Random Numbers

Simulation, sampling, Optimization, Numerical integration, Testing
Decision making, Recreation

Uncomprehensible - ~~Pseudo~~ Random
Pseudo random

Random number ⇒ uniform distribution
normal/poisson can be generated

10,000,000

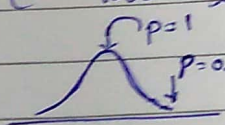
$\binom{200,000}{100,000} \times \frac{1}{10^5} \times \frac{1}{10^5} \times$

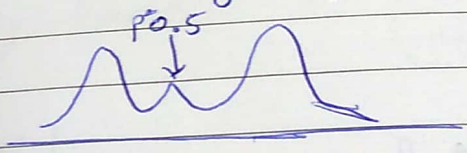
$\frac{1000,000!}{(200,000!)(800,000!)} \times \frac{200,000!}{(100,000!)(100,000!)} \times \frac{1}{10^{10}}$

* Null test ; Multiple simulation hypotheses
p-value of surprise high p value - less surprise

(Non-) Parametric
T-test method; $Z\text{score} = \frac{x-\mu}{\sigma}$ How many std deviations are you away from the mean

One tail test.





* Coding the Matrix; Book
# Computer Age Statistical Inference.

Jon von Neumann's Middle square method;
1359; 1225; 0484; ...

Knuth's Algorithm K; "Super-random" number generator
Started to converge
Random number should be generated randomly;

→ Random Number Generators;

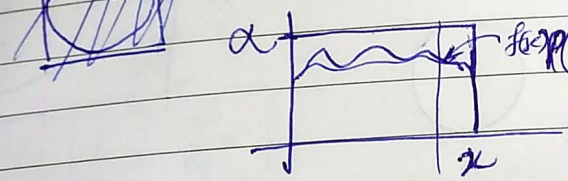
$$1 - \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \cdots$$

$$1 - \prod_{n=1}^{n} \frac{365 - n}{365}$$

# of people: n+2

Buffon's needle

Monte Carlo Simulation for integration



$$\text{for } P(f(x) < f(x)) \times P(a \cdot x)$$

Linear Congruential Generator;

$$X_{n+1} = (aX_n + c) \bmod (m), \quad 0 \le a < m$$

m; modulus; m > 0; 0 ≤ c < m; X_0 = seed value

Random seed

Try: m = 16, a = 7, c = 7, X_0 = 7;
m = 4096; a = 69, c = 853, X_0 = 0;

Bootstrapping;

Scanned by CamScanner

$A x = \lambda x$ RQ

$QRx = \lambda x$

$RQRx = \lambda(Rx)$ Rx

$A = Q \Lambda Q^{-1}$

$RQR = R\Lambda R^{-1}$

=

Monte Carlo Simulation:

bayesian relative theory & frequency theory

$P_r(A) = \lim_{n \to \infty} \frac{n_a}{n}$

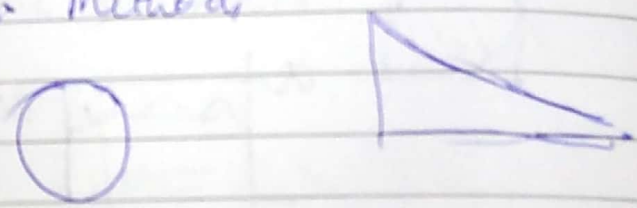
A random Variate

$0 < u < 1$; $a \le a + [b-a]u] \le b$

Always try to use a built in random generator

randperm gives random

Acceptance rejection method



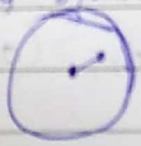
$f_r(c)$

→ Bootstrapping → data analytics & statistical inferences

→ Discrete event simulation

→ Prob. stochastics Pillai

→ Markov chains



$f(r, \theta) = ?$

$f_R(r) = \int f(\theta) d\theta$; $F_R(r) = \int$

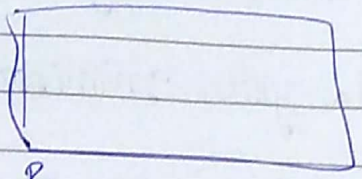
$f_\theta(\theta) = \int f(r, \theta) dr$

f

$\int f_\theta$

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

~~for~~



P=

$$A = \begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix} \begin{matrix} 1/3 & 2/3 \\ -1/3 & 1/3 \end{matrix}$$

$$\begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 7^{25} & 0 \\ 0 & 4^{25} \end{bmatrix}$$

$$\begin{bmatrix} 7^{25} & -2 \cdot 4^{25} \\ 7^{25} & 4^{25} \end{bmatrix} \begin{bmatrix} 1/3 & 2/3 \\ -1/3 & 1/3 \end{bmatrix}$$

$$= \frac{7^{25} + 2 \cdot 4^{25}}{3} \quad \frac{2 \cdot 7^{25} - 2 \cdot 4^{25}}{3}$$

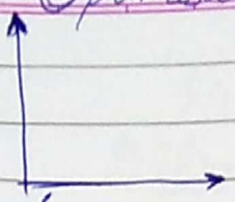
$$\frac{7^{25} - 4^{25}}{3} \quad \frac{2 \cdot 7^{25} + 4^{25}}{3}$$

8/3/2019

Optimisation

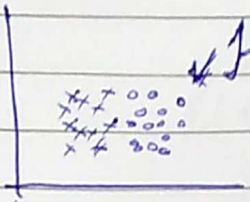
$$f(\theta) = \sum_i \left(\frac{x_{mi} - x_{pi}(\theta)}{x_{mi}} \right)^2$$

$x_{pi}(\theta)$ is a very complex function.



maximize / minimize $f(\theta)$

Given a function How would you minimize $f(\theta)$?



* How would you divide these points into clusters?

* Gradients

Minimize $f_0(x)$ subject to $f_i(x) \leq b_i$; $i = 1, 2, \ldots m$

Constraints of optimisation.

Special Case Convex optimisation: Objective & constraints are convex.

A set such that on joining two points all points lie within the set.

$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$, $\alpha + \beta = 1$; $\alpha \geq 0$; $\beta \geq 0$;

Convexity is more generale than linearity

Common Optimisation:

Least Square problems, LP; QP; (Mixed) integer programming, solutions are integer

Dynamic Programming, Local optimisation & global optimisation, multi-objective optimisation — more than one objective functions; pareto optimal point

→ Practical methods of Optimization.

→ How to solve it: Modern Heuristics

→ Clever Algorithms.

Gradient based Algos & Direct search algos;

Local, Global, Convex & Stochastics.

Gradient based Methods:

① → General recipe: Start with initial guess x_0

$$x_{k+1} = x_k + \alpha_k p_k$$

② → Test for convergence (Testing quality of the solution)

2. Find a search direction p_k (opposite to the direction of gradient)

3. Decide a step-length : α_k

4 Update x_{k+1} and go back to 0;

$$x_{k+1} = x_k + \alpha_k p_k$$

$$f(x,y) = 4x^2 - 4xy + 2y^2;$$

$f(2,3)$ ~~$f(2,3) = 16$~~

Search direction: $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (8x - 4y, 4y - 4x) = 6$

$$x_{k+1} = x_k - \alpha_k (8x - 4y, 4y - 4x)$$
$$= (2,3) - \alpha (-4\alpha, 4\alpha) = (2 - 4\alpha, 3 + 4\alpha)$$

$$f(x,y) = 8 \cdot (2-4\alpha)^2 - 4(-4(3-4\alpha) + 4(3-4\alpha) - 4(2-4\alpha)(-4)$$
$$\Rightarrow \alpha = 1/2;$$
$$= (0,1)$$

$F(x,y) = F(2-4\alpha, 3-4\alpha) = (2-4\alpha)^2 - 4(2-4\alpha)(3-4\alpha) + 2(3-4\alpha)^2$

$F'(2-4\alpha, 3-4\alpha) = 0$ find α, and then $x_{k+1} = x_k - \alpha_k$

4/3/2019

→ Steepest Descent Method
→ Conjugate Gradient Method.
→ Newton's Method (requires Hessian at every step)
→ Modified Newton's Method
→ Quasi - Newton Methods (avoids Hessians.
→ DFP;
→ SFGS ; → SR1
→ Trust region methods

Direct search methods

~~Many real~~ challenges -

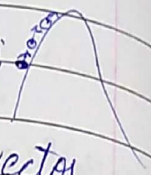
→ Very high dimensionality

→ Many local minima

→ discrete search spaces

→ Mixed variables

* Methods cannot compute / use gradient information



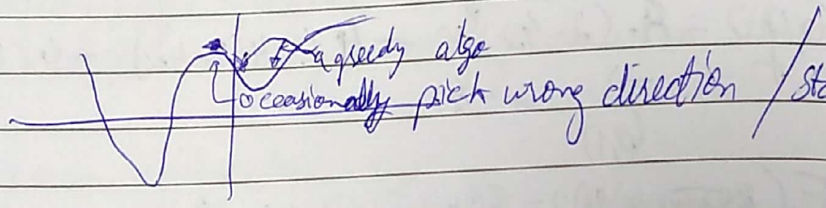
→ strategy to vary parameter vector

→ Strategy to accept / reject a new parameter vector

Storn & Price 1997 J. of Global optimization

Greedy - locally optimal choice / short sighted

→ Greedy decision process, converges fairly fast - ~~more~~ risk of getting track in local minima



→ Ability to handle non-linear, non-differentiable / multimodal cost function.

→ Parallelizable

→ Ease of use. (Eg: α - variable steering minimization / hyperparameter minimizing hyper parameters.

→ Robust & easy to choose

→ Consistent Convergence to global minimum in consecutive independent trial.

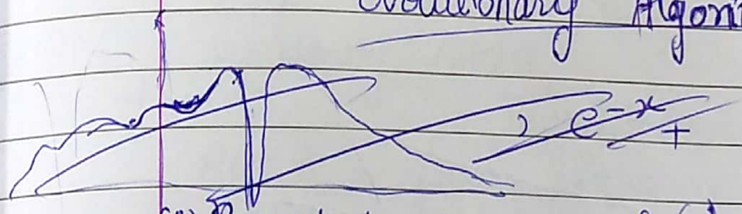
→ Hooke – Jeeve's Pattern Search
→ Downhill Simplex (fminsearch)/ Nelder – Mead Simplex
→ Grid Search
→ Random Search
→ Hill climbing

Stochastic Search Algorithms:

(i) Simulated ~~Algorithms~~ Annealing
(ii) Swarm Algorithms (Eg Ant modelled / Nature inspired)
(iii) ~~Genetic Algorithm~~ Evolutionary Algos

→ Simulated Annealing:
More exploratory at "higher ~~no of iteration~~ temperature" & more convergi[ng] at low temp.
Usually accept moves that reduce cost and occasionally accept the moves that increase the cost.
→ Metropolis Criterion
Initial temperature; ~~annealing~~ cooling schedule, Length of run; stopping conditions, trials & errors.

Evolutionary Algorithms



(i) Property to pass on & (ii) population of solution
(iii) Selection pressure property
"Computational procedures patterned on biological evolution"

13/3/2019 Search procedure that probabilistically applies search operators to a set of points in a search space

Lamarckian Algorithm
Darwin: – Natural selection of the fittest.
→ Mendelian genetics → Genotype – Phenotype mapping

Evolutionary Algorithms
→ Genetic Algorithms (Binary/Integer)
→ Evolutionary Programming
→ Evolution Strategies (Real Numbers)

Genetic Programming
→ Evolvable softwares
→ Evolvable hardwares

GAi
Population

Individuals/Chromosome

Mutation

Recombination/Crossover

Fitness function

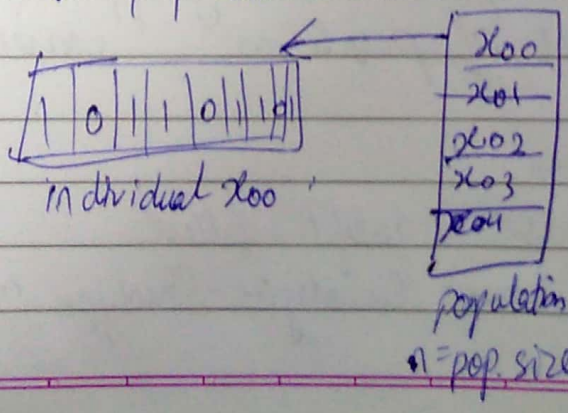
Selection

Generation,

Population size.

Total generation HP3 Hyperparameters

n - Population size 0 → x^{th} generation



individual x00

population
n = pop size HP-2

Perform mutation: / Swap / Redefine operators in each chose
Select a bit and flip it.

Nμ = Number of mutation in a generation — HP3

Make few children & recombine.

10101 011 → 10101 110
10000 110 → 10000 011

} Many diff strategies
Hyper strategy!

→ Fitness ↑ function   Cost ↓ function   Hyper strategy:

Tornament playing, probabilistic/fitness proportionate selection, ~~elitist~~ elitist, random selection.
much like hill climbers

NASA antenna design.

~~NASA~~ Adrian Thomson
Evolutionary algorithms

~~→~~ Evolutionary strategies
→ Self-adaptation - genotype adapts to alter the evolutionary process

~~→~~ Impementation of EAs
FPGA - field programmable gate array - special arrays that specialise in doing a particular function.
→ Simple binary chromosome
→ Trees and complex data structures
→ Cartesian Genetic ~~algorithm~~ Programming (For Evolvable Hardware)

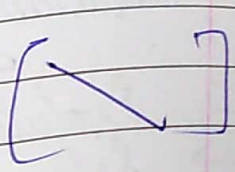
GGPU - General purpose graphic processing

Macro-mutation: Large changes in alleles without recombination

Any optimisation problems:
→ Specially useful if search space is poorly characterized.
→ Might help understand the problem better.

Gene expression programming
Ant colony Opt.
Particle Swarm Opt
Tabu Search.

1. diag solv
2. back subs
3. Goussian
4. LU
5. Cholesky
6. Householder
7. Power iter
8. QR factorisation
9. Birthday
10. Integration Monte Carlo
11. Value of Pi
12. Unit circle
13. Singular Value decomposition



$A_0 = Q_0 R_0$

$A_1 = R_0 Q_0 = Q_1 R_1$

$A_1 = Q_1 R_1$

$A_{k+1} = Q_k R_k Q_k$

$A_{k+1} = Q_k^{-1} A_k Q_k$

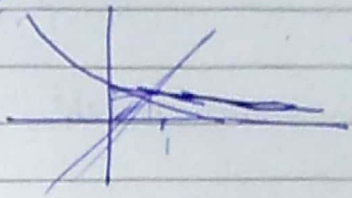
Diagonal

18/3/2019 Solution of Nonlinear Equations.
(Root Finding Problems).

Given a continuous function $f(x)$, find the value x such that $f(x) = 0$?

Analytical solutions → only special equations

Graphical method $x - e^{x}$



root ≈ 0.6;

root ∈ [0, 1];

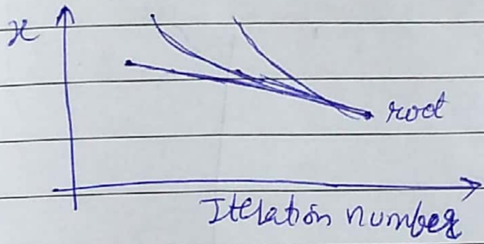
Numerical methods:

- → Bisection Method
- → Newton's Method
- → Secant Method

→ false position Method
→ Muller's method
→

→ Open Method:
starts with initial guess and in each iteration, a new guess of the root is estimated. ~~Usually more efficient than the bracket method~~

→ Convergence notation of order p: $\left|\dfrac{x_{n+1} - x}{(x_n - x)^p}\right| \leq C$



→ Bracketed Method: Boundaries are set.

Bisection Algorithm,
Let $f(x)$ be defined on interval $[a, b]$;
IMVT;
$f(a) \cdot f(b) < 0$, then atleast one zero in the interval $[a, b]$;