Topics in Astronomical Research
Computational Astrophysics

PHYS 913

Dr. Mark Richardson
ZoomID: 990 2595 7333: Pass: PHYS913
Mark.Richardson@queensu.ca

Last time …

- Simulating observations
- Fitting with regression
- Markov Chain Monte Carlo
- Bayesian Approach to MCMC

Today:

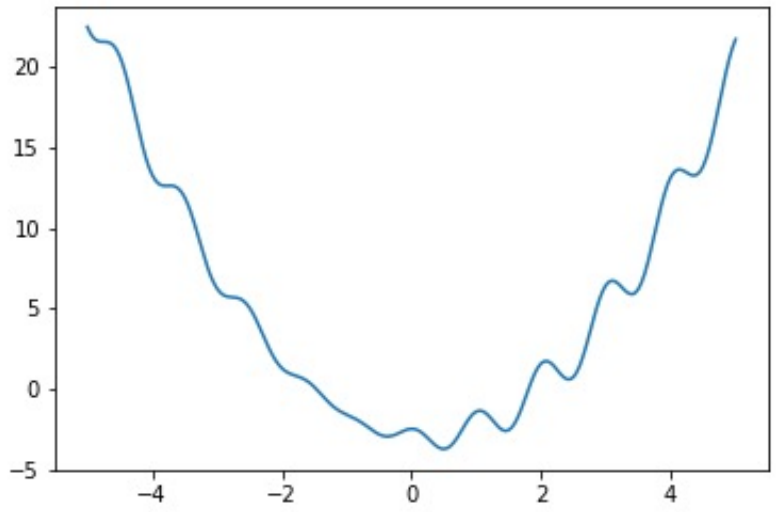
- Revisit MCMC: Simulated annealing
- More discussion about Bayesian Inference
- Brief intro to Machine Learning

Refresher on MCMC methods

- Markov Chain Monte Carlo is a method for sampling from a probability distribution, P(x). x could be a vector state.
- It yields importance sampling.
- Key steps:
  - Start with a some vector state x_i.
  - Propose a new vector state x_j.
  - The probability of accepting the new state = min($P(x_j)/P(x_i)$, 100%)
  - If rejected, reuse state *i*.
- Transitions: Obey detailed balance:
  - $$\frac{T_{ij}}{T_{ji}} = \frac{P(E_j)}{P(E_i)} = \frac{e^{-E_j\ /kT}/Z}{e^{-E_i\ /kT}/Z} = e^{-(E_j - E_i)/kT}$$

MCMC: Optimization

- While MCMC is great at sampling the parameter space, as seen last class, It's also a useful tool for minimizing a function.
- Recall: we often are looking for the global minimum.
- Methods like gradient descent (GD) will only find the bottom of the basin you start in.




```
In [87]: x0 = np.arange(-5,5.1,1.)
         for x00 in x0:
             x1 = grad_disc(x00,fofx,fprime,h=0.01,c=0.002,tol=1.e-5)
             print(x00,x1,fofx(x1))
```

```
-5.0 -4.7653113612906 21.5031533064071
-4.0 -3.8000651363889144 12.59523155571228
-3.0 -0.3763473516852742 -2.9038118234513406
-2.0 -0.376355933983961 -2.903811236048604
-1.0 -0.3763271539248255 -2.9038132012199287
0.0 -0.367786929212471 -2.9038141441120957
1.0 0.4981634324644772 -3.68195701212003
2.0 1.4549552609428011 -2.5628951007376566
3.0 2.426942723068116 0.6175994573939354
4.0 3.39830550316508 5.888852991198876
5.0 4.35739983480212 13.228583791227603
```

MCMC: Simulated Annealing (Newman 10.4)

- MCMC on its own will only sample the probability:
  - P(f(x);T) ~ exp(-f(x)/T)
- More points will sample minima in f(x), and you could look at your chains to determine the smallest f(x).
- This could inform an initial guess(es) for GD.
- However, let's consider the importance of T.
- What happens for large T?
- What happens as T → 0 ?

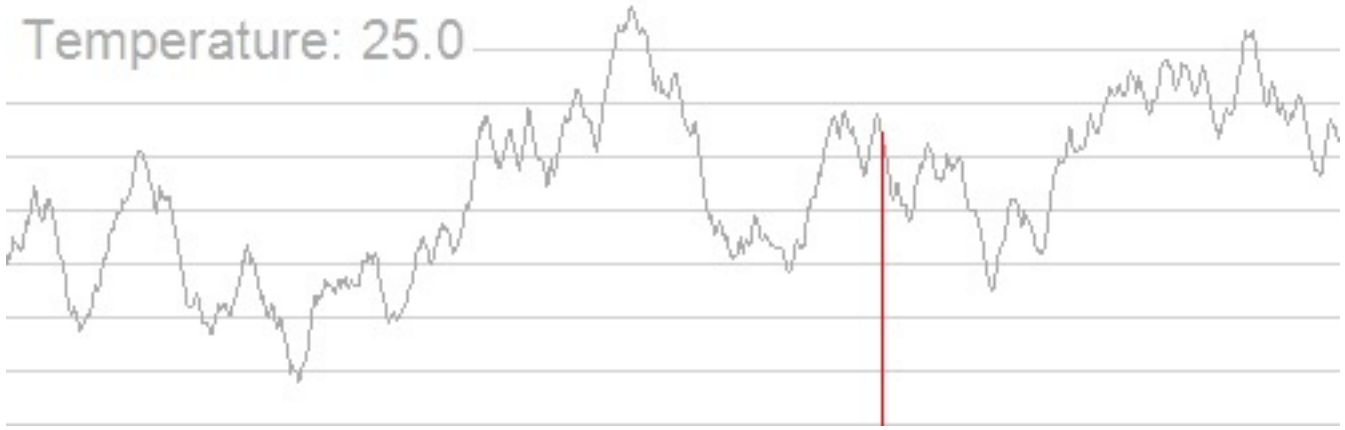
MCMC: Simulated Annealing (Newman 10.4)

- Simulated annealing is an approach to finding the global minimum (or maximum) of a function by emulating nature.

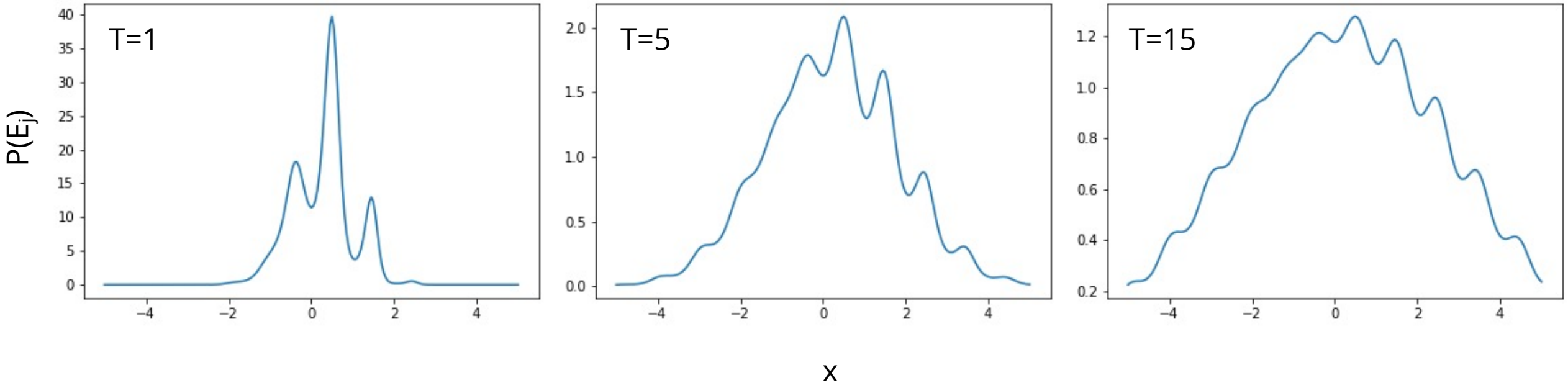
- Consider a metal or glass that cools very quickly: it will reach some local minimum of energy, but this isn't the global minimum.
  - This can mean it is still brittle, weak.
- Consider a metal of glass that cools slowly: it will find its lowest possible energy state.
  - This will make it the strongest.
  - This is called *annealing*.

Simulated Annealing

- Recall the transition probability: Probability of accepting state *j* from state *i:*
- $P(x_i, x_j; T) = e^{-(f(x_j)-f(x_i))/kT}$
- Large T → even if $f(x_j) > f(x_i)$ there is a high probability of accepting the new state.
- Low T → only accept state if $f(x_j) < f(x_i)$.



Example: Probability of x ~ exp(-f(x)/T)



Simulated Annealing

- General approach:
- Start with a large $T=T_0$, typically >> Δf
- Slowly decrease T→ $T=T_0 e^{-t/\tau}$
- Finding good values of T_0 and τ requires trial and error. Generally, better solutions are found for large T_0 and τ but will require longer to converge.
- For choosing a new state:
  - Discrete states: swap indices.
  - Continuous: choose a state nearby (use a normal distribution, with any constraints added).

Simulated Annealing: Traveling Salesperson

- Imagine a list of cities (with [x,y] positions on a map).
- You want to travel to all of the cities in the shortest amount of time (minimizes distance of root).
- Also typically want to return where you started.
- Typically very hard. Simulated Annealing was actually created to solve it.

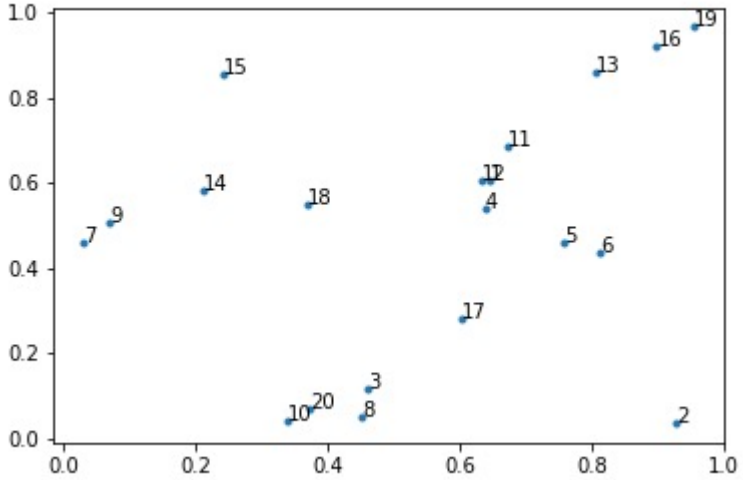
- Brute force: try combination. e.g., 5 cities: (n-1)! ways

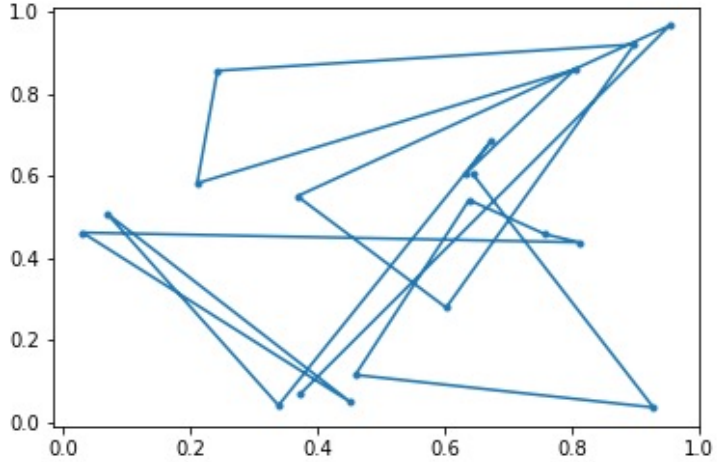
(1,2,3,4,5,1)	(1,3,2,4,5,1)	(1,4,2,3,5,1)	(1,5,2,3,4,1)
(1,2,3,5,4,1)	(1,3,2,5,4,1)	(1,4,2,5,3,1)	(1,5,2,4,3,1)
(1,2,4,3,5,1)	(1,3,4,2,5,1)	(1,4,3,2,5,1)	(1,5,3,2,4,1)
(1,2,4,5,3,1)	(1,3,4,5,2,1)	(1,4,3,5,2,1)	(1,5,3,4,2,1)
(1,2,5,3,4,1)	(1,3,5,2,4,1)	(1,4,5,2,3,1)	(1,5,4,2,4,1)
(1,2,5,4,3,1)	(1,3,5,4,2,1)	(1,4,5,3,2,1)	(1,5,4,3,2,1)

Example cont.

- Can be a huge parameter space.
- Minimizing the total distance function:
  - $d = \sum_{i=1}^{N} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$, with the N+1 point = first point to close the loop.
  - Here *i* refers to city in the ith place in the list.

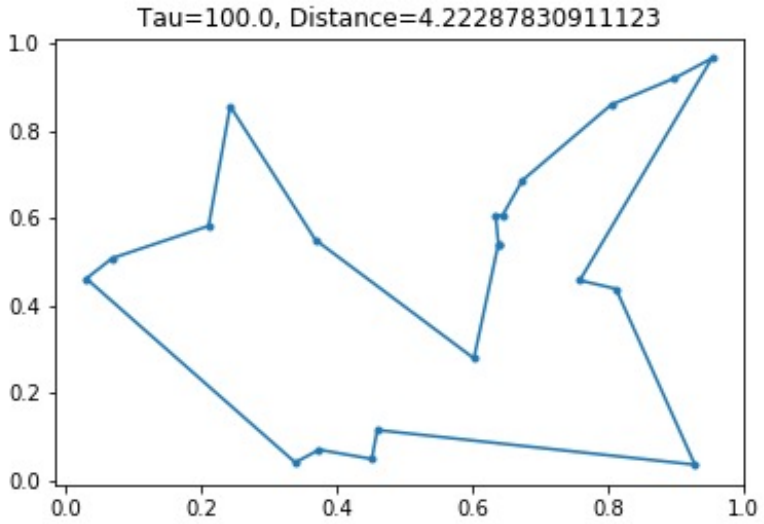
Example

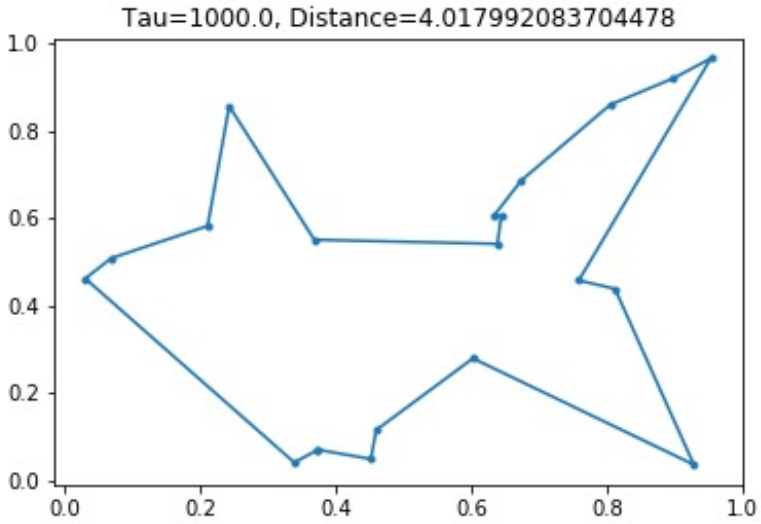


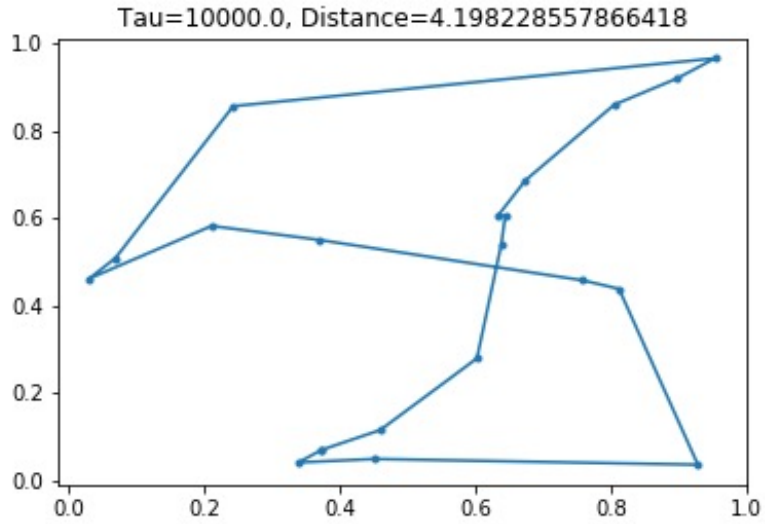


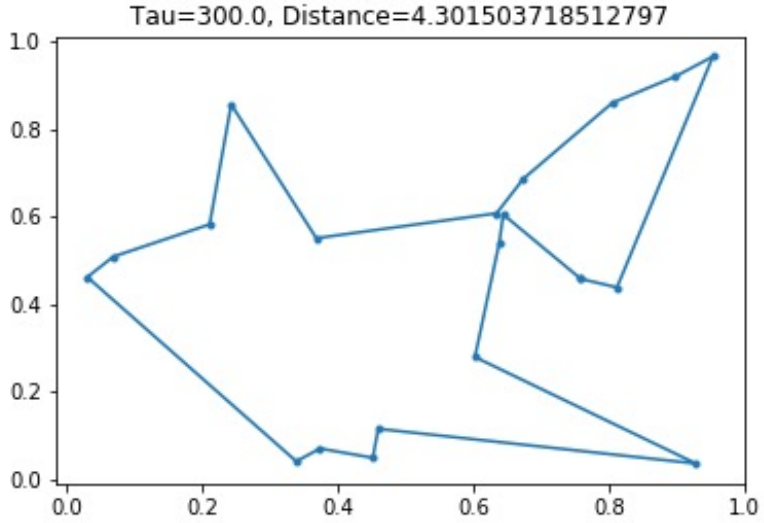
- I took T_0 to be N*sqrt(2)*2 → twice as long as going back and forth between the diagonals N times … this is much larger than the worst case scenario.
- τ took trial and error. I ended up using 10,000 which took ~20 seconds.
- Found a minimum of ~4: ~ tracing the outside of the domain.

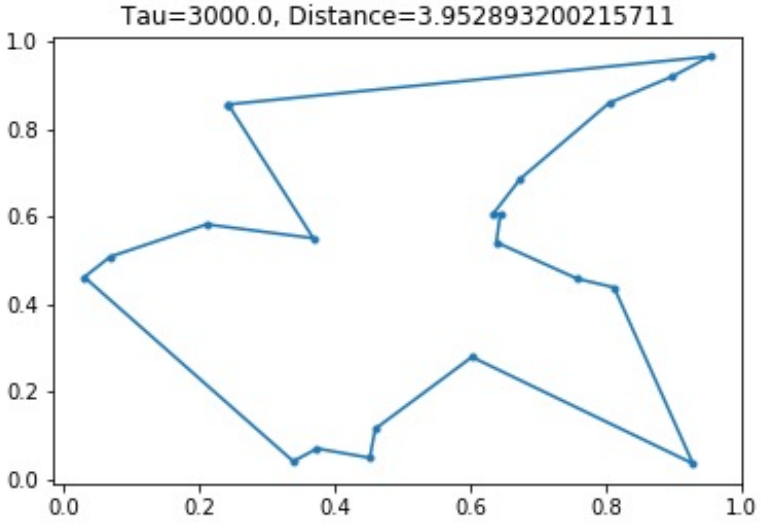
Traveling salesperson

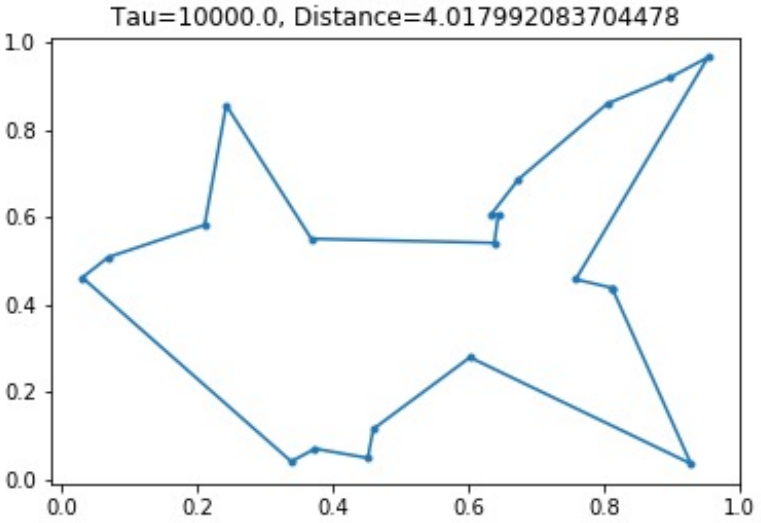










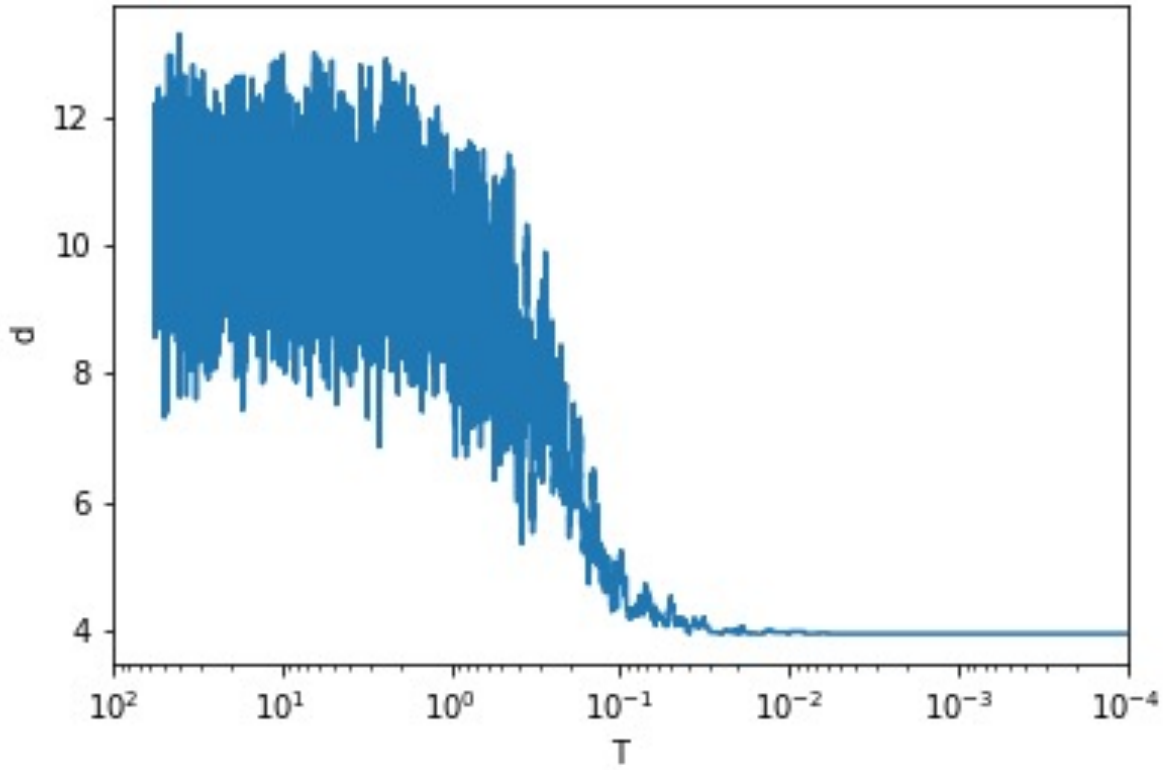


Simulated Annealing

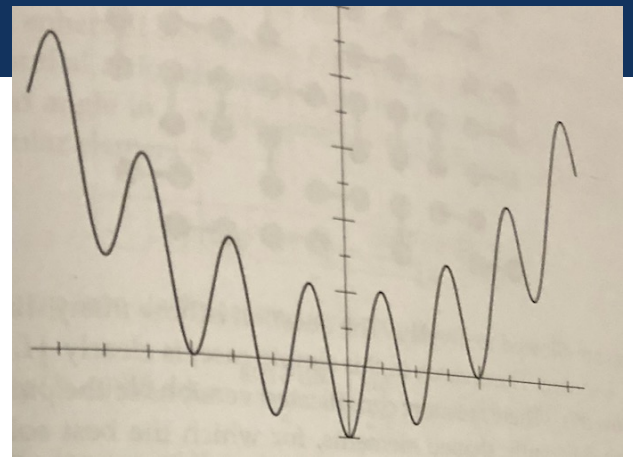
- **NB: Simulated annealing is a better approach to finding the global minimum, but it is not guaranteed to do so.
- Some extra tips:
- Restarting: Restart the annealing process from previous minimum value, possibly with a different temperature etc.
- Run several chains: Method may still get stuck in local minima: It finds a good solution, not necessarily the best.
- Some adaption for initial T: Generate an initial state, try N moves, if >= 80% rejected, increase T.

Simulated Annealing

- You can monitor your function value during the process: How well does our cooling function work?



Simulated Annealing for a function



- This is Q3 from Assignment 3 (Newman 10.10).
- Differences:
  - Choice of T_0 and τ
  - Finding x_j from x_i :
    - chose a random number, δ, on Norm(x=0,σ).
    - Let $x_j = x_i + \delta$
  - Function is given rather than dependent on the state.
  - Not guaranteed to get global minimum.
- Questions?

Back to Bayesian Inference

- As seen previously: Bayes theorem is:
- $\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Marginal Likelihood}} \rightarrow P(\boldsymbol{\theta}|D)d\boldsymbol{\theta} = \frac{P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}}{\int P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}}$

- MCMC allows you to sample the distributions with importance sampling.

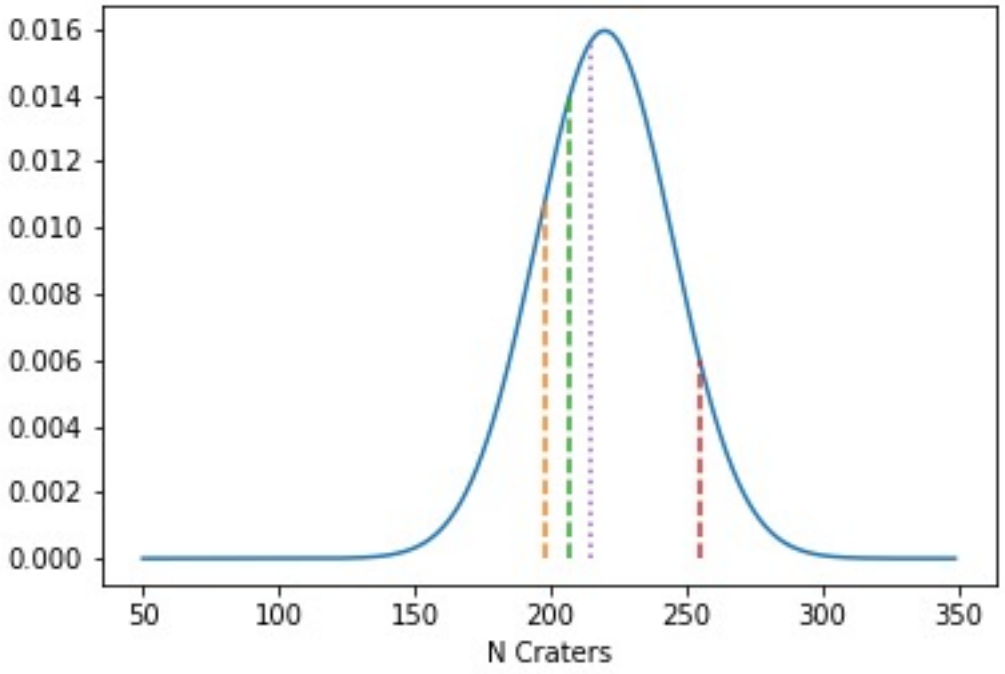
- Essentially this allows you to approximately integrate complex probability functions in high-dimensional parameter space.

Bayesian Inference quick example

- You're estimating the age of a region of Mars. You do this by crater counting (more craters means geologically older).
- A computer algorithm can look at a picture very quickly and count the number of craters, but it's fairly in accurate. Correct within 20%, 95% of the time.
- Humans are usually more accurate (say correct within 5%, 95% of the time), so you verify the number by having three students count by hand.
- Computer: There are 215 craters.
- Humans: There are {198, 207, 255} craters.

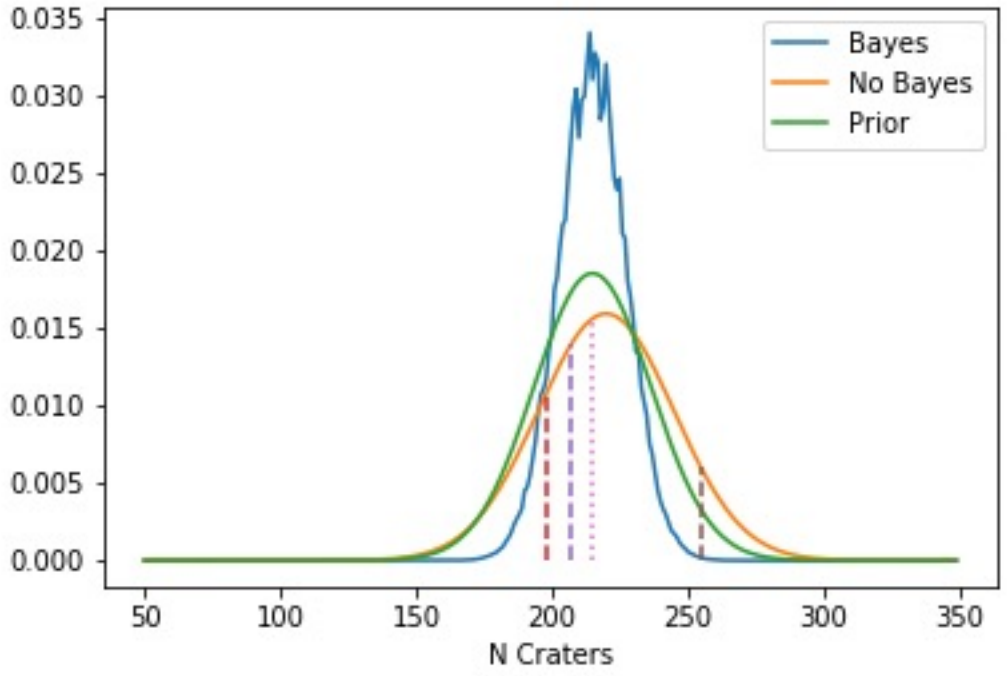
Example

- Without Bayes' and ignoring the computer:
  - <N> = 1/3(198 + 207 + 255) = 220.
  - $\sigma(N) = (1/3 [(198-220)^2 + (207-220)^2 + (255-220)^2])^{1/2} = 25.$
  - So Expect 220 +- 25.
  - Gives $P(N_{crater} \mid \text{Human Data})$
- Can we do better?



Example

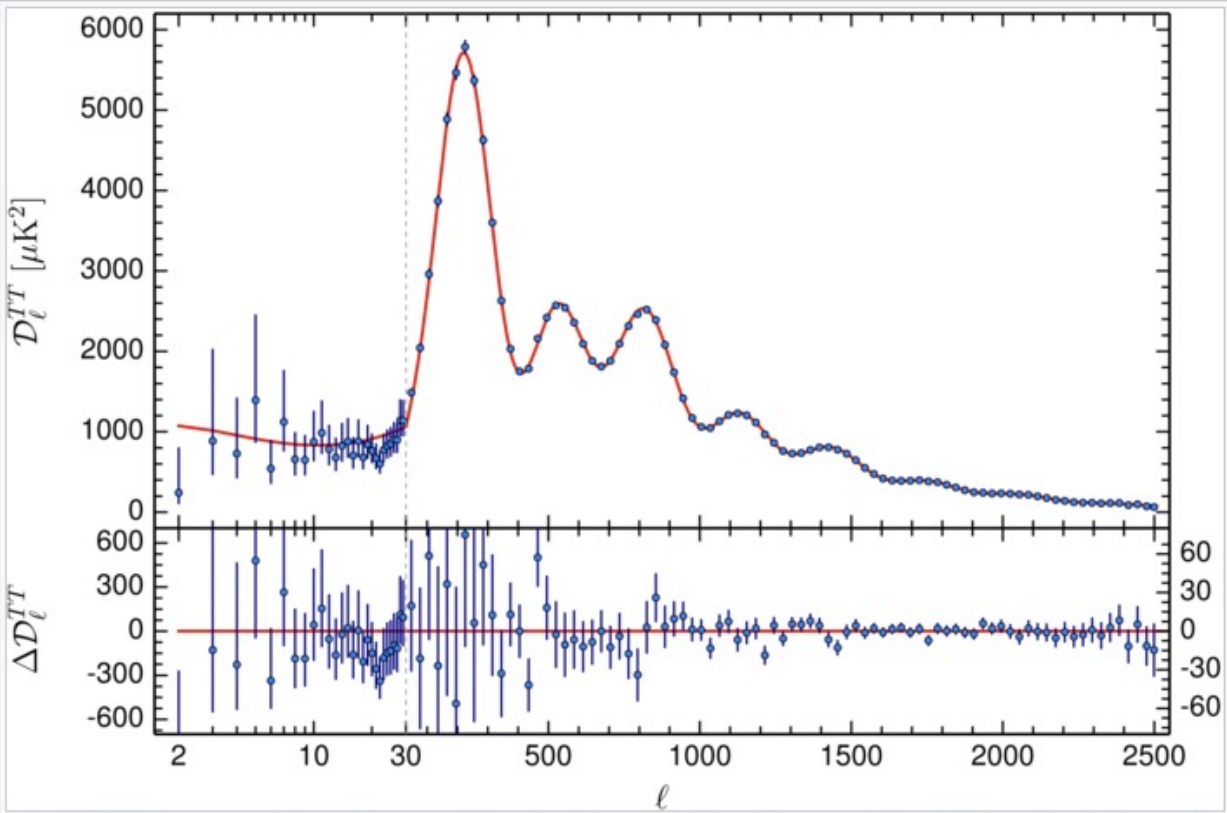
- Prior: P(θ)= Computer has 215 +- 22.
- $P(D|\theta) = P(D_1|\theta)* P(D_2|\theta)* P(D_3|\theta)$
  - P(x|θ) = normal(θ,σ,x)
- P(D) = normalization



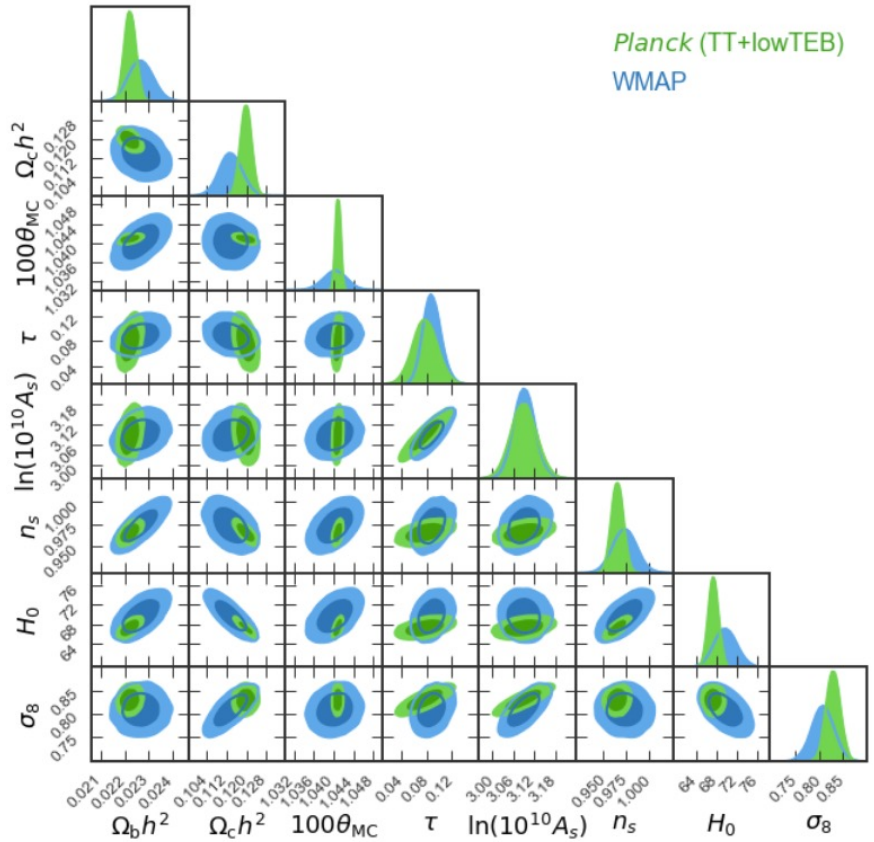
Another Example

- Example: A galaxy's star formation rate and metallicity can be determined by looking at specific nebular emission lines.
  - What are the:
    - Data, D?
    - Parameters, $\boldsymbol{\theta}$?
    - Priors?
    - Likelihood?
    - Nuisance parameters?

E.g., Planck Data



Planck 2015 TT power spectrum. The *x*-axis is logarithmic up to ℓ = 30 and linear at higher ℓ. The red line is the Planck best-fit primordial power spectrum (cf. Planck TT+lowP in table 3 of Planck-2015-A15[3]). Residuals with respect to this model are shown in the lower panel. The error bars show ±1σ uncertainties.



Machine Learning

- Broadly speaking:
  - An automatic system to learn, improve, optimize, and/or characterize data without being explicitly programmed to do so in a particular way.
- Common examples/uses:
  - Image classification: Hand writing, facial recognition, crater counting, Animal recognition etc.
  - SNe and other transient identification (visual, audio, etc.)
  - Diagnosis
- Given how quick humans are for some classification, it's surprising how hard it is to program such a process.

Machine Learning

- In the 60s, Marvin Minsky assigned a couple of undergrads to spend the summer programming a computer to use a camera to identify objects in a scene. He figured they'd have the problem solved by the end of the summer. Half a century later, we're still working on it.

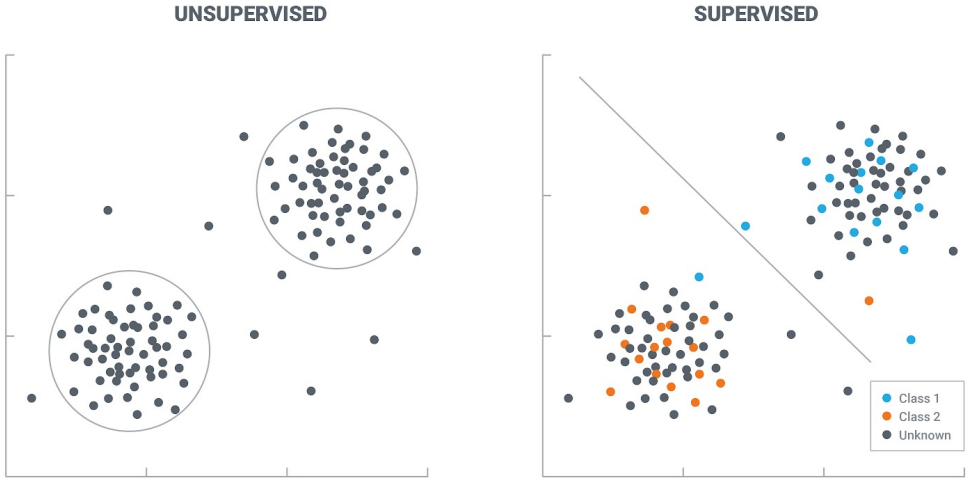
xkcd

Machine Learning: Unsupervised vs Supervised

- Supervised: You have training data that is already characterized, and you use this data train your machine learning program to characterize input data.
- Supervised: Typically done with Neural Networks. We will mostly look at these.
- Examples: Classification (Dog or Cat? Chihuahua or Muffin), Regression (result of a bunch of data; e.g., stock market or stellar mass, etc.)

Machine Learning: Unsupervised vs Supervised

- Unsupervised: Finds features, clusters, or associations in a dataset. Typically exploring high-dimensional dataspace where visually characterizing the data is difficult for humans.
- Here there is no clear answer (at least not one known in advance)
- Unsupervised: Commonly principal component analysis & cluster analysis.



Other machine learning

- Reinforcement learning: Determine a transition process (e.g., in a Markov Chain) to maximize a reward measure.
- Genetic Algorithms: Sometimes paired with e.g., reinforcement learning, using the ideas of genetic offspring to let successful models procreate and create new ones e.g., https://rednuht.org/genetic_cars_2/
- Semi-supervised: Mix of supervised (known answers) and unsupervised.

QSSETs

- Expires April 4
- Sorry that it’s early.
- Please do fill it in!

Next time:

- More Machine Learning
- Conclusion of the course
- Logistics for the presentations