MATH 242: Monday, May 4, 2020

TODAY: Reminders — seminar/colloquium assignment
Final Project

SIMULATED ANNEALING: using MCMC to solve optimization problems

from metalworking

PROBLEM: Find 10 numbers whose sum is 100 and whose product is maximum.

Example: $5+7+20+8+10+1+19+4+6+20 = 100$
product: 5107 20000

Markov Chain:

States are sets of 10 non-negative integers that sum to 100.

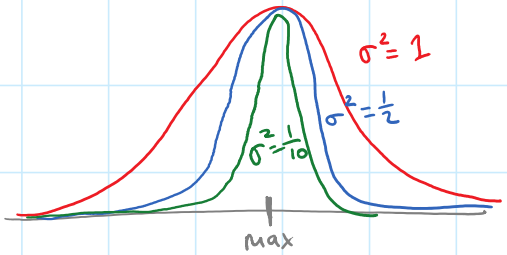
How many states? 6,292,069

state: $(n_1, n_2, n_3, \ldots, n_{10})$

Frequency Function: product $g(n_1,\ldots,n_{10}) = n_1 \cdots n_{10}$ ⇐ Maximize This!

Define $f(n_1,\ldots,n_{10}) = e^{g(n_1,\ldots,n_{10})/\sigma^2}$

"variance" tuning parameter



As σ^2 decreases towards zero, near-optimal states become much more frequent than states that are far from optimal

We could try: $f(n_1, \dots, n_{10}) = e^{n_1 \cdots n_{10}/\sigma^2}$ Too big!

Instead, maximize the log of the product:

IMPORTANT → $h(n_1, \dots, n_{10}) = \log(n_1 \cdots n_{10} + 0.1)$

($n_1 \cdots n_{10}$ could be zero)

frequency: $f(n_1, \dots, n_{10}) = e^{h(n_1, \dots, n_{10})/\sigma^2} = e^{\log(n_1 \cdots n_{10} + 0.1)/\sigma^2}$

Transitions: Choose two of the ten numbers.
Add 1 to a number, subtract 1 from the other.

Example: 5, 7, 20, 8, 10, 1, 19, 4, 6, 20

5, 8, 20, 8, 10, 0, 19, 4, 6, 20 ← proposed state

RECALL: If proposed state has greater frequency than current state, then move to proposed state.

Otherwise, move with probability

$$\rho = \frac{f(\text{proposed state})}{f(\text{current state})}$$

We have:

$$\rho = \frac{e^{h(\text{prop State})/\sigma^2}}{e^{h(\text{curr State})/\sigma^2}} = e^{[h(\text{prop State}) - h(\text{curr State})]/\sigma^2}$$

```
def doMove(currState, sig2):
    # propose a move
    propState = proposal(currState)

    # compute change in h
    dh = h(propState) - h(currState)

    # decide whether or not to move
    if dh >= 0:  # then move
        currState = propState
    else:  # then maybe move
        rho = math.exp(dh/sig2)
        rand = random.random()
        if rand < rho:
            currState = propState

return currState
```

What do we do with σ^2?

Plan:

```
sig2 = 1
decFac = 0.9999
```

EXPERIMENT

```
# choose a random starting state
# simulate the random walk
loop:
    # take a step
    # decrease sig2
    sig2 = sig2 * decFac
# output solution
```

We don't want to decrease σ^2 too fast .... or too slow!