Random Walks and optimization

Find global minimum/maximum of $f(x)$ objective/cost function

$f(x)$ has many local minima/maxima

Maybe f is not smooth enough to take derivatives

How can we use random walks in a search algorithm for the minima?

Hill climbing algorithm
Start at some initial point move to the best neighboring point that reduces the value of $f(x)$

Trouble: Can get stuck on local minima.

Simulated annealing

Inspired by annealing in metallurgy
heat metal to high temp then cool slowly reducing temp so metal reaches a state of low energy where it's stronger.

Combine random walk and hill climbing

Instead of picking the best move, pick a random move

If selected move improves solution, accept it
results in f of lower value

Selected move can still be accepted even if it worsens the solution ie. bad move
with some probability

Probability that a bad move will be accepted goes down the more steps of the random walk are taken

Allows the algorithm to search a wider area for possible solutions more likely to escape local minima if stuck

At beginning of random walk:

"temperature is high" - algorithm more likely to accept bad moves

"Temperature" gradually goes down - probability of accepting bad moves goes down

"At low temperatures" - algorithm less likely to accept bad moves
- mostly good moves

Applications

1. Traveling Salesman Problem

Given a list of cities & distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the original city.

operations research & theoretical computer science

2. Protein folding

3. Thermodynamics

4. Many other optimization areas

Why Simulated annealing?

→ Use ideas to minimize some objective function in your model

→ Recognize usefulness of stochasticity in problem solving

* Simulated annealing still doesn't use historical information, no memory

Pseudocode

Let $x = x_0$, $f(x) = f(x_0)$

Define n total number of steps in random walk

For j to n

Set temperature

Pick random move $x_{new} = x +$ random variable

Compute $f(x_{new})$

if $f(x_{new}) < f(x)$ accept

else if $f(x_{new}) > f(x)$ but temperature high accept

else try another step

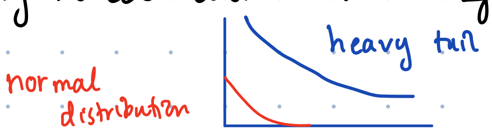
$x \leftarrow x_{new}$

Lévy Flight

Generalization of random walk that uses random step and random direction.

Step lengths taken from heavy tailed random distribution
Directions taken from uniform distribution

Heavy tailed distributions eg. log-normal distribution, Cauchy distribution



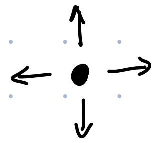
Levy flights are random processes where large steps are more likely.

Levy flights model activities involving a lot of small steps interspersed with occassional long steps

eg. Hunting patterns of animals — an animal taking equal steps will take longer to get food.
Games of hide & seek

Visit few areas within a short distance before moving on to a far-off area

Random walk in 2D



Run code
What are the differences between random walk with integer steps vs. random walk with real steps?

Modify code to plot histograms : Look at Levy flight code & random_walk1D

ns = 100, N = 1e5

Define X1store = zeros(N,2)

X2store = zeros(N,2)

add an outer for loop for Monte Carlo — for k 1:N

Initialize X1 & X2 to zero

```
for j=1:ns
    X1=
    X2=
end
    X1store(k,:) = X1(end,:)
    X2store(k,:) = X2(end,:)
end
```

Can modify to overwrite

hist3(X, {ax, ax})

Copy ax from levy code

Plot 2 subplots

left — Random walk integer

right — Random walk real