Scientific Computing

Monday, March 23

Announcements

* Homework 4 due Friday!
* Monday, April 6: No lecture, work from home day

Office Hours:

Mon, 9:30-10:30
Fri, 2:00-3:00

Cudahy 307

Topic 13 - Simulated Annealing

* Annealing Metal

[youtube video]

* Simulated Annealing is a MH inspired by the physical process.

Hill climbing: only moves that improve score are allowed.

SA: Worsening moves are accepted with some probability

At the start, the system has a high temperature and the prob. of accepting a worse move is high.

Over time, the system cools down (lower temp.) and the probability of accepting a worse move decreases.

Very high temp: basically a random walk, accepting all tweaks

Very low temp: basically hill climbing

Intuition: As the system cools down, you hope to wander onto a good hill and stay there.

[Demo: Cns. Func. #1]

Technical Details:

Assuming maximizing!

* **Acceptance Condition:**

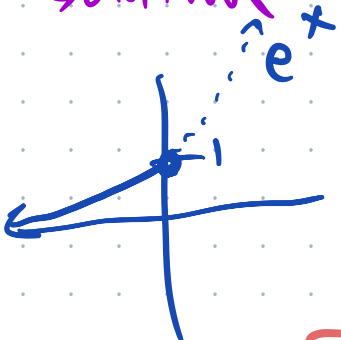
Suppose the current temperature is T.

current solution

Let $s = \text{tweak}(x)$ and $\underbrace{\Delta = \text{score}(s) - \text{score}(x)}_{\text{change in score}}$

possible next solution

If $\Delta > 0$, then s is an improvement over x, always accept



If $\Delta \le 0$, then s is worse than x.

Accept with probability $p = e^{\Delta/T}$. $e \approx 2.71$

$\left[T > 0, \text{ so } \Delta/T \le 0, \text{ so } 0 < e^{\Delta/T} \le 1 \right]$

High $T \Rightarrow \Delta/T$ close to $0 \Rightarrow e^{\Delta/T}$ close to 1

Why this formula $p = e^{\Delta/T}$?

Comes from the Boltzmann distribution in physics.

(probability a system is in a certain state given its energy)

If minimizing, $p = e^{-\Delta/T}$

* Cooling Schedule:

How the temperature changes over time.

- Geometric (most common)

  parameter α (decay rate)

  each time the temperature changes, we set

  $$T = T \cdot \alpha$$

  0.95 0.99 0.999

Ex: Initial Temp $T_0 = 10$, $\alpha = 0.9$

T goes $10 \to 9 \to 8.1 \to 7.29 \to \ldots$

(approaches 0)

The n^{th} temperature is $T_n = T_0 \cdot \alpha^n$

- Linear (less common)

  parameter β

  each time the temperature changes,

  we set $T = \underline{T - \beta}$.

  The n^{th} temperature is $T_n = T_0 - \beta \cdot n$

- Many other possibilities, even non-monotone ones!

Process:

Pick an initial temperature T (How?)

x = random solution

best = x ← best ever seen

Repeat: (How long?)

For a while: (How long?)

s = tweak(x)

Δ = score(s) - score(x)

if $\Delta > 0$:

x = s

if score(x) > score(best):

best = x

always move to improvements, save if best ever so far

else: (s is worse than x)

r = random # in [0,1] uniformly

if $r < e^{\Delta/T}$: prob. of accepting

x = s

adjust the temperature according to the (How?) cooling schedule

Questions to answer:

* How to pick initial temp
* How long to loop for each temp
* When to stop
* How to cool

art, not science

Picking the Initial Temp

First pick p_0: the initial probability with which you want worsening moves to be accepted.

What should it be? Opinions differ. Depends on the problem/landscape/etc. (art!)

↳ lots of little hills?
fewer huge hills?

$$p_0 \rightsquigarrow T_0$$

$p_0 = 0.9$: safer, if you have plenty of time, very random at the start

$p_0 = 0.5$: good in many cases

$p_0 = 0.2$: sometimes good, especially in slow cases or landscapes with many short hills (it's easy to get down off short hills to get to new ones)

Okay, how can we find the temp T that leads to p_0? Recall $p = e^{\Delta/T}$. But we don't know Δ.

We will approximate the average value of Δ, then use that:

```
trials = []
while len(trials) < 1000:    (or whatever)
    x = random solution
    s = tweak(x)
    if score(s) < score(x):    (worsening)
        trials.append(score(s) - score(x))
avg = sum(trials)/1000
```

how much worse it is

how much worse each worse thing is

Now we have an average Δ for a worse solution

$$p = e^{\Delta/T} \Rightarrow \ln(p) = \Delta/T \Rightarrow \boxed{T = \Delta/\ln(p)}$$

avg

like 0.9

So, a good starting temperature T_0 to force a particular p_0 is:

$$T_0 = \frac{\text{avg}\,\Delta}{\ln(p_0)}$$

This far from exact:

(1) We just took an average

(2) The avg. worsening in the context of running SA (going up and down) might be very different than the avg. worsening from a random solution.

While you're running Simulated Annealing, print the % of worsening solutions accepted at each temperature, and then you can adjust T_0 as needed.

How long do we run at a fixed temp before moving to the next temp?

Some possibilities:

* N tweak attempts in total
* K worsenings rejected or L worsenings accepted, whatever comes first

(ex: 1000 accepted or 75000 rejected)

How long do we cool the system before stopping

* Run out of time

* No worsening moves accepted in a while (basically hill-climbing)

* Pre-set end temperature $T_f = 0.001 \cdot T_0$.

No better solution found for 10 temperature changes