forall a$, suppose $\tilde{\theta} = c^Ty$ is unbiased for $a^T\beta$

$Var[a^T\hat{\beta}] \leq Var[c^Ty]$ $\quad E[\tilde{\theta}] = a^T\beta$

$\forall$ estimator $\hat{\theta}$ for θ $\quad MSE[\hat{\theta}] = \mathbb{E}[(\hat{\theta} - \theta)^2]$

(mean squared error) $= Var[\hat{\theta}] + (\mathbb{E}[\hat{\theta}] - \theta)^2 = Var[\hat{\theta}] + Bia$

Trade-off between variance & bias

SICK Sensor Intelligence.

Date: 2021 Dec 15

dim of Θ $\begin{cases} \uparrow - \text{var}\uparrow, \text{bias}\downarrow & \text{(overfitting)} \\ \downarrow - \text{var}\downarrow, \text{bias}\uparrow & \text{(underfitting)} \end{cases}$

(iii) Model Selection:

☆ Forward-Stepwise: starts with the intercept (β_0), then sequentially adding "β_i" into model to improve the fitting. ($\beta_0 \to \overset{\beta_1}{x_1} \cdots \overset{\beta_p}{x_p}$ fit n times, & choose best)

☆ Backward-Stepwise: starts with the full model, sequentially delete the predictor with the smallest Z_k-score ($= \frac{\text{coefficient}}{\text{s.t.d error}}$)

↳ Only work for $p<N$ (otherwise X^TX not pos. def.)

☆ Forward-Stagewise: starts with forward-stepwise with $\beta_0=\bar{y}$, adds the most correlated predictor every time (for some variables), stop when all variables NOT correlated with residuals ⇒ Fit for $p<N$

Other popular ways

1) Ridge Regression $\boxed{\min_\beta \|y-X\beta\|^2 \text{ s.t. } \beta^T\beta-\beta_0^2 \le t}$

(W.l.o.g. assume $\bar{y}=0$, $\bar{x}=0$, $\beta\in\mathbb{R}^p$, $\hat{\beta}^{ridge}=(X^TX+\underset{\text{penalty}}{\lambda}I)^{-1}X^Ty$)

Degree of freedom $df(\lambda)=\sum_{j=1}^{p}\frac{d_j^2}{d_j^2+\lambda}$, d_j's are eigenvalues of X^TX.

2) LASSO $\boxed{\min_\beta \|y-X\beta\|^2 \text{ s.t. } \|\beta\|_1-|\beta_0|\le S}$ more strict than ridge regression

(solved with proximal gradient descent)

Date: 2021 Dec 15

3) Cross Validation: K-fold cross validation (usually $K\sim 10$):

Every time (break into $\frac{n}{K}$ size) train the model with $\frac{n}{K}-1$ data groups / variables, validation & checking errors with the chosen / (choose the least test data (mean squared error (MSE)))

(iv) Logistic Regression (classification)

⇒ Aim: infer $\mathbb{E}(Y=0|X)$, $\mathbb{E}(Y=1|X)$ (linear models not good)

⇒ A non-linear transformation $P(Y=0|X)=\frac{1}{1+\exp(X\beta)}$

⇒ Fitting method: maximize log-likelihood

$\max \ln \prod_{i=1}^{N}\left(\frac{1}{1+\exp(x_i^T\beta)}\right)^{1-y_i}\left(\frac{\exp(x_i^T\beta)}{1+\exp(x_i^T\beta)}\right)^{y_i}$ (Gradient Descent Method)

⇒ Decision Boundary $\ln\left(\frac{P[Y=1|x]}{P[Y=0|x]}\right)=0 \Leftrightarrow x^T\hat{\beta}=0$ (hyperplane)

☆ For multiple cases, need K vectors $\beta_1,\cdots,\beta_K$ $P(Y=\beta_i|X)=\frac{\exp(X\beta_i)}{1+\sum_{m=1}^{K}\exp(X\beta_m)}$

Decision Boundary: $\hat{c}(x)=\arg\max_{k\in\{0,\cdots,K\}} x^T\hat{\beta}_k$ (with $\hat{\beta}_0=0$)

☆ L_1-regularized Logistic Regression: $\max_\beta\left(\sum_{i=1}^{N} y_i x_i^T\beta-\ln(1+\exp(x_i^T\beta))\right)-\lambda\sum_{j=1}^{p}|\beta_j|$ (LASSO)

(v) Data Compression & PCA (principal component analysis)

PCA: Give $x_1,\cdots x_N\in\mathbb{R}^d$ (large N & d), we want a close representation in $\mathbb{R}^{d'}$ with $d'\ll d$.

i.e. $\mu+\sum_{j=1}^{d'}\lambda_j v_j=\mu+V_{d'}\lambda$ ($\lambda\in\mathbb{R}^d$, $V_{d'}\in\mathbb{R}^{d\times d'}$, $V_{d'}^TV_{d'}=I$)

SICK
Sensor Intelligence.

Date: 2021 Dec 15

minimize the reconstruction error $\min_{\mu,\{\lambda_i\},V_{d'}} \sum_{i=1}^{N} \|x_i-\mu-V_{d'}\lambda_i\|^2$

Another approach (SVD) $X=UDV^T$, with $D_{11}\geq D_{22}\geq\cdots$

Optimal $V_{d'}$ consists of first d' column of V

$(X^T=(x_1-\bar{x}, x_2-\bar{x}, \cdots x_N-\bar{x}))$ $X\in\mathbb{R}^{d'\times d}$

(vi) Data Clustering — partition into groups. pairwise dissimilarities between same cluster < different ~s.

* Combinatorial Algorithm (no probability model)
* Mixture Modeling (suppose all data is i.i.d.)

K-mean clustering:

1° For a given assignment (cluster) C, minimize the total cluster variance w.r.t. $(m_1,\cdots,m_K)$ yield means of currently assigned clusters. $\bar{x}_S=\arg\min_{m_S}\sum_{i\in S}\|x_i-m_S\|^2$

2° Given a current set of means $(m_1,\cdots,m_K)$ assign each observation $C(i)=\arg\min_{1\leq k\leq K}\|x_i-m_K\|^2$

3° Repeat 1° & 2°, stop when the assignment does NOT change

EM Algorithm (expectation & maximize)

Density for K i.i.d classes (Gaussian Distribution)

$$\Phi_{\theta_K}(x)=\frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma_K|}}\exp\left((x-\mu_K)^T\Sigma_K^{-1}(x-\mu_K)\right)$$

1° Take intinal guesses on $\hat{\mu}_K$, $\hat{\Sigma}_K$, $\hat{\pi}_K$ (proportion of data in K)

Date: 2021 Dec 15

2° Expectation step: (Baye's Theorem)

$$\hat{\gamma}_{i,k}=\frac{\hat{\pi}_k\Phi_{\hat{\theta}_k}(x_i)\rightarrow \text{likelyhood}}{\sum_{k=1}^{K}\hat{\pi}_k\Phi_{\hat{\theta}_k}(x_i)}$$

3° Maximization step: $\hat{\mu}_k=\frac{\sum_{i=1}^{N}\hat{\gamma}_{i,k}x_i}{\sum_{i=1}^{N}\hat{\gamma}_{i,k}}$ ← weighted average

$$\hat{\Sigma}_k=\frac{\sum_{i=1}^{N}\hat{\gamma}_{i,k}(x_i-\bar{x})(x_i-\bar{x})^T}{\sum_{i=1}^{N}\hat{\gamma}_{i,k}}$$ ← weighted covariance estimation

$$\hat{\pi}_k=\frac{\sum_{i=1}^{N}\hat{\gamma}_{i,k}}{N}$$

4° Repeat 2° & 3°. Stop when convergence.

MAT3300 Mathematical Modeling: Midterm Exam
March 25, 2019

Answer the questions in the answer book.

Question:	1	2	3	4	5	6	Total
Points:	10	15	20	20	20	15	100
Score:							

Name: _____ Student ID: _____ Time limit: 2 hours

1. (10 points) Consider the multiple linear regression model $\mathbf{y} = \mathbf{a}x + \mathbf{b}$ where $\mathbf{y}, \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ and $x \in \mathbb{R}$. We regress this model to a data set $(x_i, \mathbf{y}_i)$, $i = 1, 2, \cdots, n$ by the least square criteria, i.e., minimizing $\sum_{i=1}^n \|\mathbf{y}_i - \mathbf{a}x_i - \mathbf{b}\|_2^2$ where $\|\mathbf{z}\|_2 = \sqrt{\sum_{j=1}^d \mathbf{z}_j^2}$ for $\mathbf{z} \in \mathbb{R}^n$. Find the optimal $\mathbf{a}, \mathbf{b}$ analytically.

To minimize the function $f(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^n (\boldsymbol{y_i} - \boldsymbol{a}x_i - \boldsymbol{b})^T (\boldsymbol{y_i} - \boldsymbol{a}x_i - \boldsymbol{b})$, we have to find the station point of the function, which is

$$\nabla_{\boldsymbol{a}} f(\boldsymbol{a}, \boldsymbol{b}) = 0, \quad \Rightarrow \quad \sum_{i=1}^n -2x_i(\boldsymbol{y_i} - \boldsymbol{a}x_i - \boldsymbol{b}) = 0,$$
$$\nabla_{\boldsymbol{b}} f(\boldsymbol{a}, \boldsymbol{b}) = 0, \quad \Rightarrow \quad \sum_{i=1}^n -2(\boldsymbol{y_i} - \boldsymbol{a}x_i - \boldsymbol{b}) = 0, .$$

Solve the equations, we have

$$\boldsymbol{a} = \frac{\sum_{i=1}^n (x_i - \bar{x})(\boldsymbol{y_i} - \bar{\boldsymbol{y}})}{\sum_{i=1}^n (x_i - \bar{x})^2},$$
$$\boldsymbol{b} = \bar{\boldsymbol{y}} - \boldsymbol{a}\bar{x},$$

where $\bar{x} = \sum_{i=1}^n x_i/n$, $\bar{\boldsymbol{y}} = \sum_{i=1}^n \boldsymbol{y}_i/n$.

2. Consider the following population model for two species.

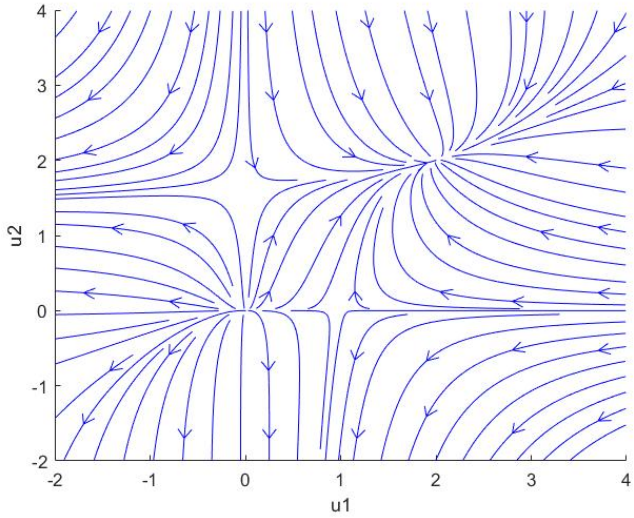
$$u_1'(t) = u_1(t)(1 - u_1(t) + 0.5u_2(t)),$$
$$u_2'(t) = u_2(t)(2.5 - 1.5u_2(t) + 0.25u_1(t)).$$

Here $u_1(t)$ and $u_2(t)$ are the populations (in suitable units, say thousands of animals) at time t of two species that share the same environment.

(a) (5 points) Find all the equilibrium points of the system.

(b) (5 points) Identify the stability properties of these equilibrium.

(c) (5 points) Is it the case that, for all initial populations with $u_1(0) > 0$ and $u_2(0) > 0$, the trajectories all converge to the same equilibrium point? Explain by sketching phase lines.



(a) Let

$$u_1(t)(1 - u_1(t) + 0.5u_2(t)) = 0,$$
$$u_2(t)(2.5 - 1.5u_2(t) + 0.25u_1(t)) = 0.$$

Solve the equation system, we can obtain the equilibrium points are

$$① \begin{cases} u_1(t) = 0, \\ u_2(t) = 0, \end{cases} \quad ② \begin{cases} u_1(t) = 0, \\ u_2(t) = \frac{5}{3}, \end{cases} \quad ③ \begin{cases} u_1(t) = 1, \\ u_2(t) = 0, \end{cases} \quad ④ \begin{cases} u_1(t) = 2, \\ u_2(t) = 2, \end{cases}$$

(b) From the phase diagram, we can easily see the result. The point ①②③ are unstable nodes. The point ④ is a stable node.
(c) YES. From the phase diagram, we can see that when $u_1(0) > 0$ and $u_2(0) > 0$, there is only one equilibrium point, which is a stable node. Therefore, the trajectories all converge to the same equilibrium point.

3. Consider the following PDE for $u(t, x)$.

$$\begin{cases} \partial_t u = \partial_{xx} u + \gamma \partial_x u, \ t > 0, \ -\infty < x < \infty, \\ u(0, x) = f(x). \end{cases}$$

(a) (5 points) Find the PDE satisfied by $v(t, x) = u(t, x)/g(x)$ where $g(x)$ is a known function.

(b) (5 points) Find an appropriate $g(x)$ such that the PDE of $v(t, x)$ does not involve $\partial_x v$.

(c) (10 points) Solve for $v(t, x)$ based on the $g(x)$ you find in (b) and then find $u(t, x)$ for the original PDE.

(a) Substitute $u(t,x) = g(x)v(t,x)$ in the original function, we can get

$$\begin{cases} g(x)\partial_t v(t,x) = (\partial_{xx} g(x) + \gamma \partial_x g(x))v(t,x) + (2\partial_x g(x) + \gamma g(x))\partial_x v(t,x) + g(x)\partial_{xx} v(t,x), \\ g(x)v(0,x) = f(x). \end{cases}$$

(b) From the condition, it's easy to see that,

$$2\partial_x g(x) + \gamma g(x) = 0.$$

The solution is $g(x) = ce^{-\frac{\gamma x}{2}}$.
(c) From the solution of (b), we have

$$\partial_{xx} g(x) + \gamma \partial_x g(x) = -\frac{\gamma^2}{4} g(x).$$

Then the original PDE can be transformed to

$$\begin{cases} \partial_t v(t,x) = -\frac{\gamma^2}{4} v(t,x) + \partial_{xx} v(t,x), \\ v(0,x) = \frac{1}{c} e^{\frac{\gamma x}{2}} f(x) \end{cases}$$

Here, we do not consider the trivial solution $g(x) = 0$. Let $\hat{v}(t,\omega) = \mathcal{F}[v(t,x)](\omega)$. Taking Fourier transform on both sides,

$$\begin{cases} \partial_t \hat{v}(t,\omega) = -\frac{\gamma^2}{4} \hat{v}(t,\omega) - \omega^2 \hat{v}(t,\omega), \\ v(0,x) = \frac{1}{c} \hat{f}(\omega + i\gamma) \end{cases}$$

Solve the ODE, we have

$$\hat{v}(t,\omega) = \frac{1}{c} \hat{f}(\omega + i\gamma) e^{-(\frac{\gamma^2}{4} + \omega^2)t}$$

By using the IFT,

$$v(t,x) = \frac{1}{2c\sqrt{\pi t}} e^{-\frac{\gamma^2 t}{4}} \int_{-\infty}^{\infty} f(\xi) e^{-(x-\xi)^2/(4t)} d\xi$$

With the solution in (b), we can find that

$$u(t,x) = \frac{1}{2\sqrt{\pi t}} e^{-\frac{\gamma^2 t + 2\gamma x}{4}} \int_{-\infty}^{\infty} f(\xi) e^{-(x-\xi)^2/(4t)} d\xi$$

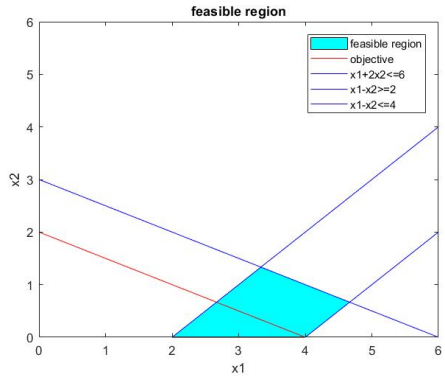
4. Consider the following linear programming (LP) problem.

   Minimize $z = 3x_1 + 6x_2$, subject to $x_1 + 2x_2 \leq 6,\ x_1 - x_2 \geq 2,\ x_1 - x_2 \leq 4,\ x_1,\ x_2 \geq 0$.

   (a) (5 points) Sketch the feasible region of this LP problem and solve it graphically.

   (b) (5 points) Enumerate all extreme points of the feasible region and determine the optimal solution.

   (c) (10 points) Transform the LP problem into the standard form and solve it with the simplex method.



(a) As shown in the figure.
(b)There are four extreme points, which are

$$\begin{cases} x_1 = 2, \\ x_2 = 0, \end{cases} \quad \begin{cases} x_1 = 4, \\ x_2 = 0, \end{cases} \quad \begin{cases} x_1 = \frac{10}{3}, \\ x_2 = \frac{4}{3}, \end{cases} \quad \begin{cases} x_1 = \frac{14}{3}, \\ x_2 = \frac{2}{3}, \end{cases}$$

From the picture, we can see that the first point is the solution, $z = 6$.
(c)The standard form is

$$\begin{aligned} \max \quad & z' = -3x_1 - 6x_2 \\ s.t. \quad & \begin{cases} x_1 + 2x_2 + s_1 = 6, \\ x_1 - x_2 - s_2 = 2, \\ x_1 - x_2 + s_3 = 4, \\ x_1, x_2, s_1, s_2, s_3 \geq 0. \end{cases} \end{aligned}$$

To remedy the predicament, artificial variables are created. For $\geq$ or $=$ constraint, add artificial variables. Add sign restriction $s_i \geq 0$. The problem can be converted as

$$\begin{aligned} \max \quad & z' = -3x_1 - 6x_2 - Ms_4 \\ s.t. \quad & \begin{cases} x_1 + 2x_2 + s_1 = 6, \\ x_1 - x_2 - s_2 + s_4 = 2, \\ x_1 - x_2 + s_3 = 4, \\ x_1, x_2, s_1, s_2, s_3, s_4 \geq 0. \end{cases} \end{aligned}$$

where M denotes a very large positive number.
Build the tableau:

		x_1	x_2	s_1	s_2	s_3	s_4	b
		-3	-6	0	0	0	$-M$	
0	s_1	1	2	1	0	0	0	6
$-M$	s_4	[1]	-1	0	-1	0	1	2
0	s_3	1	-1	0	0	1	0	4
	σ	$-3+M$	$-6-M$	0	$-M$	0	0	
0	s_1	0	3	1	1	0	-1	4
-3	x_1	1	-1	0	-1	0	1	2
0	s_3	0	0	0	1	1	-1	2
	σ	0	-9	0	-3	0	$-M+3$	

From the table, we can see that, $x_1 = 2$, $s_3 = 2$, $s_1 = 4$. Hence, $x_2 = 0$, $s_2 = 0$. The solution is $\max z' = -6$, $\min z = 6$.

5. You have to design a 3-D block as a water storage. The volume of the block has to be at least 9 cubic meters. The base area of the block is at most 6 square meters, while the height of the block is at least 1 and at most 2 meters. Your task is to design the block that satisfies the above conditions and the difference between the base area and the height of the block is maximal.

   (a) (5 points) Formulate the above problem as a nonlinear programming (NLP).

   (b) (5 points) Show that the height is not 1 at an optimal solution.

   (c) (5 points) Write down the KKT condition of the NLP in (a).

   (d) (5 points) Check if the height=1.5 meters, base-area=6 square meters corresponds to an optimal solution.

(a)

$$\begin{aligned} \max \quad & f = s - h \\ s.t. \quad & \begin{cases} sh \geq 9, \\ s \leq 6, \\ 1 \leq h \leq 2, \\ s \geq 0. \end{cases} \end{aligned}$$

where $s = ld$.
(b) If $h = 1$, then from the first condition $ld \geq 9$ which contradicts with the second condition. Hence it is not an optimal solution.
(c)

$$\begin{aligned} \min \quad & f(h,s) = h - s \\ s.t. \quad & \begin{cases} g_1(h,s) = -sh + 9 \leq 0, \\ g_2(h,s) = s - 6 \leq 0, \\ g_3(h,s) = h - 2 \leq 0, \\ g_4(h,s) = -h + 1 \leq 0, \\ g_5(h,s) = s \geq 0. \end{cases} \end{aligned}$$

The KKT condition is

$$\nabla_h f(h,s) + \sum_{i=1}^{5} \mu_i \nabla_h g_i(h,s) = 1 + \mu_1 s + \mu_3 - \mu_4 = 0,$$

WRONG answer!

$$\nabla_s f(h,s) + \sum_{i=1}^{5} \mu_i \nabla_s g_i(h,s) = -1 + \mu_1 h + \mu_2 - \mu_5 = 0,$$

$$\mu_j^* \geq 0, \qquad j = 1, \ldots, 5,$$

$$\mu_j^* g_j(h^*, s^*) = 0, \qquad j = 1, \ldots, 5.$$

where μ^* is the value of $\mu_1, \ldots, \mu_5$ at the optimal, $g_j(x^*)$ is the condition of the function.
(d) Substitute $h = 1.5, s = 6$ to the KKT condition, we have,

$$\begin{aligned} \mu_3^* g_3(1.5, 6) = -0.5\mu_3^* = 0, \quad & \Rightarrow \quad \mu_3^* = 0, \\ \mu_4^* g_4(1.5, 6) = -0.5\mu_4^* = 0, \quad & \Rightarrow \quad \mu_4^* = 0. \end{aligned}$$

Then, from the first KKT condition, $\mu_1^* = -1.6$ which contradicts with $\mu_1^* \geq 0$. Hence, it is not an optimal solution.

WRONG answer!

6. (15 points) Solve the following knapsack problem with dynamic programming. There are four items and the ith item has value v_i, size w_i listed in the table. The knapsack size limit is W. Find the maximum total value of items that can be put into the knapsack and the corresponding optimal strategy for any positive real W.

i	1	2	3	4
v_i	10	40	30	50
w_i	5	4	6	3

The problem can be written as

$$\max V = \sum_{i=1}^{4} v_i x_i,$$
$$s.t. \sum_{i=1}^{4} x_i w_i \leq W, \qquad x_i = 0, 1.$$

Let $m(i, w)$ denotes the maximum total value can be obtained using the first i items with capacity limit w.

$V(i,w)$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$i = 0$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	10
2	0	0	0	0	40	40	40	40	40	50	50	50	50	50	50	50	50	50	50
3	0	0	0	0	40	40	40	40	40	50	70	70	70	70	70	80	80	80	80
4	0	0	0	50	50	50	50	90	90	90	90	90	100	120	120	120	120	120	130

From the table, we can conclude that

$$V(i,w) = \begin{cases} 0, & 0 \leq w \leq 2, & x_i = 0, i = 1, 2, 3, 4, \\ 50, & 3 \leq w \leq 6, & x_4 = 1, \\ 90, & 7 \leq w \leq 11, & x_4 = x_2 = 1, \\ 100, & w = 12, & x_4 = x_2 = x_1 = 1, \\ 120, & 13 \leq w \leq 17, & x_4 = x_2 = x_3 = 1, \\ 130, & w \geq 18, & x_i = 1, i = 1, 2, 3, 4. \end{cases}$$

$\gamma(x,y,t)$	$\Delta_t = \Delta_x^2 = \frac{1}{4}$	$\Delta_t = \Delta_x^2 = \frac{1}{16}$	$\Delta_t = \Delta_x^2 = \frac{1}{64}$	$\Delta_t = \Delta_x^2 = \frac{1}{256}$
$\sin(xyt + \frac{2\pi}{5})$	2.4286×10^{-2}	7.8620×10^{-3}	2.0681×10^{-3}	5.3845×10^{-4}
$\cos(xyt + \frac{1}{100})$	2.3408×10^{-2}	7.4865×10^{-3}	1.9660×10^{-3}	5.0486×10^{-4}
$\frac{e^{xyt} - \sin(xyt)}{10}$	1.2208×10^{-2}	3.9143×10^{-3}	1.0309×10^{-3}	3.4173×10^{-4}
$\frac{e^{xyt} + \cos(xyt)}{20}$	1.1952×10^{-2}	3.8370×10^{-3}	1.0112×10^{-3}	3.3466×10^{-4}
$\frac{e^{xyt} - xyt}{8}$	1.1091×10^{-2}	3.5259×10^{-3}	9.2618×10^{-4}	3.0723×10^{-4}
$\frac{e^{xyt} - (xyt)^3}{12}$	1.2447×10^{-2}	4.0124×10^{-3}	1.0597×10^{-3}	3.4946×10^{-4}
$e^{xyt-2.5}$	1.2492×10^{-2}	4.0195×10^{-3}	1.0606×10^{-3}	3.5039×10^{-4}
$e^{-xyt-1.8}$	1.0361×10^{-2}	3.2696×10^{-3}	8.5603×10^{-4}	2.8827×10^{-4}
$\frac{\sqrt{xyt}+1}{15}$	1.2664×10^{-2}	4.1117×10^{-3}	1.0883×10^{-3}	3.6031×10^{-4}
$\frac{1+(xyt)^5}{9}$	1.1781×10^{-2}	3.7736×10^{-3}	9.9398×10^{-4}	3.2941×10^{-4}

Numerical Analysis

MAT 4001 Notebook

Youthy WANG

通 訊 錄 Address List

Name/姓名
Fax
E-Mail
微信 QQ

Name/姓名
Fax
E-Mail
微信 QQ

Name/姓名
Fax
E-Mail
微信 QQ

Name/姓名
Fax
E-Mail
微信 QQ

Name/姓名
Fax
E-Mail
微信 QQ

Mo Tu We Th Fr Sa Su

Memo No. 01
Date 2022 / Sep / 06

MAT 4001 Numerical Analysis

- Introduction & Motivations

From 1990's, with the development of computer.

Traditional / limited: Biology / physical / financial / material science / fluid mechanics / ... } modelling (based on some known rules) → mathematical models: ODE, PDE, non-linear system, optimization ...

↓ trans. into math languages

Numerical Analysis ↓ Approximate solution —interpret→

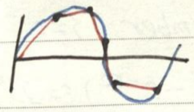
Deep Learning (today) — with Deep Neural Networks (AI for science)

*1. Some concrete topics below:

1) Numerically Solving Non-linear Equations

(e.g. 1) Find root for $f(x) = xe^x - 1$ on $[-1, 1]$

(e.g. 2) Lambert W function: inverse func of $f(w) = we^w$, find values of $w(z)$.

2) Interpolation  (different interpolations)

3) Numerical Integration:

(e.g.) $\int_0^2 e^{-\frac{t^2}{2}} dt$ } Taylor expansion (make sure convergence) $\sum \int_{\cdots}^{\cdots} e^{-\frac{t^2}{2}} dt$

4) Linear System: $Ax = b$ (or $Ax = \lambda x$ - eigenvalue prob)

Memo No. 02
Date 2022 / Sep / 06

5) Least Square Approximations:

*2. Computational Errors

1) Truncation Error: (made by algorithms with finite steps)
(截断)
(e.g.) $f(x) = P_n(x) + R(x)$ (Taylor theorem)

2) Raund-off Error: (made by computer – finite # of precision)
(e.g.) $\pi \approx 3.14159265358979\ldots$
Unavoidable!

3) Computer Arithmetic: binary system; 64-bit system – double precision.
gives floating point number $(-1)^s 2^{c-1023}(1+f)$ (s – 1 digit, c – 11 digits, f – 52 digits)

Smallest normalized machine number: $s=f=0$, $c=1$;
$(-1)^0 2^{1-1023}(1+0) = 2^{-1022}$ (Any number < that 2^{-1022} will cause underflow (positive) → neglected to be zero)

Largest normalized machine number: $s=0$, $c=\sum_{i=0}^{10} 2^i$, $f=\sum_{i=-1}^{-52} 2^{+i}$
we then get $2^{1023}(2-2^{-52}) \approx 2^{1024}$

4) (Defn) The error that results from replacing by its floating-point form is called round-off error. (regardless of whether chopping / rounding is applied)

Memo No. 03
Date 2022 / Sep / 08

(Defn) Suppose p^* is an approximation of p, we then call $p-p^*$ actual error, $|p-p^*|$ absolute error & $\left|\frac{p-p^*}{p}\right|$ relative error provided that $p \neq 0$.

(Defn) The number p^* is said to approximate p to t significant digits if $\left|\frac{p-p^*}{p}\right| \leq 5\times10^{-t}$ but $\left|\frac{p-p^*}{p}\right| > 5\times10^{-t-1}$

Q: How to avoid the loss of accuracy due to rounding / chopping?

Ans. (i) reformulate the problem (without subtraction of nearly equal numbers)

(ii) Horner's Method (秦九韶 Algorithm)
↳ express polynomials into nested forms
$f(x) = (\ldots(a_n x + a_{n-1})x + \cdots)\cdot x + a_0$ → minimized the total number of calculation

(iii) Avoid large number eat small numbers:
when doing $fl\left[\sum_{i=1}^{n} f(x_i)\right]$ (the way computer calculates), add small numbers firstly.
Replace using floating numbers

*3. Numerical Algorithm

1) Algorithms & Pseudo codes: (skipped)

2) Convergence & Stability:

(i) Rate of Convergence: (Defn) Suppose $\{\beta_n\}_{n=1}^{\infty} \to 0$, while $\{\alpha_n\}_{n=1}^{\infty} \to \alpha \in \mathbb{R}$, if $\exists K>0$, with $|\alpha_n - \alpha| \leq K|\beta_n|$ (For large n)
we say $\{\alpha_n\}_{n=1}^{\infty}$ converges to α with the rate / order of convergence $O(\beta_n)$, written as $\alpha_n = \alpha + O(\beta_n)$ (Usually $\beta_n = 1/n^p$)

Memo No. 04
Date 2022 / Sep / 08

Similarly, if $\lim_{h\to 0} F(h) = L$, with $|F(h)-L| \le Kh^p$ with small positive h, written as $F(h) = L + O(h^p)$.

(ii) Stability: "small" changes in initial data → "small" changes in results (stable)

(iii) Relationship: convergence — theoretical errors; stability — computation errors. (cannot imply)

• Solve Non-linear Equations Numerically

*1. Solns of Eqns with single variable (i.e. $f(x)=0$)

i) Bi-section Method: "concrete algorithm" — see in MAT3007 optimization Notes

☆ Convergence Analysis:

(i) Suppose that $f \in C[a,b]$ & $f(a)f(b)<0$. Bisection Method gives sequence $\{p_n\}_{n=1}^{\infty}$ approximating zero p of f.

We then have that $|p_n - p| \le \frac{b-a}{2^n}$ → rate of convergence "linear" $O(1/2^n)$

(proof.) $a_n - b_n = \frac{b-a}{2^{n-1}}$, by the algorithm, $n \in \mathbb{N}$

Because $p_n = \frac{a_n+b_n}{2}$ & $p \in [a_n, b_n]$, we get $|p_n - p| \le \frac{1}{2}(a_n - b_n) \overset{"="}{\le} \frac{b-a}{2^n}$.

(ii) Conservative Error Bound: the actual error may be quite small, compared with quite conservative error bound. (only a bound)

bi-section → Good for finding initial point (converge slowly!)

Memo No. 05
Date 2022 / Sep / 13

2) Fixed Point Iteration: want to solve $f(p)=0$, to achieve it, construct an auxiliary function $g(x)$, s.t. [$p=g(p)$ exactly when $f(p)=0$]. Then the question becomes to find "p" s.t. $p=g(p)$.

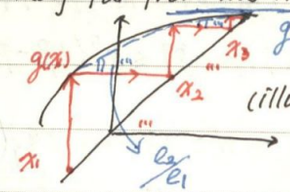
(e.g.) consider $f(x) = x - \cos x$. then $g(x) = \cos x$
(NOT unique construction)

(i) (Thm) Existence of fixed point: if $g \in C[a,b]$, $g(x) \in [a,b]$ $\forall x \in [a,b]$, then $\exists c \in [a,b]$ s.t. $g(c) = c$.
(proof is omitted)

(ii) (Thm) Uniqueness of fixed point: Provided that the above holds, with an addition that $|g'(x)| \le k < 1$, $\forall x \in [a,b]$. Then the fixed point is unique on $[a,b]$. (strong assumption, carefully find g!)

☆ Iteration Steps for [fixed-point algorithm]:
Every step let $x_{n+1} = g(x_n)$, stop if $|x_{n+1} - x_n| < \varepsilon$ (or $\left|\frac{x_{n+1}-x_n}{x_{n+1}}\right| < \varepsilon$). Note that in bi-section method, $|f(x_n)| < \varepsilon$ is NOT a good stop criterion, since x_n may far from the true soln.

suppose g cts. $\{p_n\} \to p$ ($n \to \infty$)
$p = \lim_{n\to\infty} g(p_{n-1}) = g(p)$



(illustration 1)

(e.g.) Suppose $x^3+4x^2-10=0$, some choice of $g(x)$: (start from 1.5)
- $x - x^3 - 4x^2 + 10$; (blow up)
- $\sqrt{\frac{10}{x} - 4x}$ (complex root); $\frac{1}{2}\sqrt{10-x^3}$; $\sqrt{\frac{10}{4+x}}$
- $x - \frac{x^3+4x^2-10}{3x^2+8x}$ (Newton's Method)

→ different convergence rate!

Memo No. 06
Date 2022 / Sep / 15

☆ Convergence Analysis : (from experiment, $\frac{e_n}{e_{n-1}} \to f'(p)$) a sense of convergence analysis

(iii) (Thm) Let $g \in C[a,b]$ with $g(x) \in [a,b]$, $\forall x \in [a,b]$ → existence
Let $g'(x)$ exist on (a,b) with $|g'(x)| \le k < 1$, $\forall x \in [a,b]$
$\forall$ point $p_0 \in [a,b]$, $\{p_n\}_{n=0}^{\infty}$ is defined by $p_n = g(p_{n-1})$. $\forall n \ge 1$, $n \in \mathbb{N}$
will converge to the unique fixed point $p \in [a,b]$.

(proof.) Consider $p_{n+1} - p_n = g(p_n) - g(p_{n-1}) \overset{MVT}{=} g'(\xi_n)[p_n - p_{n-1}]$
(w.l.o.g) with $\xi_n \in [p_{n+1}, p_n]$
$\therefore |p_{n+1} - p_n| = |g'(\xi_n)| \cdot |p_n - p_{n-1}| \le k |p_n - p_{n-1}| \quad \forall n \ge 1$
Thus $|p_{n+1} - p_n| \le k^n |p_1 - p_0| = k^n |g(p_0) - p_0| \le k^n (b-a) \to 0$
which means $\{p_n\}_{n=0}^{\infty}$ converges (C.C. or comparison test) (say, to p)
$\therefore p = \lim_{n\to\infty} p_{n+1} = \lim_{n\to\infty} g(p_n) \overset{cts}{=} g(\lim_{n\to\infty} p_n) = g(p)$. .□

(iv) (Corollary - convergence rate) With above hypothesis,
$|p_n - p| \le \frac{k^n}{1-k} |p_1 - p_0|$. → can be faster than bi-section with good k
only a bound!
(proof.) $|p_n - p| = |g(p_{n-1}) - g(p)| \overset{M.V.T}{=} |g'(\xi)| |p_{n-1} - p| \le k |p_{n-1} - p|$
$\therefore |p_n - p| \le k^n |p_0 - p|$ & $|p_n - p| \le k^{n-1} |p_1 - p|$
$(1-k)|p_n - p| \le k^n (|p_0 - p| - |p_1 - p|) \le k^n |p_0 - p_1|$. □ ← e_0

Back to the previous e.g. $|(\frac{1}{2}\sqrt{10 - x^3})'| = |\frac{3x^2}{4\sqrt{10-x^3}}| \le 3/\sqrt{2}$ > 1, but get "1.7", it is true.
$|(\sqrt{\frac{10}{4+x}})'| = |\frac{\sqrt{10}}{2}(4+x)^{-\frac{3}{2}}| \le \frac{\sqrt{2}}{10}$ & $|(x - \frac{x^3+4x^2-10}{3x^2+8x})'| = \frac{|(x^3+4x^2-(0)+(6x+8)|}{(3x^2+8x)^2}$
≤ 0.58. → can be used!
↑ Note that $g'(p) = 0$!!
"Fastest!"

Memo No. 07
Date 2022 / Sep / 15

Note than $x - x^3 - 4x^2 + 10 \notin [1,2]$ & No interval s.t. $|(\sqrt{\frac{10}{x}} - 4x)'| < 1$.
↑ diverges!

3) Newton's Method : hard to find in practice

☆ Derivation : Suppose $f \in C^2[a,b]$, let $p_0 \in [a,b]$ be an initial approximation to p s.t. $f(p_0) \ne 0$ & $|p_0 - p|$ is small enough.
Consider $f(p) = f(p_0) + f'(p_0)(p - p_0) + \frac{f''(\tilde{p})}{2!}(p - p_0)^2$ (Taylor's thm)
where $\tilde{p}$ between p & p_0 since $f(p) = 0$, we get $0 = f(p_0) + f'(p_0)(p - p_0)$
$+ \frac{f''(\tilde{p})}{2}(p - p_0)^2 \Rightarrow p_1 \approx p_0 - \frac{f(p_0)}{f'(p_0)}$ (Or graphically, use tangent lines to estimate zeros)
drop "linearization"

(i) Iteration steps : similar to "bi-section" & "fixed point" with
$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}$. (Fastest among all fixed-point functions!)

(ii) Convergence Analysis :
(Thm) Let $f \in C^2[a,b]$, if $p \in (a,b)$ is such that $f(p) = 0$ & $f'(p) \ne 0$
Then there exists a $(\delta > 0)$ s.t. Netwon's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ defined by $p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$ converges to p
with $p_0 \in [p - \delta, p + \delta]$. ← weak

(proof.) Write $g(x) = x - \frac{f(x)}{f'(x)}$ (idea - use the results from "fixed point")
Let $k \in [0,1)$, Since f' is continous & $f'(p) \ne 0$, $\exists \delta_1 > 0$
s.t. $f'(x) \ne 0$, $\forall x \in V_{\delta_1}(p)$. Thus, g is well-defined & continuous on
$V_{\delta_1}(p)$. $g'(x) = \frac{-f(x)f''(x)}{(f'(x))^2}$, $\forall x \in V_{\delta_1}(p)$. $g'(x)$ is continuous, for

$f \in C^2[a,b]$. $\exists \delta_2 > 0$ s.t. $|g'(x)| \le k$, $\forall x \in V_{\delta_2}(p)$, since $g'(p)=0$ ($\delta_2 < \delta_1$, obviously). For $x_0 \in [p-\delta_2, p+\delta_2]$, MVT $\Rightarrow |g(x_0)-g(p)| \le |g'(\xi)||x_0-p| \le k\cdot\delta_2 < \delta_2$, thus $|g(x_0)-g(p)| = |g(x_0)-p| < \delta_2$

(ξ between x_0 & p)

$\therefore g(x_0) \in V_{\delta_2}(p)$. Thus $g(x) \in [p-\delta_2, p+\delta_2]$ ($x \in [p-\delta_2, p+\delta_2]$). Let $\delta_2 = \delta$, with thm for fixed-point iteration, $\{p_n\}_{n=1}^{\infty} \to p$.

☆ Remark: seldom used in practice, for we don't know δ generally. An initial approx. should be carefully selected.

(iii) Defects with Newton's Method:

Need to know f' at every approximation

↳ generally, f' is hard & cost many computations to get.

To approximate $f'(p_n)$ by $\frac{f(p_{n-1})-f(p_n)}{p_{n-1}-p_n}$ → secant

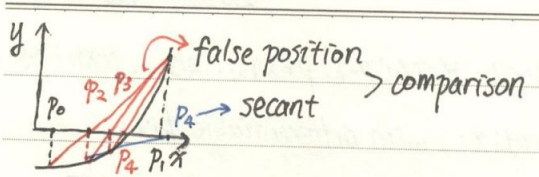
called↳ Secant Methods → only one func. evaluation $f(p)$ is required every interation!

Convergence of secant method is much faster than ... but slight slower than Newton's method.

(iv) The Method of False Position: (盈不足术)

(Regula Falsi)

Choose p_0 & p_1 s.t. $f(p_0)f(p_1) \Rightarrow p_2$ - intersection of secant crossing $(p_1, f(p_1))$, $(p_2, f(p_2)) \Rightarrow$ { sgn $f(p_2)\cdot$ sgn $f(p_1) < 0$ use $p_1, p_2 \Rightarrow p_3$; otherwise use $p_0, p_2 \Rightarrow p_3$ }



☆ Convergence Rate:

(i) Suppose $\{p_n\}_{n=1}^{\infty}$ converges to p, with $p_n \neq p$. $\forall n \in \mathbb{N}$. If $\exists$ constants $\lambda > 0$ & $\alpha > 0$ s.t. $\left[\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|^{\alpha}} = \lambda\right]$ generally, $\lambda < 1$

Then $\{p_n\}_{n=1}^{\infty}$ converges to p of order α with asymptotic error constant λ. ($\alpha=1$, linearly; $\alpha=2$, quadratically)

(ii) (Defn) A sequence $\{p_n\}_{n=1}^{\infty}$ is said to be super-linearly convergent to p if $\left[\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|} = 0.\right]$

(Thm) Fixed point iteration converges linearly (obvious to prove)

($\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|} = \lim_{n\to\infty} |g'(\xi)| = |g'(p)|$, linear convergence)

△ Remark: high-order convergence for fixed-point must have $g'(p)=0$. ← condition

(Thm') Let p is a solution s.t. $x=g(x)$. Suppose $g'(p)=0$ & (g'' continuous, $|g''(x)| < M$ on an open interval I) $p \in I$. $\exists \delta > 0$ s.t. $\forall p_0 \in [p-\delta, p+\delta]$ The sequence given by $p_{n+1} = g(p_n)$, $n \in \mathbb{N}$ converges at least quadratically, to p. Moreover, $\exists N \in \mathbb{N}$ s.t. $|p_{n+1}-p| < \frac{M}{2}|p_n-p|^2$ $\forall n \ge N$.

(proof.) Taylor's thm: $p_{n+1} = p + g'(p)(p_n-p) + \frac{g''(\xi_n)}{2!}(p_n-p)^2$.

Thus $\frac{|p_{n+1}-p|}{|p_n-p|^2} \le \frac{|g''(\xi_n)|}{2!} < \frac{M}{2}$ ~~$\forall$ sufficiently large n~~ → $\xi_n \to p$ ($p_n \to p$)

Memo No. 10
Date 2022 / Sep / 22

(Because we can set δ s.t. $g'(x)<1$, $\forall x\in[p-\delta, p+\delta]$ & also $g(x)\in[p-\delta, p+\delta]$.)

Construction: $g(x)=x-\phi(x)f(x)$, with differentiable $\phi(x)$.

need $g'(x)=1-(\phi' f+f'\phi)$. $g'(p)=0=f(p)\Rightarrow \phi(p)=\frac{1}{f'(p)}$.

☆ Newton's!

[Corollary] With $f(p)=0$, $f'(p)\neq 0$ & $f\in C^2$. For starting value sufficiently close to p, Newton's method converges at least quadratically.

(Thm") Secant Methods converge super-linearly!

(proof.) $p_{n+1}=p_n-\frac{f(p_n)(p_n-p_{n-1})}{f(p_n)-f(p_{n-1})}$. Let $e_n=p_n-p$

Then $e_{n+1}=e_n-\frac{f(p_n)(e_n-e_{n-1})}{f(p_n)-f(p_{n-1})}=\frac{e_{n+1}f(p_n)-e_n f(p_{n-1})}{f(p_n)-f(p_{n-1})}$ MVT

$e_{n+1}e_n\left[\frac{f'(\xi_n)-f'(\xi_{n-1})}{f(p_n)-f(p_{n-1})}\right]$ MVT $\frac{e_{n+1}e_n}{e_n-e_{n-1}}\left[\frac{f'(\xi_n)-f'(\xi_{n-1})}{f'(\eta_n)}\right]$ ∠'Höpital Rule

$\frac{f'(\xi_n)-f'(\xi_{n-1})}{e_n-e_{n-1}}\to f''(p)$ & $f'(\eta_n)\to f'(p)$, Thus, $e_{n+1}\propto e_n e_{n-1}$

Suppose $e_{n+1}\propto e_n^p\Rightarrow e_{n+1}\propto e_n^{\frac{1}{p}+1}\Rightarrow p=\frac{1}{p}+1\Rightarrow p=\frac{1+\sqrt{5}}{2}\approx 1.618$.

*2. Solns for Systems:

Bi-section method is hard to apply for systems, thus we generalize Newton's method.

1) For 2x2 systems: Let $F(\vec{x})=\begin{bmatrix} f(x,y)\\ g(x,y)\end{bmatrix}$, $\vec{x}=\begin{bmatrix} x\\ y\end{bmatrix}$.

Jacobian matrix $D_F=\begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y}\\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y}\end{bmatrix}$. update $\vec{x}_{k+1}=\vec{x}_k-D_F^{-1}(\vec{x}_k)F(\vec{x}_k)$

2) For nxn systems: $\vec{x}_{k+1}=\vec{x}_k-D_F^{-1}(\vec{x}_k)F(\vec{x}_k)$.

Need some further techniques for $D_F(\vec{x}_k)\Delta x_k=F(\vec{x}_k)$.

Memo No. 11
Date 2022 / Sep / 27

Another way in proofs of convergence of secant method:

$e_{n+1}=-\left[\frac{p_n-p_{n-1}}{f(p_n)-f(p_{n-1})}\right]\left[\frac{f(p_{n+1})/e_{n+1}-f(p_n)/e_n}{p_n-p_{n-1}}\right]e_n e_{n-1}$ (use Taylor's thm)

simplify

*3. Ways to get complex roots.

1) Müller's Method: (for polynomials with real coefficients, it can converge to a complex root)

(i) Connection with secant method: with $(p_0, f(p_0)), (p_1, f(p_1))$ — linear approximation (secant method); with $(p_0, f(p_0)), (p_1, f(p_1)), (p_2, f(p_2))$ — quadratic approximation (Müller's method)

(ii) Derivation: Lagrange's Interpolation Formula

The quadratic curve: $\sum_{i=0}^{2} f(p_i)\frac{\prod_{j\neq i}(x-p_j)}{\prod_{j\neq i}(p_i-p_j)}$, send it to zero, we get $p_3=r_1, r_2$, we need to choose r_1 or r_2 closer to p_2

(iii) Algorithms: in order to save computation.

write the curve as $a(x-p_2)^2+b(x-p_2)+c$; then (to reduce the round-off error) $p_3=p_2+\frac{-2c}{b+\text{sgn}(b)\sqrt{b^2-4ac}}$

write $a=\frac{\frac{f(p_2)-f(p_1)}{p_2-p_1}-\frac{f(p_1)-f(p_0)}{p_1-p_0}}{(p_2-p_1)+(p_1-p_0)}$ & $b=\frac{f(p_2)-f(p_1)}{p_2-p_1}+(p_2-p_1)a$

☆ Müller's method gives complex roots, for roots of a quadratic curve can be complex.

(iv) Convergence rate: comes from $p^3-p^2-p-1=0$, $p\approx 1.84$

Memo No. 12
Date 2022 / Sep / 27

Summary: slow but reliable: bi-section, fixed point
fast but may NOT converge: Newton's, secant, Müller's

2) Multiple roots:

(i) Drawback of Newton's Method: $f'(p) \neq 0$ must hold.
cannot work for $f(x) = (x-p)^m q(x)$, with multiplicity m ($q(p) \neq 0$)

(ii) Modifications: define $\mu(x) = \frac{f(x)}{f'(x)}$
write $f = (x-p)^m q(x)$, then $\mu(x) = (x-p)\frac{q(x)}{mq(x)+(x-p)q'(x)}$
Thus, $\mu(x)$ has a simple zero at p!
$\Rightarrow$ Do Newton's method for $\mu(x) \Rightarrow g(x) = x - \frac{f(x)\cdot f'(x)}{f'(x)^2 - f(x)f''(x)}$
(☆ Drawback: require f'' calculations!)

(iii) Analysis: If g has the required continuity conditions, fixed-point iteration applied to g will be quadratically convergent
In practice, multiple roots can cause serious round-off error!

[Thm] The function $f \in C^1[a,b]$ has a simple zero at p (on $[a,b]$) iff $f(p)=0$ & $f'(p) \neq 0$.

(proof.) Let $f(x) = (x-p)h(x)$. then $f'(p) = h(p) + (p-p)h'(x) = h(p)$
$h(p) \neq 0 \Leftrightarrow f'(p) \neq 0$.
Conversely, $f(x) \overset{MVT}{=} (x-p)f'(\xi_x)$, $f'(p) = f'(\lim_{x\to p}\xi_x) \overset{\text{continuity}}{=} \lim_{x\to p} f'(\xi_x)$
$\neq 0$. Then, let $g(x) = f'(\xi_x)$ with $g(p) \neq 0$.

[Corollary] multiplicity "m" zero
$\Rightarrow f = f' = \cdots = f^{(m-1)}$ at $p = 0$ but $f^{(m)}(p) \neq 0$

Memo No. 13
Date 2022 / Sep / 29

- Interplotation & Polynomial Approximation

*1. Lagrange interplotating polynomials:

(i) Algebric Polynomials: $P_n(x) = \sum_{i=0}^{n} a_i x^i$ with $n \in \mathbb{N}$ & $a_i \in \mathbb{R}$

(ii) Algebraic polynomials are important:
[Weierstrass Approximation Thm] (see in MAT2006 Notes)
Suppose that $f \in C[a,b]$, $\forall \varepsilon > 0$, $\exists$ a polynomial $P(x)$, with the property that [$|f(x)-P(x)| < \varepsilon$. $\forall x \in [a,b]$.] good approximation
(proof.) See later.

(iii) Degree of n Lagrangian Polynomial:
Addition of simple curves. let $L_{n,m} = \frac{\prod_{i\neq m}(x-x_i)}{\prod_{i\neq m}(x_m-x_i)}$ (crossing $(x_i,0)$ & $(x_m,1)$ $\forall i \neq m$) Thus, the polynomial is attained as $P(x) = \sum_{m=0}^{n} f(x_m)\cdot L_{n,m}(x)$

(iv) Error Analysis:
(Theoretical error bound) Suppose $x_0, x_1, \cdots, x_n$ are distinct numbers in $[a,b]$ & $f \in C^{n+1}[a,b]$. Then, for every $x \in [a,b]$, a number $\xi(x)$ between $x_0, x_1, \cdots, x_n$ in (a,b) exists with

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

(proof.) If $x \in \{x_0, x_1, \cdots, x_n\}$. done. Otherwise, let $g(t) = P(t) - f(t) + [P(x)-f(x)]\prod_{i=0}^{n}\frac{(t-x_i)}{(x-x_i)}$ with fixed x_0. Then $g(x_i) = 0$ & $g(x) = 0$. Thus, by Rolle's thm, $\cdots\cdots \exists \xi(x)$ s.t. $f^{(n+1)}(\xi(x)) = 0$

Since we go from $n+2$ zeros for f to 1 zero for $f^{(n+1)}$, use Rolle's thm $n+1$ times. $\therefore 0 = f^{(n+1)}(\xi(x)) + [p(x)-f(x)](n+1)! \prod_{i=1}^{n} \frac{1}{x-x_i}$ finished. □

☆ Proofs of the "WAT"

Step 1° Approximate the Absolute value Functions

(i) Taylor expansion with Cauchy's Remainder for $\sqrt{1-x}$ around $x=0$ is $\sqrt{1-x} = 1 - \frac{1}{2}x + \sum_{n=2}^{N} \frac{-(2n-3)!!}{(2n)!!} x^n + \boxed{\frac{f^{(N+1)}(c)}{N!}(x-c)^N \cdot x}$

Take $q(x)$ as above, we get on $[-1,1]$ (with x replaced by $1-x^2$)

$$|\,|x| - q(x)\,| = \left|\sqrt{1-(1-x^2)} - q(x)\right| = \left|\frac{f^{(N+1)}(c)}{N!}(1-x^2-c)^N(1-x^2)\right|$$

where $c \in [0, 1-x^2]$. RHS $= \left|\frac{(2N-3)!!}{(2N)!!} \cdot (1-c)^{-\frac{2N+1}{2}} (1-x^2-c)^N (1-x^2)\right| \overset{\text{def}}{=} |E_N(x)|$

$\lim_{N\to\infty} \left|\frac{E_{N+1}(x)}{E_N(x)}\right| = \lim_{N\to\infty} \left|\frac{2N-1}{2N+2} \cdot \frac{(1-x^2-c)}{(1-c)}\right| = 1 - \frac{x^2}{1-c} < 1$, $\forall x \in [-1,1]$ (ratio test) which means $\sum_{N=1}^{\infty} E_N(x)$ converges on $[-1,0)\cup(0,1]$, thus $E_N(x) \xrightarrow{N\to\infty} 0$

when $x=0$, the question becomes the convergence of $\sqrt{1-x} = \sum_{n=0}^{\infty} a_0 x^n$ (at $x=1$) with a_0 as coefficients above. Cauchy Remainder fails here, but $0 = \sum_{n=0}^{\infty} a_0$ also holds, since $a_K = (-1)^K \binom{\frac{1}{2}}{K}$, $\forall K \in \mathbb{N}$ ($\sum_{K=0}^{\infty} a_K = (1-1)^{\frac{1}{2}} = 0$). Thus $E_N(x) \xrightarrow{N\to\infty} 0$, which means $\exists$ some $M \in \mathbb{N}$ s.t. $|E_M(x)| < \varepsilon$, $\forall x \in [-1,1]$. Then, we can find such $q(x)$.

(ii) To generalize $|\,|x| - q(x)\,| < \varepsilon$ from $[-1,1]$ to $[a,b]$, we have $\exists \hat{q}(x)$ s.t. $\left|\left|\frac{x-a}{b-a}\right| - \hat{q}\left(\frac{x-a}{b-a}\right)\right| < \frac{\varepsilon}{b-a}$. since $t = \frac{x-a}{b-a} \in [-1,1]$

Let $h(x) = |b-a| \cdot \hat{q}\left(\frac{x-a}{b-a}\right)$ still a polynomial, thus $|\,|x-a| - h(x)\,| < \varepsilon$.

Do translation. $|\,|x| - h(x+a)\,| = |\,|x| - q(x)\,| < \varepsilon$, $\forall x \in [a,b]$

Step 2° WAT is true on $[-1,1]$. for polygonals! (polygonal)

(i) Consider $h_a(x) = \frac{1}{2}(|x-a| + (x-a))$, with fixed $a \in [-1,1]$

$h_a(x)$ can be uniformly approximated on $[-1,1]$ since $\exists \tilde{q}_n(x)$ s.t. $|\,|x-a| - \tilde{q}_n(x)\,| < \frac{1}{n}$, $\forall x \in [-1,1]$ (from above). Then, let $q_n(x) = \frac{1}{2}[(x-a) + \tilde{q}_n(x)]$, we have $|h_a(x) - q_n(x)| = \frac{1}{2}|\,|x-a| - \tilde{q}_n(x)\,| < \frac{1}{2n}$. Thus, $\{q_n(x)\}_{n=1}^{\infty} \to h_a(x)$ uniformly. (uniform approximation)

(ii) Generalize to any polygonal ϕ: linear on every subinterval of $-1 = a_0 < a_1 < \cdots < a_n = 1$. Consider $\phi(x) - \phi(-1)$ on $[a_0, a_1]$, it must be a linear function with zero a_0 & slope $b_0 \in \mathbb{R}$. then $b_0 \cdot h_{a_0}(x)$ can express it wonderfully. Next step consider $\phi(x) - \phi(-1) - b_0 h_{a_0}(x)$ Then, zero becomes a_1, on $[a_1, a_2]$, similarly we let $b_1 h_{a_1}(x)$ be this function on $[a_1, a_2]$. Sequentially do this, we have. [$\phi(x) = \phi(-1) + \sum_{j=0}^{n-1} b_j h_{a_j}(x)$.] ← general polygonal. Then without any doubt we can find a polynomial $q(x) = \sum_{j=0}^{n-1} q_j(x) + \phi(-1)$, making $|\phi(x) - q(x)| < \varepsilon$, $\forall x \in [-1,1]$.

Step 3° Approximate uniformly continuous function by polygonals

Since $f \in C[a,b]$, f is uniformly continuous on $[a,b]$. Fix $\varepsilon > 0$ $\exists \delta > 0$ s.t. $|f(x) - f(y)| < \varepsilon$. $\forall |x-y| < \delta$. Then do partitions of $[a,b]$ $a_0 = a < a_1 < \cdots < a_N = b$, with $a_{i+1} - a_i \le \delta/2$, $\forall i \in \{0, \ldots N-1\}$.

Memo No. 16
Date 2022 / Sep / 30

Let $\phi(x) = \frac{f(a_{i+1})-f(a_i)}{a_{i+1}-a_i}(x-a_i)$, $\forall x\in[a_i,a_{i+1}]$, $i\in\{0,1,\cdots,N-1\}$.
Then $|f(x)-\phi(x)| = |f(x)-[\lambda f(a_i)+(1-\lambda)f(a_{i+1})| < \varepsilon$, $\forall x\in[a_i,a_{i+1}]$, $i\in\{0,\cdots,N-1\}$. Thus, $\forall x\in[a,b]$, $|f(x)-\phi(x)|<\varepsilon$.

Step 4° Proof of WAT.

From above, let $f(x)\in C[-1,1]$, then $\exists$ such $\phi(x)=\phi_\varepsilon(x)$ (fix $\varepsilon>0$) s.t. $|f(x)-\phi_\varepsilon(x)| < \frac{\varepsilon}{2}$, $\forall x\in[-1,1]$. Since $\phi_\varepsilon(x)$ is a polygonal, thus satisfying WAT, $\exists\, q(x)\in\mathbb{R}[x]$ (polynomials) s.t. $|\phi_\varepsilon(x)-q(x)|<\frac{\varepsilon}{2}$. $\forall x\in[-1,1] \Rightarrow |f(x)-q(x)|<\varepsilon$. $\forall x\in[-1,1]$. Then, go to $[a,b]$ like 1°. •□

(v) Dangerous Error Bound: "Runge's Phenomenon" $f(x)=(1+25x^2)^{-1}$

we can find that $\lim_{n\to\infty}\left(\max_{-1\le x\le 1}|f(x)-P_n(x)|\right)=\infty$ (worse & worse performance as n increases.)

Other: Gibb's phenomenon in Fourier analysis

*2. Improvements of Lagrangian Methods.

(i) Drawbacks of Lagrange Interpolation:

Difficult to apply the error term (NOT known degree n until (for desired accuracy) computations done). Waste of calculations (repeat)

(ii) Let f be defined at $x_0,\cdots,x_n$, & suppose $m_1,\cdots,m_K$ are K distinct integers, with $0\le m_i\le n$. $\forall i$.

Denote (save, in real-word computations) the Lagrangian agreeing with f(x) at k points $x_{m_1},\cdots,x_{m_K}$ by $P_{m_1,\cdots,m_K}(x)$

Then, recursively $P_{0,\cdots,n}(x) = \frac{(x-x_j)P^{(x)}_{0,\cdots,j-1,j+1,\cdots,n} - (x-x_i)\cdot P^{(x)}_{0,\cdots,i-1,i+1,\cdots,n}}{x_i-x_j}$ ☆

with $i\neq j\in\{0,\cdots,n\}$.

(Since $P^{(x)}_{0,\cdots,n}$ passes all $(x_i, f(x_i))$, $i=0,\cdots,n$, easily verified)

(iii) Neville's Method:

Recursion can be applied for finding P_n with table

x_0	P_0			
x_1	P_1	P_{01}		
⋮	⋮	⋮		
x_n	P_n	$P_{n-1,n}$	⋯	$P_{0,\cdots,n}$

"Lower Triangular"

Const/data → linear → quad → ⋯ Derivation with table $O(n^2)$

P_0, P_1, P_2 → P_{01}, P_{12} → P_{012} ⋯

$$Q_{i,j} = \frac{(x-x_{i-j})Q_{i,j-1} - (x-x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}, \quad \text{for } i=1,\cdots,n \ \&\ j=1,\cdots,i.$$

P in (i,j) position of the form/table; iterated

☆ Stopping Criterion: $|Q_{i,i}-Q_{i-1,i-1}|\le\varepsilon$ ($Q_{i,i}$: estimate of f if TRUE)

(iv) A new algebric representation:

Suppose $P_n(x)$ is the n^{th}-Lagrangian polynomial with $x_0,\cdots,x_n$

The divide differences of f w.r.t. $\{x_i\}_{i=0}^n$ is of the form

$$P_n(x) = a_0 + a_1(x-x_1) + \cdots + a_n(x-x_1)\cdots(x-x_n)$$

☆ Aitken's Δ^2 Notation for divided-difference

$f[x_i] := f(x_i)$ & $f[x_{i_1},\cdots,x_{i_n}] = \frac{f[x_{i_2},\cdots,x_{i_n}] - f[x_{i_1},\cdots,x_{i_{n-1}}]}{x_{i_n}-x_{i_1}}$

$\forall i_1<\cdots<i_n\in\mathbb{N}$, $n\in\mathbb{N}$.

x	First divided-diff.	Second divided-diff.
x_0		
x_1	$f[x_0,x_1]=\frac{f[x_1]-f[x_0]}{x_1-x_0}$	$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_1}$
x_2	$f[x_1,x_2]=\frac{f[x_2]-f[x_1]}{x_2-x_1}$	
⋮	⋮	⋮

(Thm) $P_n(x)$ can be written as $P_n(x)=f(x_0)+\sum_{k=1}^{n} f[x_0,\dots,x_k]\prod_{i=1}^{k-1}(x-x_i)$

(proof.) By defn, $f[x,x_0,\dots,x_{k+1}]=f[x_0,\dots,x_k]+f[x,x_0,\dots,x_k](x-x_k)$ $\forall k\in\{1,\dots,n\}$ & $f(x)=f(x_0)+f[x,x_0]$. Thus $f(x)=P_n(x)+R_n(x)$ where $R_n(x)=f[x,x_0,\dots,x_n]\prod_{i=1}^{n}(x-x_i)$. Then construct $g(t)=f(t)-P_n(t)-[f(x)-P_n(x)]\frac{\pi(t-x_i)}{\pi(x-x_i)}$, $g^{(n+1)}(\xi(x))=0$ (Rolle's, let $f\in C^{(n+1)}$)
$\Rightarrow f[x,\dots,x_n]=\frac{f^{(n+1)}(\xi(x))}{(n+1)!}$. □

*3. Hermite Interpolation (Osculating Polynomials)

Given $x_0,\dots,x_n$ distinct in $[a,b]$ & $m_0,m_1,\dots,m_n\in\mathbb{Z}^+\cup\{0\}$. $m=\max_{i=0}^{n}\{m_i\}$. The Osculating polynomials approximate $f\in C^m[a,b]$ for $\{x_i\}_{i=0}^{n}$, and has the least degree with the same values as f and [all its derivatives] up to m_i at x_i, $\forall i$.

(i) [Defn] Let $x_0,\dots,x_n$, $m_0,\dots,m_n$ & m be given as above. Suppose $f\in C^m[a,b]$, the osculating polynomial approximating f is $p(x)$ of least degree s.t. $\frac{d^k p}{dx^k}(x_i)=\frac{d^k f}{dx^k}(x_i)$, $\forall i=0,\dots,n$ & $k=0,\dots,m_i$.

Note that the degree of P is at most $\sum_{i=0}^{n} m_i+n$

(Remark.) The osculating poly. is the m_0^{th} Taylor poly. for f at x_0 & the n^{th} Lagrangian poly. interpolating f on $x_0,\dots,x_n$ when $m_i=0$, $\forall i>0$. (Generalization of the two)

(ii) Explicit Form for Hermite Polynomials

when $m_i=1$, $\forall i\in\{0,1,2,\dots,n\}$, the osculating polynomials become Hermite polynomials.

(Thm) Let $f\in C^1[a,b]$, $x_0,\dots,x_n\in[a,b]$ are distinct. The unique polynomial of least degree agreeing with f & f' at $x_0,\dots,x_n$ is the Hermite polynomial of degree at most $2n+1$ given by $H_{2n+1}(x)=\sum_{j=0}^{n} f(x_j)H_{n,j}(x)+\sum_{j=0}^{n} f'(x_j)\hat{H}_{n,j}(x)$

where $H_{n,j}=[1-2(x-x_j)L'_{n,j}(x_j)]L^2_{n,j}(x)$ & $\hat{H}_{n,j}=(x-x_j)L^2_{n,j}(x)$ ($L_{n,j}=\frac{\prod_{i\neq j}(x-x_i)}{\prod_{i\neq j}(x_j-x_i)}$)

Moreover, if $f\in C^{2n+2}[a,b]$, $f(x)=H_{2n+1}(x)+\frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\left(\prod_{i=0}^{n}(x-x_i)\right)^2$ For some $\xi(x)\in[a,b]$.

(proof.) Firstly $H_{2n+1}(x_k)=f(x_k)$, & $H'_{2n+1}(x_k)=f'(x_k)$, $\forall k\in\{0,\dots,n\}$

Suppose $f\in C^{2n+2}[a,b]$, then let $g(t)=f(t)-H_{2n+1}(t)-[f(x)-H_{2n+1}(x)]\cdot\left[\frac{\pi(t-x_i)}{\pi(x-x_i)}\right]^2$, $g'(t)$ has $n+n+1=2n+1$ number of zeros. (n: Rolle's; $n+1$: $x_0,\dots,x_n$)

Thus, continuously using Rolle's thm, $g^{(2n+2)}(\xi(x))=0$ ($\exists\xi(x)\in[a,b]\Rightarrow\dots$

Memo No. 20
Date 2022 / Oct / 13

(iii) Determine Hermite polynomials by Newton divided-difference formula: setting $z_{2i+1} = z_{2i} = x_i$, $\forall i \in \{0, \cdots, n\}$, with

$$f[z_{2i+1}] = f[z_{2i}] = f(x_i) \quad \& \quad [f[z_{2i}, z_{2i+1}] = f'(x_i)]$$

(z, $f(z)$; 1st diff.; 2nd-diff.)

$z_0 = x_0$ $f[z_0] = f(x_0)$ ☆ $f[z_0, z_1] = f'(x_0)$

$z_1 = x_0$ $f[z_1] = f(x_0)$ $f[z_1, z_2] = \frac{f[z_2] - f[z_1]}{z_2 - z_1}$ $f[z_0, z_1, z_2] = \frac{f[z_1, z_2] - f[z_0, z_1]}{z_2 - z_1}$

$z_2 = x_1$ $f[z_2] = f(x_1)$ ⋮ ⋮ ⋮

Then, $H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{n} f[z_0, \cdots, z_k] \prod_{i=0}^{k-1} (x - z_i)$

☆ Interpolation Algorithm:

$z_{2i+1} = x_i$, $Q_{2i,0} = f(x_i)$, $Q_{2i+1,0} = f(x_i)$, $Q_{2i+1,1} = f'(x_i)$

$i \neq 0$: $Q_{2i,1} = \frac{Q_{2i,0} - Q_{2i-1,0}}{z_{2i} - z_{2i-1}}$ & $i = 2, \cdots, 2n+1$, $j = 2, \cdots, i$ $Q_{ij} = \frac{Q_{i,j-1} - Q_{i-1,j-1}}{z_i - z_{i-j}}$

* 4. Cubic-Spline Interpolation

(i) locally, (piece-wise polynomials) approximation

cubic-spline: a special (order 3) piece-wise polynomial approx. (containing "derivatives" information)

(ii) Requirements: cross points $x_0, x_1, x_2, \cdots, x_n$ & $S_1'(x_1) = S_2'(x_1)$, $\cdots$, $S_{n-1}'(x_{n-1}) = S_n'(x_{n-1})$ Totally $2n + n - 1 = 3n - 1$ equations but unknown coefficients $\geq 3n$. (for $\geq$ quadratic poly.s)

Boundary conditions: $S_1'(x_0)$ & $S_n'(x_{n-1})$ (need at least 2 for symmetry.) — Quadratic NOT good enough.

Memo No. 21
Date 2022 / Oct / 18

(iii) Quadratic: similarly with S & S' conditions. with additional S'' conditions (end-points). Totally, $2n + 2(n-1) = 4n-2$ equations with 2 more boundary conditions $\Rightarrow$ $4n$ unknowns (symmetric)

{ choice I natural / free boundary $S_1''(x_0) = S_n''(x_n) = 0$

choice II clamped boundary $S_1'(x_0) = f'(x_0)$ & $S_n'(x_n) = f'(x_n)$

(iv) Construction: (cubic spline)

$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$. $j = 0, 2, \cdots, n-1$

&
$$\begin{cases} S_j(x_j) = f(x_j),\ S_j(x_{j+1}) = f(x_{j+1}),\ j = 0, \cdots, n-1 & (2n \text{ eqns}) \\ S_j'(x_{j+1}) = S_{j+1}'(x_{j+1}),\ j = 0, \cdots, n-2 & (n-1 \text{ eqns}) \\ S_j''(x_{j+1}) = S_{j+1}''(x_{j+1}),\ j = 0, \cdots, n-2 & (n-1 \text{ eqns}) \\ \text{Boundary conditions} & (2 \text{ eqns}) \end{cases}$$

Procedures: get a_j $\Rightarrow$ $a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$; $j = 0, \cdots, n-1$

$\Rightarrow b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2$ (back) $\Rightarrow d_j = \frac{c_{j+1} - c_j}{3h_j}$ (back); $j = 0, \cdots, n-2, n-1$

$(c_n = c_0 = 0) \to$ B.C. $(j = 0, \cdots, n-2)$ with $h_j = x_{j+1} - x_j$, $(a_n = f(x_n))$

$\therefore \frac{1}{3}h_{j-1}c_{j-1} + \frac{2}{3}(h_{j-1} + h_j)c_j + \frac{1}{3}h_j c_{j+1} = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{1}{h_{j-1}}(a_j - a_{j-1})$ for $j = 1, 2, \cdots, n-1$

Let $\alpha_i = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$, we get

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & \cdots \\ 0 & h_1 & 2(h_1+h_2) & h_2 \\ \vdots & \ddots & \ddots & \ddots \\ 0 & \cdots & h_{n-2} & 2(h_{n-2}+h_{n-1}) \\ & & & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \\ \alpha_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

(construction) $\alpha_0 = c_0 = 0 = \alpha_n = c_n$

(Invoke algorithms for solving tridiagonal linear systems!)

$|a_{ii}| \geq \sum_{i \neq j} |a_{ij}|$ → Diagonally Dominant matrix (invertible)

↳ (proof. $Ax \neq 0$ $\forall x \neq 0$)

Memo No. 22
Date 2022 / Oct / 20

For clamped boundary condition:

(thm) If f is defined on $a=x_0<x_1<\cdots<x_n=b$ & differentiable at a & b. Then, f has a unique clamped spline interpolation on S with nodes $x_0, x_1, \cdots, x_n$.

(proof.) Boundary conditions: $2h_0C_0+h_0C_1=\frac{3}{h_0}(a_1-a_0)-3f'(a)\triangleq\alpha_0$

Others $\frac{1}{3}h_{j-1}C_{j-1}+\frac{2}{3}(h_{j-1}+h_j)C_j+\frac{1}{3}h_jC_j=\frac{1}{3}\alpha_j$, $j=1,\cdots,n-1$.

Define $C_n=C_{n-1}+3h_{n-1}d_{n-1}$, $2C_nh_{n-1}+C_{n-1}h_{n-1}=3f'(b)-3\frac{a_n-a_{n-1}}{h_{n-1}}\triangleq\alpha_n$

Similarly $\begin{bmatrix} 2h_0 & h_0 & 0 & \cdots & 0 & 0 \\ h_0 & 2h_0+2h_1 & h_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & & \\ 0 & 0 & \cdots & h_{n-1} & 2h_{n-1} & \end{bmatrix}\begin{bmatrix} C_0 \\ \vdots \\ C_n \end{bmatrix}=\begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_n \end{bmatrix}\in\mathbb{R}^{n+1}$.

Diagonally Dominant Matrix (invertible) $\Rightarrow$ uniqueness. □

- Numerical Differentiations & Integrations.

*1. Numerical Differentiations:

(i) Recall $f'(x_0)=\lim_{h\to0}\frac{f(x_0+h)-f(x_0)}{h}$; we can use $\left(\frac{f(x_0+h)-f(x_0)}{h}\right)$ (*) to estimate $f'(x_0)$ numerically. (derive: { Taylor's thm / Lagrange poly. with x_0, x_0+h }

Remark: { $h>0$, (*) forward difference formula / $h<0$, (*) backward difference formula }

(ii) General derivative approximation formulas with $n+1$ points

Suppose $x_0,\cdots,x_n$ are distinct $n+1$ points on $[a,b]$ containing x_j, $j\in\{0,\cdots,n\}$. $f\in C^{n+1}[a,b]$. then $f(x)=P_{n+1}(x)+\frac{\pi(x-x_i)}{(n+1)!}f^{(n+1)}(\xi(x))$

Memo No. 23
Date 2022 / Oct / 25

while $P_{n+1}(x)$ is the Lagrangian polynomials with $x_0,\cdots,x_n$

Then $f'(x)=\sum_{j=0}^{n+1}f(x_j)L'_{n,j}(x)+\sum_{k=0}^{n+1}\left[\frac{\prod_{i\neq k}(x-x_i)}{(n+1)!}f^{(n+1)}(\xi(x))\right]+\frac{\pi(x-x_i)}{(n+1)!}\frac{d}{dx}f^{(n+1)}(\xi(x))$

Take $x=x_j$ & we get $f'(x_j)=\sum_{k=0}^{n+1}f(x_k)L'_{n,k}(x_j)+\frac{\prod_{i\neq j}(x-x_i)}{(n+1)!}f^{(n+1)}(\xi(x_j))$

(Remark: by Taylor's thm, also can get the formula) more points, less error!

Special cases: (differentiation formula)

Three points x_0, x_1, x_2

x_0, x_0+h, x_0+2h (end-point)
$f'(x_0)=\frac{1}{2h}(-3f(x_0)+4f(x_0+h)-f(x_0+2h))+\frac{h^2}{3}f^{(3)}(\xi_0)$

x_0-h, x_0, x_0+h (mid-point)
$f'(x_0)=\frac{1}{2h}(f(x_0+h)-f(x_0-h))-\frac{h^2}{6}f^{(3)}(\xi_1)$

2nd-order error

(iii) High-order derivatives & their approximation formulas

For 2nd-order: Taylor's thm $\Rightarrow f(x_0+h)=f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+O(h^3)$

& $f(x_0-h)=f(x_0)-hf'(x_0)+\frac{h^2}{2}f''(x_0)$. Thus, $\left[\frac{1}{h^2}[f(x_0+h)+f(x_0-h)-2f(x_0)]\right]+\frac{1}{h^2}O(h^3)=f''(x_0)$ ← estimation

(Note that $f''(x_0)\overset{\text{forward}}{=}\lim_{h\to\infty}\frac{f'(x_0+h)-f'(x_0)}{h}\overset{\text{backward}}{=}\lim_{h\to\infty}\frac{\frac{f(x_0+h)-f(x_0)}{h}-\frac{f(x_0)-f(x_0-h)}{h}}{h}$

$=\lim_{h\to\infty}\frac{f(x_0+h)+f(x_0-h)-2f(x_0)}{h^2}$)

Richardson's Extrapolation (for higher orders)

can be applied whenever an approximation technique is known to have an error term with a predictable form. (e.g. $O(h)$, $O(h^2)$, ⋯)

Suppose $\exists N_1(h)$ ← step-size s.t. the truncation error involved has the form $M-N_1(h)=\sum_{i=1}^{\infty}k_ih^i\approx k_1h$.

Memo No. 24
Date 2022 / Oct / 27

The objective of extrapolation is to combine these rather inaccurate approximations ⇒ formula with a higher-order trunc. errors.
(e.g. $O(h)$ to $O(h^2)$, construct $N_2(h) = 2N_1(\frac{h}{2}) - N_1(h)$.)
When truncation error has the form $\sum_{j=1}^{m-1} k_j h^{\alpha_j} + O(h^{\alpha_m})$ ($O(h^{\alpha_m})$ error!)
with $\alpha_1 < \alpha_2 < \cdots < \alpha_m$ known (to get $O(h^{\alpha_m})$ error), it can be applied.
(3-point & 5-point methods from extra-polation.)

▲ Additional Materials: use Bernstein function: $\sum_{i=0}^{n} \binom{n}{i} x^i (1-x)^{n-i} f(\frac{i}{n})$ to prove WAT
Last time we use polygonals to estimate uniformly cts functions
& then use polynomials to estimate polygonals.
Here, a concise proof is given, with Bernstein function. (for $x \in [0,1]$)

$$\left| \sum_{i=0}^{n} \binom{n}{i} x^i (1-x)^{n-i} f(\tfrac{i}{n}) - f(x) \right| \le \sum_{i=0}^{n} \binom{n}{i} x^i (1-x)^{n-i} \left| f(\tfrac{i}{n}) - f(x) \right|$$

$$(\le)^{\text{2 parts}} \sum_{i:\left|\frac{i}{n}-x\right|<\delta} \binom{n}{i} x^i (1-x)^{n-i} \left| f(\tfrac{i}{n}) - f(x) \right| + \sum_{i:\left|\frac{i}{n}-x\right|\ge n^{-\frac{1}{3}}} \binom{n}{i} x^i (1-x)^{n-i} \left| f(\tfrac{i}{n}) - f(x) \right|$$

For the 1st term, with uniformly continuous (compact set) $< \frac{\varepsilon}{2}$
For the 2nd term, Chebyshev Inequality comes to rescure.

$$2^{\text{nd}} \text{ term} = \sum_{i:|i-nx|\ge n^{\frac{2}{3}}} P(X=i) \left| f(\tfrac{i}{n}) - f(x) \right| \le 2M \, P(|X-nx| \ge n^{\frac{2}{3}})$$

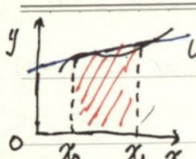
Chebyshev's $\le n^{-\frac{4}{3}} \cdot n x(1-x) \cdot 2M \le \frac{M}{2} n^{-\frac{1}{3}} < \frac{\varepsilon}{2}$, with overlapping:
Let $n \ge \max\{(\frac{M}{\varepsilon})^3, (\frac{1}{\delta})^3\}$, while $|f(x) - f(y)| < \frac{\varepsilon}{2}$ if $|x-y| < \delta$. ∎
(Generalize to $[a,b]$)

Memo No. 25
Date 2022 / Oct / 28

Taylor's series & its remainders: (w.l.o.g.) → Maclaurin ($x_0 = 0$)
(i) Peano's Form: $E_N(x) = o(x^{N+1})$
(ii) Lagrangian Form: $E_N(x) = \frac{f^{(N+1)}(\xi)}{(N+1)!} x^{N+1}$, where ξ between 0 & x.
(iii) Cauchy Form: $E_N(x) = \frac{f^{(N+1)}(\xi)}{N!}(x-\xi)^N x$, where ξ is between 0 & x.
(iv) Integral Form: $E_N(x) = \int_0^x \frac{f^{(N+1)}(t)}{N!}(x-t)^N dt$
(proof.) From $\int_a^b uv^{(n+1)} dx = [uv^{(n)} + (-1)u'v^{(n+1)} + \cdots + u^{(n)}v(-1)^n]_a^b + (-1)^{n+1}\int_a^b u^{(n+1)} v \, dx$
let $u = f(t)$, $v = \frac{(x-t)^N}{N!}$, $n = N$, we get Integral form (remove $(-1)^{n+1}$) $\Rightarrow E_N(x) = \int_0^x f^{(N+1)}(t) \frac{(x-t)^N}{N!} dt$.
From integral form ⇒ Lagrangian form: by MVT $E_N(x) = \frac{f^{(N+1)}(\xi)}{N!}\int_0^x \cdots dt$
From integral form ⇒ Cauchy form: by MVT $E_N(x) = \frac{f^{(N+1)}(\xi)}{N!}(x-\xi)^N \int_0^x 1 \, dt$
Another approach: $F(t) \triangleq f(x) - P_n(t;x)$, with $F'(t) = -\frac{f^{(n+1)}(t)}{n!}(x-t)^n$
Cachy-MVT ⇒ $F(x) - F(0) = \frac{F'(\xi)}{\phi'(\xi)}[\phi(x) - \phi(0)]$, for some $\phi \in C'[0,x]$.
$-E_N(x) = LHS = RHS = -\frac{f^{(N+1)}(\xi)}{\phi'(\xi) N!}(x-\xi)^N [\phi(x) - \phi(0)]$. Take $\phi(x) = x$ ⇒ Cachy form, while taking $\phi(t) = (x-t)^{N+1}$. ∎

*2. Numerical Integrations.
(i) Idea: to approximate $\int_a^b f(x)dx$, use Lagrangian Polynomials
$\int_a^b f(x)dx \cong \sum_i f(x_i) \int_a^b L_i(x) dx = \sum_i a_i f(x_i)$ Quadrature formula
(ii) The Trapezoidal & Simpson's Rule:

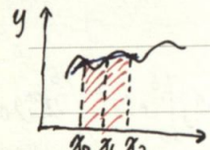
Memo No. 26
Date 2022 / Nov / 1



use the area of trapezoidal to estimate integrations with 1st-order Lagrangian polynomials

$$\int_a^b f(x)dx = \frac{b-a}{2}(f(a)+f(b)) - \frac{1}{12}(b-a)^3 f''(\xi)$$ (error term 2nd order)

(Derivation: Error term in Lagrangian polynomial, MVT in integral form & do integrations for the polynomials.)



use the area under a qudratic line to estimate integrations with 2nd-order Lagrangian polynomials (Simpson's rule)

$$\int_a^b f(x)dx = \frac{b-a}{6}[f(a)+4f(c)+f(b)] - \frac{(b-a)^5}{2880}f^{(4)}(\xi)$$ (error term 4th order)

"$c=\frac{1}{2}(a+b)$"

(Same derivation approach, or use Taylor's thm at $c=\frac{a+b}{2}$.)

(R.m.k. Generalize to n+1 points case, $\{x_0+kh\}_{k=0}^{n}$, $a=x_0$ & $b=x_0+nh$

Trapezoidal rule: $\int_a^b f(x)dx = \frac{h}{2}[(y_0+y_n)+2\sum_{j=1}^{n-1}y_j] - \frac{1}{12}(b-a)h^2f''(\xi)$

Simpson's rule: $\int_a^b f(x)dx = \frac{h}{3}[(y_0+y_n)+4(y_1+y_3+\cdots+y_{n-1})+2(y_2+\cdots+y_{n-2})] - \frac{f^{(4)}(\xi)}{180}(b-a)h^4$, when n is even.)

(iii) Measuring Precision: the standard derivation of quadrature error formula is based on determining the class of polynomials for which these formulas produce exact solutions.

The degree of accuracy / precision: largest pos. n that error is 0 for x^n

(e.g. Simpson's rule: deg = 3, trapezoidal: deg = 1)

Memo No. 27
Date 2022 / Nov / 1

(iv) Newton–Cotes Formula:

① (n+1)-pts closed formula: $\int_a^b f(x)dx \cong \sum_{i=0}^{n} a_i f(x_i)$, where

$x_i = x_0+ih$, $h=\frac{b-a}{n}$ & $a_i = \int_a^b L_{n,i}(x)dx$

(Thm) Error terms: if $\sum_{i=0}^{n} a_i f(x_i)$ denotes (n+1)-pt closed Newton-Cotes formula, $\exists\ \xi\in(a,b)$ s.t. $\int_a^b f\,dx = \sum_i a_i f(x_i) + \frac{h^{n+3}f^{(n+2)}(\xi)}{(n+2)!}\int_0^n t^2(t-1)\cdots(t-n)\,dt$

when n is even & $f\in C^{n+2}[a,b]$ (due to "symmetry") $\int_a^b f\,dx = \sum_i a_i f(x_i) + \frac{h^{n+2}f^{(n+1)}(\xi)}{(n+2)!}\int_0^n t(t-1)\cdots(t-n)\,dt$,

when n is odd & $f\in C^{n+1}[a,b]$

② (n+1)-pts open formula (excluding end points)

$x_i = x_0+ih$, with $x_{-1}=a$, $x_{n+1}=b$ & $x_0=a+h$, $x_n=b-h$, $h=\frac{b-a}{n+2}$

Still $\int_a^b \cong \sum_{i=0}^{n}$, with $a_i=\int_a^b L_{n,i}\,dx$ without x_{-1}, x_{n+1}.

(Thm) Error terms: if $\sum_{i=0}^{n} a_i f(x_i)$ denotes (n+1)-pt open Newton-Cotes formula, $\exists\ \xi\in(a,b)$ s.t. $\int_a^b f\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3}f^{(n+2)}(\xi)}{(n+2)!}\int_{-1}^{n+1} t^2(t-1)\cdots(t-n)\,dt$

when n is even & $f\in C^{n+2}[a,b]$ $\int_a^b f\,dx = \sum_i a_i f(x_i) + \frac{h^{n+2}f^{(n+1)}(\xi)}{(n+2)!}\int_{-1}^{n+1} t(t-1)\cdots(t-n)\,dt$

☆ Special Cases: closed – n=1 (Trapezoidal), n=2 (Simpson)

open – n=1 (mid-point)

(R.m.k Trapezoidal (rule) with [peroid functions] (i.e. $f(a)=f(b)$) can achieve spectual accuracy (error $\sim e^{-kh}$) ← fast (F.F.T. good for peroidic funcs)

n-points mid-point rule: $h=\frac{|a-b|}{n+2}$, $x_j = a+(j+1)h$, $\forall j=-1,\cdots,n+1$

(n is even) $\int_a^b f = 2h\sum_{j=0}^{n/2} f(x_j) + \frac{b-a}{6}h^2f''(\mu)$, $\mu\in(a,b)$

Memo No. 28
Date 2022 / Nov / 3

(v) Romberg Integrations:

(e.g.) Composite Trapezoidal Rule: $\int_a^b f = \frac{h}{2}[\cdots+\cdots+\cdots] + k_1h^2 + k_2h^4 + \cdots$ ($R_{n,1}$; known errors!)

where k_i is a constant depending on $f^{(2i-1)}(a)$ & $f^{(2i-1)}(b)$

Apply Richardson's extrapolation:

$R_{k,j+1} = R_{k,j} + \frac{1}{4^j-1}(R_{k,j} - R_{k-1,j})$, $\forall k = j+1, \cdots$ ($O(h^{2j+2})$ with 2^k points; 2 times more points $\Rightarrow (\frac{h}{2})$ in the error term)

Thus, we get a more precise approximation with error $O(h^{2j+2})$ for trapezoidal Rule.

(Note: recursive formulas for $R_{k,1}$: $R_{k,1} = \frac{1}{2}R_{k-1,1} + \frac{b-a}{2^{k-2}}\sum_{\text{new}} f(\frac{(a+b)i}{2^{k-1}})$ — some new points; reduce repetitive computations)

★ Algorithm:

n	$O(h^2)$	$O(h^4)$	$\cdots$	$O(h^{2n})$
1	R_{11}	—	$\cdots$	—
2	R_{21}	R_{22}	$\cdots$	—
$\vdots$	$\vdots$	$\vdots$		$\vdots$
n	R_{n1}	R_{n2}	$\cdots$	R_{nn}

(vi) Gaussian Quadrature: choose the points (partitions) $x_0, x_1, \cdots, x_n$ in the optimal way rather than uniformly dirstributed way.

(change to optimization) $\max_{x_i, c_i\ (\forall i)} \deg_{\text{pre}}\left[\int_a^b f - \sum_i c_i f(x_i)\right]$ ⟺ maximize degree of precision

→ with $2n$ parameters to choose

→ Note that measuring preuision must hold! "$2n-1$ degree" of precision

★ Legendre Polynomials:

a collection of polynomials $\{P_0, P_1, \cdots\}$ with properties

(i) P_n is a monic polynomial of degree n.

(ii) $\int_{-1}^{1} P(x)P_n(x)dx = 0$, $\forall P \in \mathbb{R}[x]$ with $\deg(P) \le n-1$.

(Note that $\langle P_m, P_n\rangle_{L^2} = 0$, orthogonal basis)

Memo No. 29
Date 2022 / Nov / 8

By Gram-Schmidt, $Y_{n+1} = X_n - \sum_i \langle X_n, P_i\rangle_{L^2} P_i$. $P_{n+1} = \frac{Y_{n+1}}{\|Y_{n+1}\|_{L^2}}$ ← monic ...

$\therefore \forall P_a(x)$ s.t. $\deg(P_a) \le 2n-1$, $\exists Q, R \in \mathbb{R}[x]$. s.t. Q, R has degree less than n, s.t., $P_a(x) = Q(x)P_n(x) + R(x)$ → $\int_{-1}^{1} \cdots = 0$

Choose x_i s.t. $P_n(x_i) = 0$ (root), then $\sum_i c_i Q_n(x_i) P_n(x_i) = 0$, then $\sum_i c_i R(x_i)$ with fixed x_i — only need to choose c_is.

[Theorem] Suppose $x_1, \cdots, x_n$ are the roots of n^{th} Legendre polynomial $P_n(x)$ (n real roots in $(-1,1)$) & $\forall i$, $c_i = \int_{-1}^{1} \prod_{j\ne i} \frac{x-x_j}{x_i-x_j} dx$. Then, if $P(x) \in \mathbb{R}[x]$, $\deg(P) < 2n$, $\int_{-1}^{1} P(x)dx = \sum_i c_i P(x_i)$

(proof.) If $\deg(P) < n$, $P(x) = \sum_i P(x_i) \prod_{j\ne i} \frac{x-x_j}{x_i-x_j}$ is exact.

$\int_{-1}^{1} P = \sum_i P(x_i) \int_{-1}^{1} \prod_{j\ne i} \frac{x-x_j}{x_i-x_j} dx = \sum_i c_i P(x_i)$. When $\deg(P) < 2n$

$\exists Q, R$ as above s.t. $P = QP_n + R$. Then $\int_{-1}^{1} P = \sum_i R(x_i) \int_{-1}^{1} \cdots = \sum_i c_i R(x_i)$

(since $\deg(R) < n$), RHS $= \sum_i c_i P(x_i)$ since $P(x_i) = R(x_i)$. □

(PACKAGE: Chebfun — MATLAB; FFT (trapezoidal rule for periodic func.); NUFFT (Gaussian Quadrature ...).)

Generalized to arbitrary $[a,b]$: $t = \frac{(x-a)+(x-b)}{b-a}$ [$x = \frac{1}{2}(b-a)(t+a+b)$]

Memo No. 30
Date 2022 / Nov / 10

• Numerical Linear Algebra

*1. Solving Linear Systems ($Ax=b$)

(i) When A is a square matrix

"Gaussian Elimination & Backward Substitution": $O(n^3)$

☆ Complexity of basic operations: flop (unit) (floating point operations) i.e. "+", "−", "·", "/" of floating point numbers

we are interested in rough "flop"s.

① "Vec-Vec" operations: $\vec{a}, \vec{b} \in \mathbb{R}^n$ $\begin{cases} \vec{a}+\vec{b} & n \text{ flops} \\ c\cdot\vec{a}\ (c\in\mathbb{R}) & n \text{ flops} \\ \vec{a}^T\vec{b} & 2n \text{ flops} \end{cases}$

② "Mat-Vec" product: $A\in M_{m\times n}(\mathbb{R})$, $b\in\mathbb{R}^n$ In general $2mn$ flops. For special A: permutation — 0 flops; s-sparse — $2s$ flops; k-banded ($M_{n\times n}(\mathbb{R})$) — $2nk$ flops; $\sum_{i=1}^{r} v_i u_i^T$ — $2r(m+n)$ flops

③ "Mat-Mat" product:

$A\in M_{m\times n}(\mathbb{R})$, $B\in M_{n\times k}(\mathbb{R})$ (in general) $2mnk$ flops

s-sparse A: $2sp$ flops (less if B also sparse)

Pseudo Codes: (for Gaussian-Elimination)

$\forall k=1,\cdots,n-1$ { $\forall i=k+1,\cdots,n$: $a_{ik} = \frac{a_{ik}}{a_{kk}}$ (NOT 0!); { $\forall j=k+1,\cdots,n$ $a_{ij} = a_{ij} - a_{ik}a_{kj}$ } } → 3 loops

Pivoting: { Partial ~: $|a_{\mu,k}| = \max_{k\le i\le n} |a_{ik}|$ (the 1st μ) then interchange μ^{th} & k^{th} rows

total ~: $|a_{u,v}| = \max_{k\le i,j\le n} |a_{ij}|$ (the 1st $a_{u,v}$ for several) then interchange the μ^{th}, k^{th} rows & v^{th}, k^{th} columns

Keep track of index vectors !!

Memo No. 31
Date 2022 / Nov / 11

Thus, with forward / backward substitutions ⇒ all b_is.

(ii) "Matrix Factorizations":

Factorize $A = \prod_{i=1}^{k} A_i$, then compute $x = A_k^{-1}A_{k-1}^{-1}\cdots A_1^{-1}b$.

(r.m.k ① Usually $k=2$ or 3; ② Computing factorization is expensive (n^3 flops), ③ Appling $A_1^{-1}\cdots A_k^{-1}$ is cheaper, usually n^2 flops)

↳ othorgonal / triangular / diagonal / permutation .....

"LU" − Factorization → equivalent to "Gaussian Elimination"

↳ pivoting using permutation matrix P: $\begin{cases} A=PLU & \text{for partial } \sim \\ A=P_1LUP_2 & \text{for total } \sim \end{cases}$

(Note that $P^TP=I$, "$P^TAQ^T=LU$")

☆ Measuring "Numerical Effort" ← "time" by flops counting. (in "num-lin-alg.")

(another criterion — only count "·", "/" since amount of time others like "+", "−", loops, e.t.c is roughly prop. ($\propto$) to flops).

Special Structure: positive definite matrices

Cholesky Factorization $A = LL^T$ ← lower triangular

(existence of cholesky factorization)

Induction: $M_{k\times k}(\mathbb{R})$ holds, that is $A_k = L_kL_k^T$. For $M_{k+1\times k+1}(\mathbb{R})$

$A_{k+1} = \left[\begin{array}{c|c} A_k & b_k \\ \hline b_k^T & a_{k+1} \end{array}\right]$ pos. def., construct $L_{k+1} = \left[\begin{array}{c|c} L_k & 0 \\ \hline b_k^TL_k^{-T} & \sqrt{r_k} \end{array}\right]$, L_k^{-T} exists

since $\forall x\in\mathbb{R}^k$, $L_k^Tx\neq 0$ (invertible). ⇒ $r_k = -b_k^TA_k^{-1}b_k + a_{k+1}$ positive

since let $x_{k+1} = \begin{bmatrix} -A_k^{-1}b_k \\ 1 \end{bmatrix}$

⇒ $x_{k+1}^TA_{k+1}x_{k+1} = a_{k+1} - b_k^TA_k^{-1}b_k > 0$.

Memo No. 32
Date 2022 / Nov / 15

"Algorithm" to find L: col 1 – $l_{11}=\sqrt{a_{11}}$, $l_{i1}=\frac{a_{i1}}{\sqrt{a_{11}}}$ $(\forall i\geq 1)$

col 2 – $l_{22}=\sqrt{a_{22}-l_{21}^2}$, $l_{i2}=\frac{a_{i2}-l_{i1}l_{21}}{l_{22}}$ $(\forall i\geq 2)$

Generally: $l_{kk}=\sqrt{a_{kk}-\sum_{i=1}^{k-1}l_{ki}^2}$, $l_{ik}=\frac{a_{ik}-\sum_{j=1}^{k-1}l_{ij}l_{kj}}{l_{kk}}$ $(i\geq k)$

(From left → right, up → bottom) "$\frac{1}{6}n^3+\frac{1}{2}n^2-\frac{2}{3}n$"

★ Numerical Stability. theoretically $|l_{ij}|\leq\sqrt{a_{ii}}$ bdd

(No need to do pivoting!!) numerically stable!

Complete fill-in: Gaussian-elimination change (may) sparse matrix to be dense!

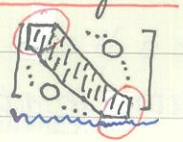
(e.g. $\begin{bmatrix} x&x&x&x&x\\ x&x&0&0&0\\ x&0&x&0&0\\ x&0&0&x&0\\ x&0&0&0&x \end{bmatrix}=A$. but with A^T, fine.) ↖ the tip of large ice-hall.

↑ Solve by pivoting

Banded-matrix:

[Defn] 2 integers $1<p,q<n$ s.t. $a_{ij}=0$, $\forall\, i+j\geq p$ or $j-i\leq -q$

band length $=p+q-1$



(r.m.k special case $p=q=2$ "tridiagonal" matrices) LU factorization: $T=LU$ (tridiagonal; L, U: 2-banded matrix)

"Algorithm": $\begin{cases} a_{i,i-1}=l_{i,i-1} \\ a_{i,i}=l_{i,i-1}\cdot u_{i-1,i}+l_i \\ a_{i,i+1}=l_{i,i}\cdot u_{i,i+1} \end{cases}$ Crout factorization

(numerically solving $u''=f$ at distinct points $u''_i=\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}$ "tridiagonal matrix")

$O(n)$ → tridiagonal matrix f''_i

Product property $\begin{cases} \|Ax\|_p\leq\|A\|_{p,q}\|x\|_q \\ \|AB\|_{p,q}\leq\|A\|_{p,r}\|B\|_{r,q} \end{cases}$

The infinity norm. $\|A\|_\infty=\max_{\|x\|_\infty=1}\|Ax\|_\infty=\max_{i=1,\dots,m}\sum_{j=1}^{m}|a_{ij}|$

(In contrast, $\|A\|_{1,1}=\max_{j=1,\dots,n}\sum_{i=1}^{m}|a_{ij}|$).

Spectral Radius: $\|A\|_{2,2}=\sqrt{\rho(A^TA)}$ ← (spectral radius of square matrix)

$\rho(B):=\max_i|\lambda_i(B)|$ ← maximum of all absolute of ei-values of B

If A is symmetric, $\|A\|_{2,2}=\rho(A)$.

[Thm] $\forall\varepsilon>0$. $\exists$ an induced norm $\|\cdot\|_{p,q}$ s.t. $\|A\|_{p,q}<\rho(A)+\varepsilon$, $\forall$ matrix A & $\rho(A)\leq\|A\|$ ← $\rho(A)$ is the greatest lower bound of all induced norm [$\rho(A)=\inf_{p,q\in\mathbb{Z}^+\cup\{\infty\}}\|A\|_{p,q}$]

(iv) Iterative Methods:

{ Direct methods: Gaussian elimination, "no truncation error" but "expensive computation"
Iterative methods: "effective storage & computation" suitable for "large systems with sparsity"

① Jacobi's Method: $x_i^{(k)}=+\frac{1}{a_{ii}}\left[-\sum_{\substack{j=1\\ j\neq i}}^{n}(a_{ij}x_j^{(k-1)}+b_i)\right]$ $\forall\, k\geq 1$ (k-1: last; k: next)

(when $k=k-1$, exact)

Matrix form: $A=D-L-U$ ← upper with 0 in diagonal; ← lower

If D^{-1} exists $x^{(k)}=\boxed{D^{-1}(L+U)}x^{(k-1)}+\boxed{D^{-1}b}$

T_j for j^{th} row C_j → for j^{th} row

Pseudo Codes: initialize $x^{(0)}$, T, C

while $\frac{\|x^{(k)} - x^{(k-1)}\|_\infty}{\|x^{(k)}\|_\infty} \geq$ tolerance

for $j = 1, \dots, n$ { $x_j^{(k)} = T_j x^{(k-1)} + c_j$ } end

end

Another type: Storage saving

Initialize $x^{(0)}, T, c, y = \mathbb{1}$ (auxiliary)

while $\frac{\|x - y\|_\infty}{\|x\|_\infty} \geq$ tolerance

$y = x$;

for $j = 1, \dots, n$ { $x_j = T_j \vec{x} + c_j$ } end

end

② Improvement: the Guass-Seidel Method

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} (a_{ij} x_j^{(k-1)}) + b_i\right]$$

> use of first several ...

matrix form: $[x^{(k)} = (D-L)^{-1} U x^{(k-1)} + (D-L)^{-1} b]$ ($T_g = (D-L)^{-1}U$, $c_g = (D-L)^{-1}b$)

general iteration method:

$$x^{(k)} = T x^{(k-1)} + c \quad (T = T_j / T_g,\ c = c_j / c_g)$$

[Thm] (Convergence result) $\forall x^{(0)} \in \mathbb{R}^n$, $\{x^{(k)}\}_{k=0}^{\infty}$ defined by $x^{(k)} = T x^{(k-1)} + c$ converges to a unique solution $x = Tx + c$ iff [spectral radius $\rho(T) < 1$]

(proof.) ($\Leftarrow$) $x^{(k)} = T^k x^{(0)} + (\sum_{i=0}^{k-1} T^i) c$, take limit: $x = (I-T)^{-1} c$.

($\Rightarrow$) arbitrary z & x unique s.t. $x = Tx + c$, $x^{(0)} \triangleq x - z$

Since by iteration $x - x^{(k)} = T(x - x^{(k-1)}) = \dots = T^k z$, take limits

LHS = 0, that is $T^k z \xrightarrow{k\to\infty} 0$, $\forall z$. By equivalent conditions below. done.

[Defn] Convergent matrix: if $[\lim_{k\to\infty} (A^k)_{ij} = 0 \quad \forall 1 \leq i, j \leq n]$

(r.m.k. Equivalent conditions of convergent matrix:

① $\lim_{k\to\infty} \|A^k\| = 0$ for some/all induced norms;

② $[\rho(A) < 1]$; ③ $\lim_{k\to\infty} A^k x = 0$, $\forall x \in \mathbb{R}^n$.)

(proof.) convergent matrix $\Rightarrow \|\cdot\|_\infty = 0$ (defn), some $\Rightarrow$ all (trivial)

(equivalence of norms) $\Rightarrow$ ③ (special $x_i = e_i$) ($\|A^k x\| \leq \|A^k\| \|x\|$) $\Rightarrow$ ② (contradiction) ($\rho(A)$ is g.l.b.) (find $\|A\|_{\dots} < 1$)

No λ_i s.t. $\lambda_i \geq 1$. otherwise $\lambda_i^k \to 1$ or ∞. ▥

(Lemma) If $\rho(T) < 1$. $(I-T)^{-1} = I + T + T^2 + \cdots$ converges. (matrix series $\sum_{k=0}^{\infty} T^k$)

(proof.) $(I-T)x = (1-\lambda)x$, i.e., λ is an ei-value of T, iff $1-\lambda$ is an ei-value of $I-T$. & 0 is NOT an ei-value of $I-T$, ($I-T$ invertible)

$S_n = \sum_{k=0}^{n} T^n \Rightarrow S_n = (I-T)^{-1}(I-T^{n+1})$, then take limits. (to $(I-T)S_n$)

[Coro.] Jacobi & Guass-Seidel methods for Strictly diagonally dominant matrix A converges to unique x s.t. $Ax = b$.

(proof.) Check $\rho(T_j)$, $\rho(T_g)$

(V) Relaxation Techniques:

(error bound) $\|x - x^{(k)}\| \leq \frac{\|T^k\|}{1 - \|T\|} \|x^{(1)} - x^{(0)}\|$ (how fast)

construct T: [$\rho(T)$ small enough]

[Defn] (residual vectors) $\tilde{x}$ is approx. soln to $Ax = b$. Then the residual vector $[r := b - A\tilde{x}]$ for r.

$\vec{r}_i^{(k)} \triangleq (r_{i1}^{(k)}; \cdots; r_{in}^{(k)})$

$\vec{x}_i^{(k)} \triangleq (x_1^{(k)}, \cdots, x_{i-1}^{(k)}, x_i^{(k-1)}, \cdots, x_n^{(k-1)})$

residual vector for Guass-Seidel. $x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}$ ⇔ choose $x_i^{(k)}$ in G-S

$x_i^{(k)}$ s.t. $r_{i,i+1}^{(k)} = 0$, (since $r_{i,i+1}^{(k)} = b_i - \sum_{j=1}^{i} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}$)

$x_i^{(k)}$ makes the i^{th} entry of next residual to be zero

"makes $\|\vec{r}_{i+1}^{(k)}\|$ less" ⇓ better

★ relaxation techs: $x_i^{(k)} = x_i^{(k-1)} + \omega \frac{r_{ii}^{(k)}}{a_{ii}}$ { $\omega \in (0,1)$ under-relaxation; $\omega > 1$ over-relaxation

Used to accelerate the convergence of system that are convergent with G-S.

[$\|\vec{r}_{i+1}^{(k)}\| = \|\vec{r}_i^{(k)} - \omega \cdot \frac{r_{ii}^{(k)}}{a_{ii}} \cdot A_i\|$], where A_i means the i^{th} column of A, when $\omega = 1$, $r_{i,i+1}^{(k)} = 0$, directly

varying ω, $\|\vec{r}_{i+1}^{(k)}\|$ can be reduced (more than when $\omega = 1$)

Update method: $x_i^{(k)} = (1-\omega) x_i^{(k-1)} + \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right]$

(matrix form) $\vec{x}^{(k)} = T_\omega \vec{x}^{(k-1)} + \vec{c}_\omega$ { $T_\omega = (D-\omega L)^{-1}[(1-\omega)D + \omega U]$; $c_\omega = \omega (D-\omega L)^{-1} \vec{b}$

(D, L, U defined as previous)

Choose optimal ω:

[Thm] (Kahan) If $a_{ii} \neq 0$ $\forall i = 1, \cdots, n$. $\rho(T_\omega) \geq |\omega - 1|$. (successive over-relaxation)

(Thus, only when $\omega < 2$, SOR method converges.)

(proof.) Since $T_\omega = [D-\omega L]^{-1}[D + \omega(U-D)]$, $\det(T_\omega) = \frac{1}{\det(D-\omega L)} \det(D + \omega(U-D)) = \frac{1}{\det D} (1-\omega)^n \cdot \det D = (1-\omega)^n$.

($\det D \neq 0$ since $a_{ii} \neq 0$ $\forall i$). Then, by pigeonhole principle, $\rho(T_\omega) = \max_{1 \leq i \leq n} |\lambda_i| \geq \sqrt[n]{(1-\omega)^n} = |\omega - 1|$, since $\prod_i \lambda_i = \det(T_\omega) = (1-\omega)^n$. ∎

[Thm'] If A is symmetric pos.def. & $0 < \omega < 2$. The SOR method converges $\forall$ initial point $\vec{x}^{(0)}$. (Ostrowski-Reich) $\Leftarrow A^T = A$

(proof.) Firstly, $A = \frac{1}{\omega}(D - \omega L) - \frac{1}{\omega}[(1-\omega)D + \omega U]$ is positive definite. Thus, $[A - T_\omega^T A T_\omega] = [\omega(D-\omega L)^{-1} A]^T [\frac{1}{\omega}(D-\omega L) + \frac{1}{\omega}(D^T - \omega L^T) - A][\omega(D-\omega L)^{-1} A]$ is positive definite since the middle one is $\frac{2-\omega}{\omega} D$.

Thus, let any ei-vec $\vec{x}$ of $T_\omega \Rightarrow \vec{x} A \vec{x} - \lambda^2 \vec{x} A \vec{x} > 0 \Rightarrow |\lambda| < 1$ $\forall$ ei-value λ of T_ω.

[Thm''] If A is positive definite & tri-diagonal. Then $\rho(T_g) = [\rho(T_j)]^2 < 1$. The optimal choice of ω for SOR is $\omega = \frac{2}{1 + \sqrt{1 - \rho(T_j)^2}}$. With this ω, $\rho(T_\omega) = \omega - 1$.

(proof.) sketch - firstly show A is consistently ordered & 2-cyclic. Under this condition, $\rho(T_j)^2 = \rho(T_1) = \rho(T_g)$

Moreover $\chi_{T_\omega}(\lambda) = (\lambda + \omega - 1)^p \prod_{i=1}^{r} [(\lambda + \omega - 1)^2 - \lambda \omega^2 \mu_i^2]$, while $\chi_{T_j}(\mu) = \mu^p \prod_{i=1}^{r} (\mu^2 - \mu_i^2)$. (relations between ei-values of T_ω & T_j)

Optimal case (draw curves for analysis): $\mu_i^2 \omega^2 - 4\omega + 4 = 0$ ...

(vi) Condition number of a matrix

[Defn] The condition number of an invertible matrix A is (w.r.t. $\|\cdot\|$, [$\kappa(A) := \|A\| \cdot \|A^{-1}\|$] { $\kappa(A)$ closed to 1 : well-conditioned; $\kappa(A)$ far away from 1 ($\kappa(A) \gg 1$) : ill-conditioned

Memo No. 38
Date 2022 / Dec / 1

☆ Estimation of "$\text{cond}^{K(\cdot)}(\cdot)$": LAPACK & MATLAB (implemented) in. (never use A^{-1} to approximate cond(A)). With $\rho(A) \le \|A\| \Rightarrow \text{cond}(A) \ge \frac{\lambda_{max}}{\lambda_{min}} (\ge 1)$ ← sharp

For orthogonal matrix Q ($Q^TQ = I$), $\text{cond}(Q) = 1$ ← as well conditioned as possible

… singular matrix – $\text{cond}(\cdot) \mapsto \infty$ ← numerical difficulties

can never get reliable soln with iterated method ← large relative error

$$\frac{1}{\|A\|\|A^{-1}\|}\frac{\|\vec{r}\|}{\|\vec{b}\|} \le \frac{\|\vec{x}-\tilde{x}\|}{\|\vec{x}\|} \le \|A\|\|A^{-1}\|\frac{\|\vec{r}\|}{\|\vec{b}\|}$$

lower bd: $K(A)^{-1} = \text{cond}(A)^{-1}$ ← relative error ← upper bound: $K(A) = \text{cond}(A)$

The key to handling ill-conditioning is to avoid it.

(e.g. Hilbert matrix $(H)_{ij} = \frac{1}{i+j-1}$. pos-def. ill-conditioned

(can be derived from polynomial estimation of f ($\min_{\alpha_1 \cdots \alpha_n} \|f - \sum_{i=1}^{n} \alpha_i x^{i-1}\|_{L^2}$)

$\Rightarrow$ [$H_n \cdot \vec{\alpha} = \vec{b}$.] Transform to well-conditioned Use Legendre's polynomial

$\alpha_k = \sum_i \frac{\langle f_i, P_i\rangle_{L^2}}{\langle P_i, P_i\rangle_{L^2}} = \frac{\langle f, P_k\rangle_{L^2}}{\langle P_k, P_k\rangle_{L^2}}$)

[Thm] Suppose A is invertible & $\|\delta A\| < \frac{1}{\|A^{-1}\|}$.

The solution $\tilde{x}$ to $(A+\delta A)\tilde{x} = \vec{b}+\delta\vec{b}$ approximates the soln $\vec{x}$ of $A\vec{x} = \vec{b}$ with error bdd: ← pertubation

$$\left[\frac{\|\tilde{x}-\vec{x}\|}{\|\vec{x}\|} \le \frac{K(A)\|A\|}{\|A\| - K(A)\|\delta A\|}\left(\frac{\|\delta\vec{b}\|}{\|\vec{b}\|} + \frac{\|\delta A\|}{\|A\|}\right)\right]$$

(proof.) $\tilde{x} = \vec{x} + \delta\vec{x}$, $\Rightarrow (I + A^{-1}\delta A)(\vec{x}+\delta\vec{x}) = A^{-1}(\vec{b}+\delta\vec{b})$

$\Rightarrow (I + A^{-1}\delta A)\delta\vec{x} = A^{-1}\delta\vec{b} - A^{-1}\delta A\vec{x}$, thus, $\|\delta\vec{x}\| \le \|(I+A^{-1}\delta A)^{-1}\|\|A^{-1}(\delta\vec{b} - \delta A\vec{x})\| \le \|(I+A^{-1}\delta A)^{-1}\|(\|A^{-1}\delta A\vec{x}\| + \|A^{-1}\delta\vec{b}\|) \le \|(I+A^{-1}\delta A)^{-1}\|\frac{K(A)}{\|A\|}(\|\delta A\|\|\vec{x}\| + \|\delta\vec{b}\|) \le \frac{K(A)}{1-\|A^{-1}\|\|\delta A\|}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta\vec{b}\|}{\|\vec{b}\|}\right)\|\vec{x}\|$. Since $\|A^{-1}\delta A\| \le \|A^{-1}\|\|\delta A\| < 1$

$\therefore \|(I+A^{-1}\delta A)^{-1}\| \le (1-\|A^{-1}\|\|\delta A\|)^{-1}$. ($\|(I \pm B)^{-1}\| \le (1-\|B\|)^{-1}$.)

need check!

Memo No. 39
Date 2022 / Dec / 1

*2. Eigen-value Problems

(i) Q-R factorization:

Least-square revisit: { normal eqn $A^TA\vec{x} = \vec{b} \Rightarrow A\vec{x}-\vec{b} \perp \text{Col}(A)$; If $A^TA > 0$, $K(A^TA) = K^2(A)$

[Defn] Let $A \in M_{m\times n}(\mathbb{R})$, QR-factorization writes A as

[$A = QR$] where $Q \in M_{m\times m}(\mathbb{R})$, orthogonal, $R \in M_{m\times n}(\mathbb{R})$ "upper-triangular" → if $m \ge n$, $R = \begin{bmatrix} U_{n\times n} \\ O_{(m-n)\times n}\end{bmatrix}$ ← upper-triangular matrix; ← zero-matrix

(e.g. Apply QR-factorization to least-square:

$\|A\vec{x}-\vec{b}\|_2^2 = \|Q^T(A\vec{x}-\vec{b})\|_2^2 = \|R\vec{x} - Q^T\vec{b}\|_2^2 \Rightarrow$ becomes $\min_{\vec{x}} \|R_1\vec{x} - Q^T\vec{b}_1\|_2^2$,

where $R_1 = U_{n\times n}$, $\vec{b}_1 = \vec{b}(1:n)$, only need to let [$R_1\vec{x} = Q^T\vec{b}_1$] ← backward substitution)

(ii) Computation QR-factorization:

[Defn] Householder Reflection: a matrix $H_{\vec{u}}$ w.r.t. u s.t. [$H_{\vec{u}} = I - 2\vec{u}\vec{u}^T$,]

where $\|\vec{u}\|_2 = 1$.

Properties of $H_{\vec{u}}$'s: symmetric, orthogonal ← ei-values either 1 or −1

$H_{\vec{u}}$ reflects vector $\vec{v}$ through the hyper-plane orthogonal to $\vec{u}$

zeroing a column: $A = QR \Leftrightarrow Q^TA = R$ ← make 1st col. $\vec{R}_1$ of R only with the first entry. Let $\vec{v}_1 = \vec{A}_1 + \alpha_1\vec{e}_1$, $H_{\vec{u}_1}$, where $\vec{u}_1 = \frac{\vec{v}_1}{\|\vec{v}_1\|_2}$

$\Rightarrow$ [$H_{\vec{u}_1}\cdot\vec{A}_1 = \vec{A}_1 - 2\frac{\vec{v}_1^T\vec{A}_1}{\|\vec{v}_1\|_2^2}\cdot\vec{v}_1 = [I - \frac{2(\vec{A}_1+\alpha_1\vec{e}_1)^T\vec{A}_1}{\vec{A}_1^T\vec{A}_1 + 2\alpha_1A_{1,1} + \alpha_1^2}]\vec{A}_1 - \frac{2(\vec{A}_1+\alpha_1\vec{e}_1)^T\vec{A}_1\cdot\vec{A}_1 - \vec{A}_1^T\vec{A}_1 + 2\alpha_1A_{1,1} + \alpha_1^2}{\vec{A}_1^T\vec{A}_1 + 2\alpha_1A_{1,1} + \alpha_1^2}\vec{e}_1$]

want: $H_{\vec{u}_1}\cdot\vec{A}_1 = r\vec{e}_1 \Rightarrow \vec{A}_1^T\vec{A}_1 = \alpha_1^2 \Rightarrow \alpha_1 = \pm\|\vec{A}_1\|_2$ choice $\alpha_1 = \text{sgn}(A_{1,1})\|\vec{A}_1\|_2$

Finally, $\vec{u}_1 = \frac{\vec{A}_1 + \alpha_1\vec{e}_1}{\sqrt{2\alpha_1(\alpha_1 + A_{1,1})}}$.

For the 2nd – let $\hat{H}_{\vec{u}_2}$ be $(n-1)\times(n-1)$ householder reflection, $H_{\vec{u}_2} = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{H}_{\vec{u}_2}\end{array}\right]$...

Memo No. 40
Date 2022 / Dec / 6

Let $Q = H\vec{u}_1 \cdot H\vec{u}_2 \cdots H\vec{u}_n$, thus get $Q^TA = R$. (Note $H\vec{u}\cdot A \sim O(n^2)$)

(iii) Find all ei-values of a square matrix

Finding ei-values by computing $\chi_A(\lambda)$ & finding roots is ill-conditioned. (sensitive to coefficient change)

Nonetheless – the opposite way works well

↓ (normalize a polynomial $p \to$ construct (companion matrix) A ($\uparrow (-1)^n$ for x^n) $\to$ compute ei-values of A)

(e.g. "roots" in MATLAB)

☆ The [power method:] (compute ei-values / ei-vectors with QR method)

IDEA: Start with a random vector & keep multiplying it with A

Suppose A has n ei-values, denoted as $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$

[1st-version] every time multiplying with A, amplifies the dominant components

Assume ei-vectors of A can be a basis of $\mathbb{R}^n$, denoted as $\vec{x}_1, \cdots, \vec{x}_n$

$$\vec{q}_0 = \sum_{i=1}^{n} \alpha_i \vec{x}_i \Rightarrow q^{(k)} = A^k \vec{q}_0 = \sum_{i=1}^{n} \alpha_i \lambda_i^k \vec{x}_i$$

(λ_i^k: as $k\uparrow$, λ_1^k dominates; If $|\lambda_1|>1$, tends to ∞)

[Normalization] (2nd-version) Still with a random $\vec{q}_0$, but $\vec{z}^{(k)} = A\vec{q}^{(k-1)}$ & $\vec{q}^{(k)} = \frac{\vec{z}^{(k)}}{\|\vec{z}^{(k)}\|}$ (often $\|\cdot\| \triangleq \|\cdot\|_\infty$)

Thus, $\vec{q}^{(k)} \underset{(k\to\infty)}{\longrightarrow}$ converges to ei-vector corresponding to λ_1

Find λ w.r.t. ei-vector $\vec{q}$: (least square $\min_\lambda \|A\vec{q} - \lambda\vec{q}\|_2^2$ → root of an 2nd order eqn)

also Rayleigh Quotient $\lambda = \frac{q^TAq}{q^Tq}$ (symmetric A); $\frac{q^T(\frac{A+A^T}{2})q}{q^Tq}$ (generally)

(Issue: $|\lambda_1| = |\lambda_2| = \cdots$, diff largest ei-value with the same modulus and different ei-vectors)

Case I: $\lambda_1 + \lambda_2 = 0$ become oscillations $\vec{q}^{(k)} = \frac{\lambda_1^k(\alpha_1\vec{x}_1 + (-1)^k\alpha_2\vec{x}_2)}{\|\cdots\|_\infty}$

do $\vec{q}^{(2k)}$, $\vec{q}^{(2k-1)}$ converge to diff vectors $\Rightarrow$ can get $\vec{x}_1$, $\vec{x}_2$ (with multiplicities)

Memo No. 41
Date 2022 / Dec / 8

Case II: $\lambda_1 = \lambda_2$, still can get $\alpha_1\vec{x}_1$, $\alpha_2\vec{x}_2$ ...

Note that "floating arithmetic", rounding-off error (→ out-of-trouble with error!) makes λ_1 component non-zero $\Rightarrow$ whatever initial point, converges to the dominant (λ_1)

[Shift the origin] Let $B = A - \mu I$, converges to δ, which is other ei-values the dominant ei-value of B (how to choose?)

$\therefore$ get $\mu + \delta \to$ ei-value of A.

[Inverse Power Method] Find the smallest (absolute value sense) ei-value

(3rd version) replace A by A^{-1}, pick $\vec{q}_0 = \sum_{i=1}^{n} \alpha_i \vec{x}_i$

$$\Rightarrow \begin{cases} \vec{z}^{(k+1)} = A^{-1}\vec{q}^{(k)} \text{ (by solving } A\vec{z}^{(k+1)} = \vec{q}^{(k)}) \\ \vec{q}^{(k+1)} = \frac{\vec{z}^{(k+1)}}{\|\vec{z}^{(k+1)}\|} \text{ (normalization)} \end{cases}$$