Positive Monte Carlo – a nested sampling primer

John Skilling (john@skilling.co.uk)

MaxEnt2025 Auckland

We wish to evaluate $$Z = \int f(\mathbf{x})\, d\mu(\mathbf{x})$$ (coordinates; function ≥ 0; measure > 0)

e.g. Mass $= \int$ Density dVolume , Evidence $= \int$ Likelihood dPrior

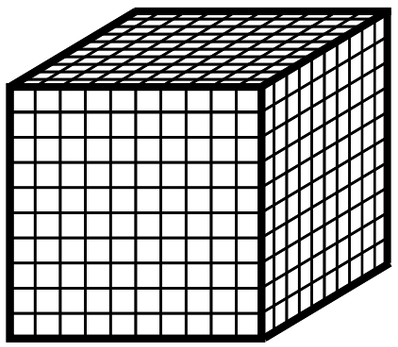
In statistical physics, Partition function $= \sum_{\text{states}} \exp(-\text{energy})$

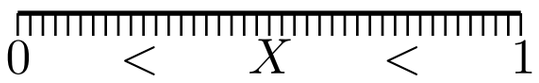
Use coordinates in which $\mathbf{x}$ represents the measure, $d\mu(\mathbf{x}) = d\mathbf{x}$, and normalise to $\int d\mathbf{x} = 1$.

$$Z = \int_1 f(\mathbf{x})\, d\mathbf{x} \;\approx\; \sum f(\mathbf{x})\, \delta\mathbf{x}$$

$$Z = \int_0^1 f(X)\, dX$$

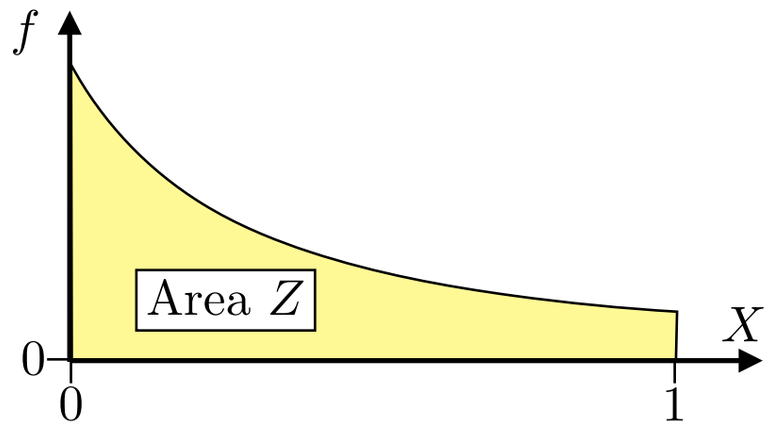
Raster* cells by decreasing value of f.



$=$ 

volume $\mathbf{x} \longleftrightarrow$ linear X

*Impractical but that will not matter!



We need to use numerical evaluations $f_i = f(\mathbf{x}_i)\,,\ i = 1, 2, 3, \ldots, n$

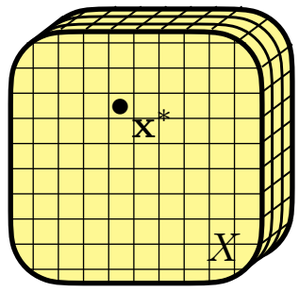
How to choose locations $\mathbf{x}_i$? **Random!** (How else?) "Monte Carlo"

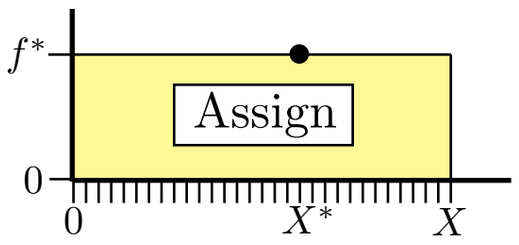
Random in volume $\mathbf{x}$ = Random on linear X.

Assignment — the key operation

Given one sample, value f^*, from $\mathbf{x}^*$ somewhere in volume X,
what is $\int f\, dX$?

Assign f(elsewhere) $= f^*$ (the MaxEnt assignment, what else?)
giving $\int f\, dX = f^* X$.

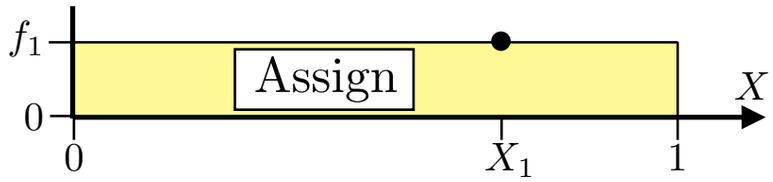




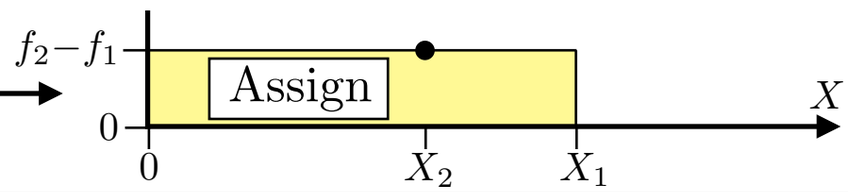
!

Feign deeper ignorance if you want, but you will then be stuck.

Now take $n = 2$ random locations, get values f_1 and f_2,
labelled as $f_1 < f_2$ so that $0 < X_2 < X_1 < 1$.
Assign contribution from f_1 (independent of X_1) as f_1.

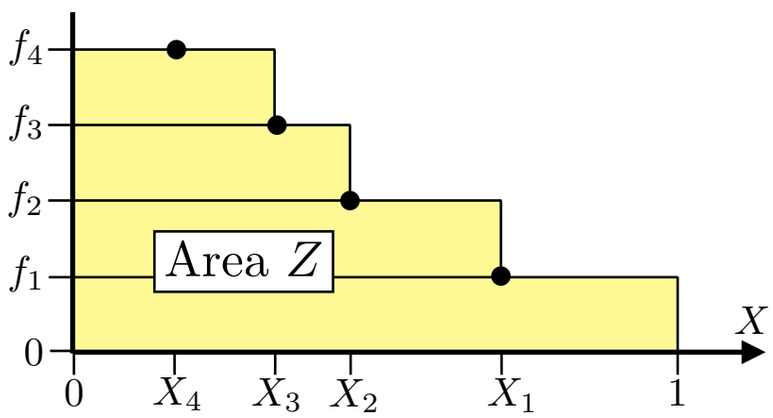


The excess $f_2 - f_1$ is positive from random location X_2 within X_1,
so assign contribution from $f_2 - f_1$ as $(f_2 - f_1)X_1$.



With $n = 4$, make 4 assignments on $0 < X_4 < X_3 < X_2 < X_1 < 1$.

$$Z = f_1 + (f_2 - f_1)X_1 + (f_3 - f_2)X_2 + (f_4 - f_3)X_3$$



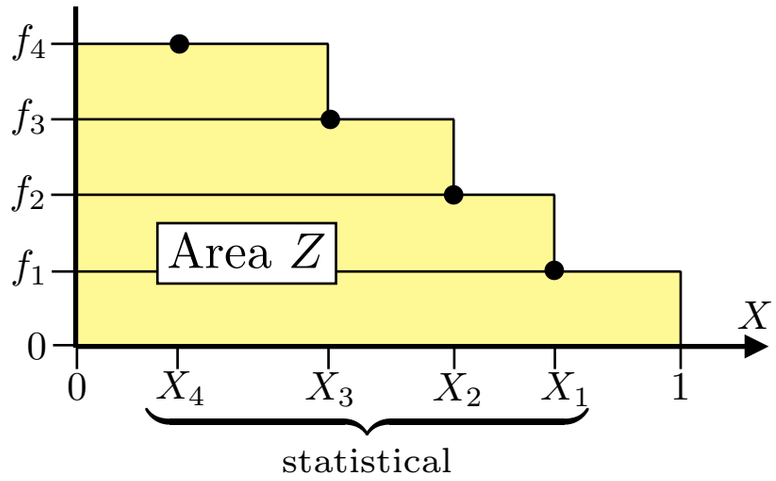
And so on.

With $n = 4$ samples, we seek $\mathbf{X} \sim \texttt{Uniform}(\, 0 < X_4 < X_3 < X_2 < X_1 < 1 \,)$.

Avoid sorting by recursing inward compression factors γ.

$$
\begin{aligned}
X_1 &= \gamma_1\,, & \Pr(\gamma_1) &= 4\gamma_1^3\,, & \gamma_1 &= \textcircled{u}^{1/4} & &\text{(outermost of 4)}\\
X_2 &= \gamma_1\gamma_2\,, & \Pr(\gamma_2) &= 3\gamma_2^2\,, & \gamma_2 &= \textcircled{u}^{1/3} & &\text{(outermost of 3)}\\
X_3 &= \gamma_1\gamma_2\gamma_3\,, & \Pr(\gamma_3) &= 2\gamma_3\,, & \gamma_3 &= \textcircled{u}^{1/2} & &\text{(outermost of 2)}\\
X_4 &= \gamma_1\gamma_2\gamma_3\gamma_4\,, & \Pr(\gamma_4) &= 1\,, & \gamma_4 &= \textcircled{u} & &\text{(outermost of 1)}
\end{aligned}
$$

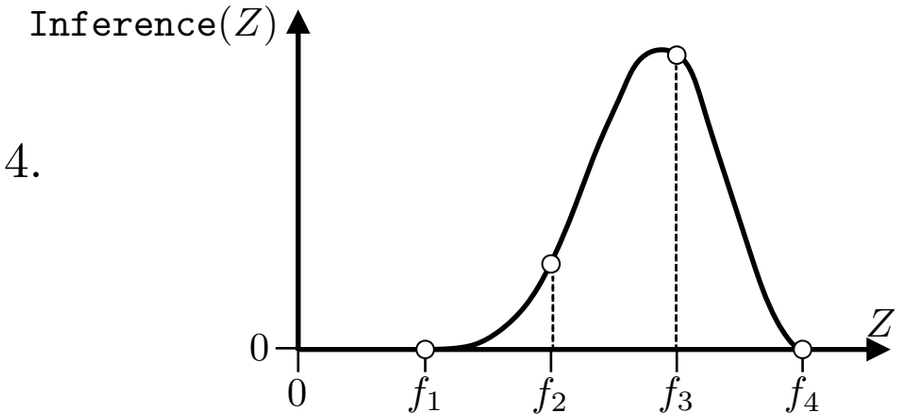
$$\textcircled{u} \equiv \texttt{Uniform}(0,1)$$



Infer statistical estimate $\Pr(Z \mid \underbrace{f_1, f_2, f_3, f_4}_{\text{data}}) = \texttt{Inference}(Z)\,.$

Example: $f_1 = 1,\ f_2 = 2,\ f_3 = 3,\ f_4 = 4.$

$$\langle Z \rangle = \frac{f_1 + f_2 + f_3 + 2f_4}{5}$$



$Z = 2.80 \pm 0.48$
mean $\pm$ stddev

This is asymmetric:

Why?

— because the f's are bounded below by 0 but *not above*.

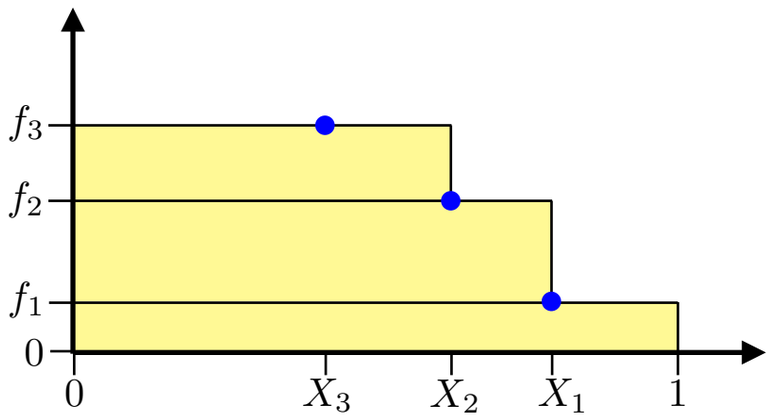
There is no up/down symmetry to force a symmetric distribution, so f_4 is anomalous.

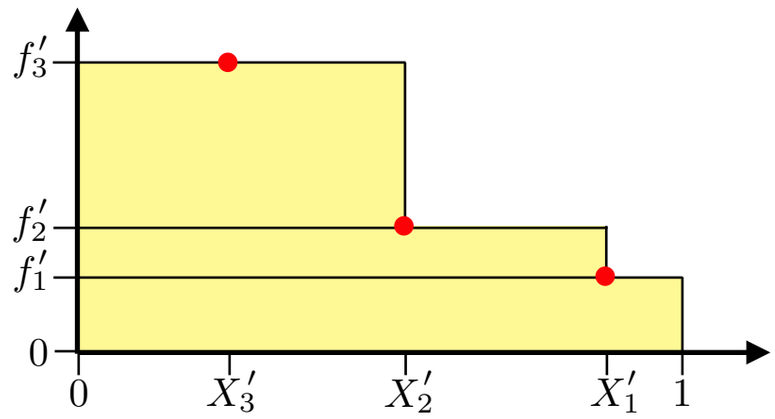
Conversely, f_1, f_2, f_3 are equivalent because they are bounded below *and* above.

Our assignments are *logically defensible. Which would you change?*

Using several runs

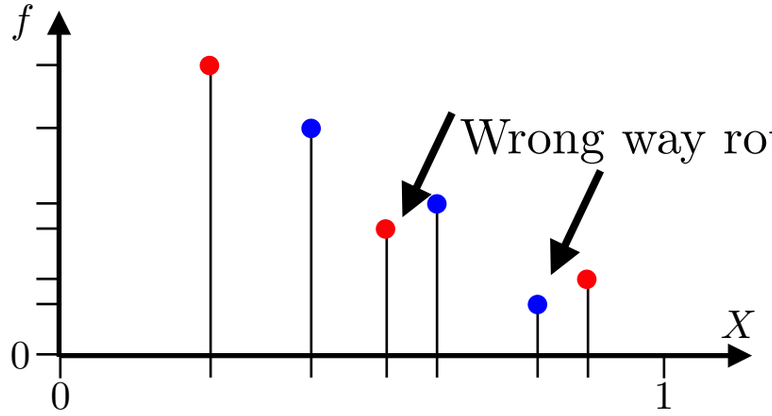
Different sampling of f and different simulation of X.



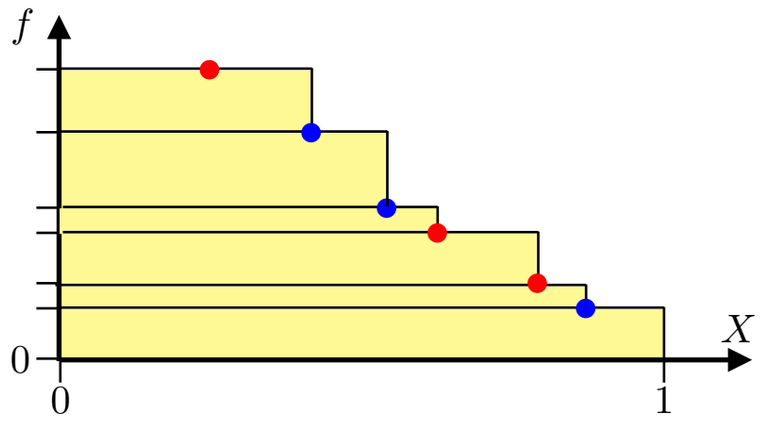


.....

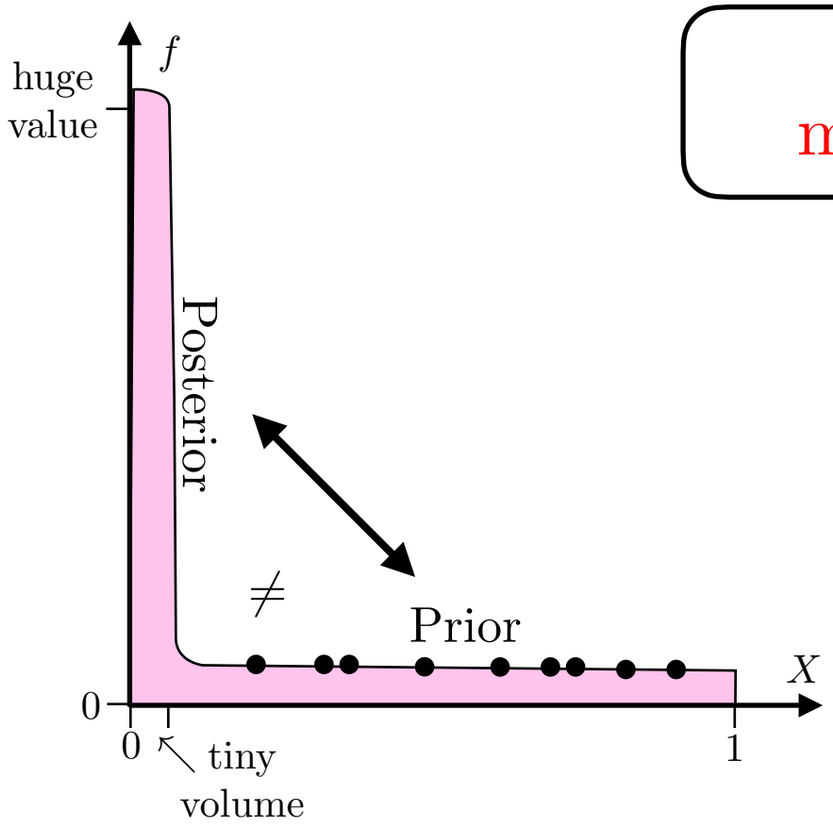
Ordering becomes conflicted between separate runs.



The f's define how the X's should interleave, so *merge* the runs before simulating the X's.



There is only one special "top" f value, not several.



Size matters

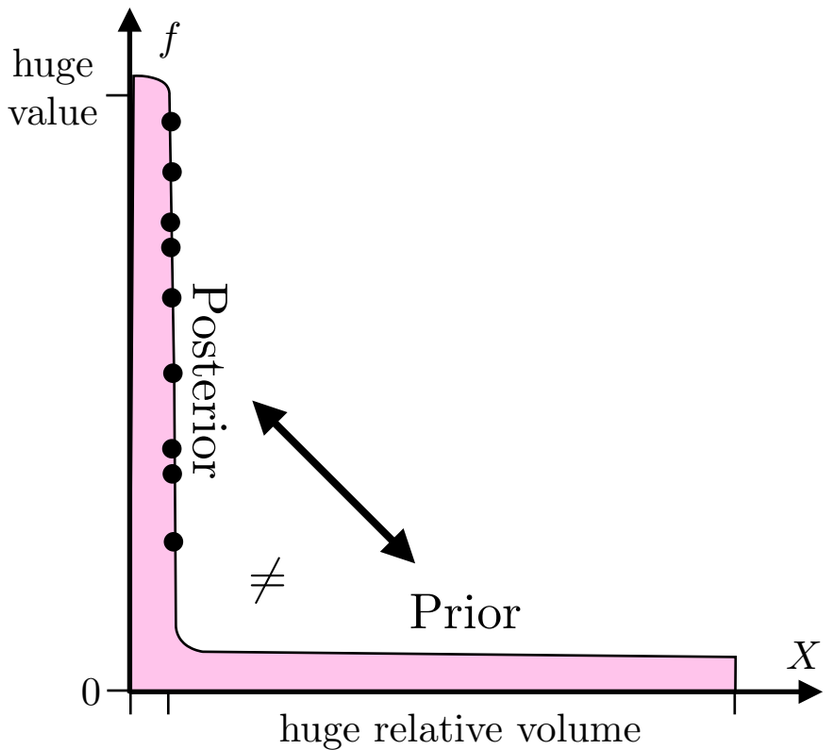
Monte Carlo

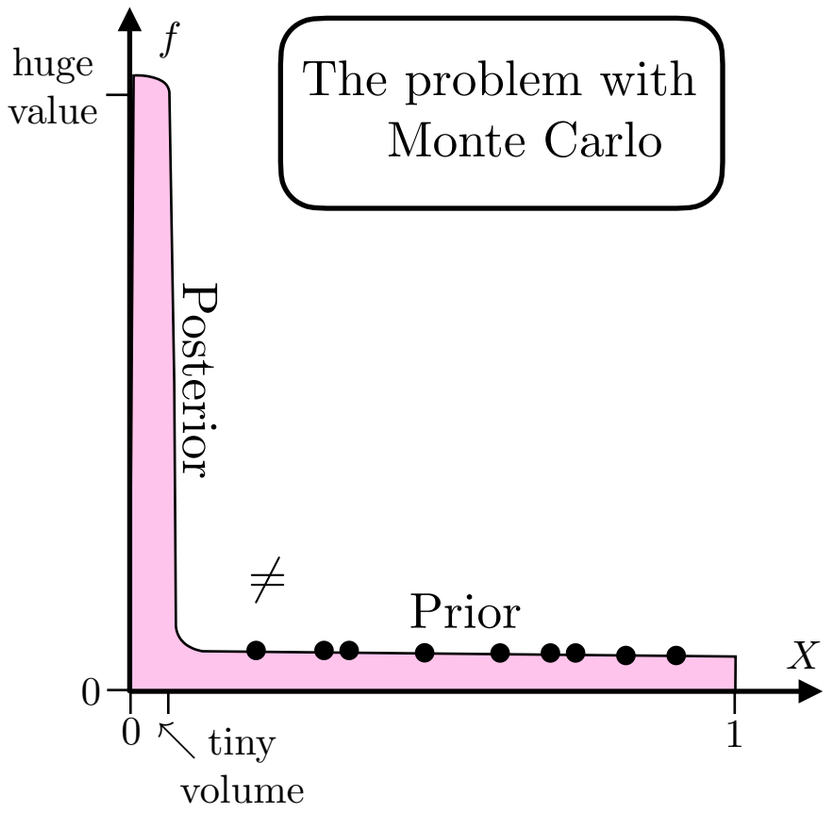
Prior $\quad Z = \left\langle f \right\rangle_{\text{prior}} = \infty \times 0$

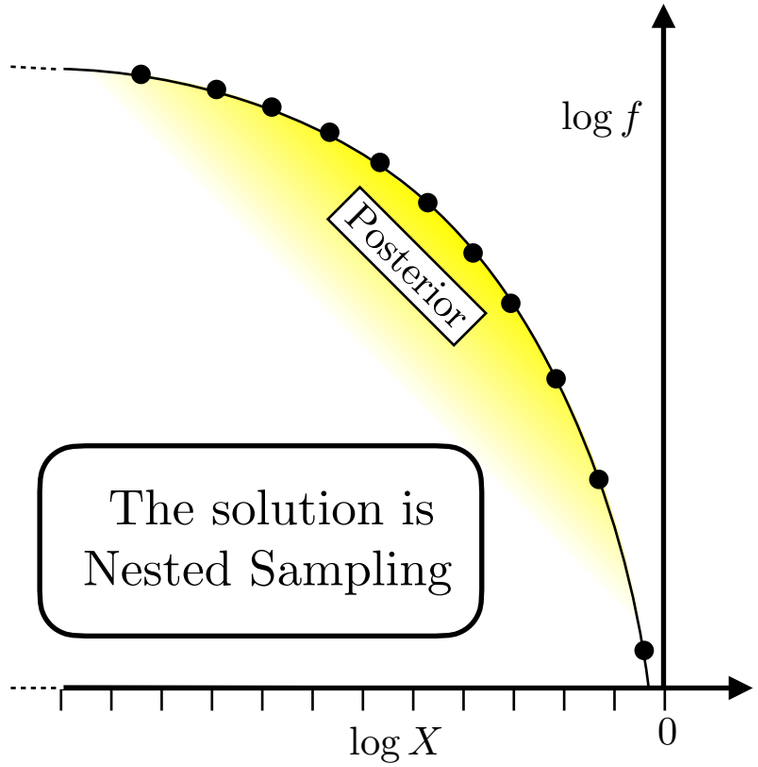
exp(data size)

Posterior $\quad \dfrac{1}{Z} = \left\langle \dfrac{1}{f} \right\rangle_{\text{posterior}} = \dfrac{\infty}{\infty}$

Harmonic mean







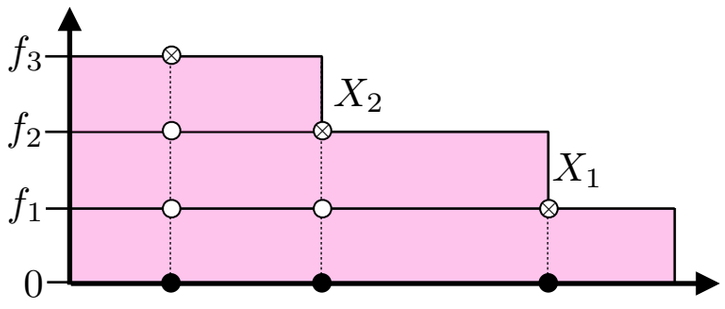
Keep the ensemble populated by resampling $\mathbf{x}$ in $f(\mathbf{x}) \geq f^*$.

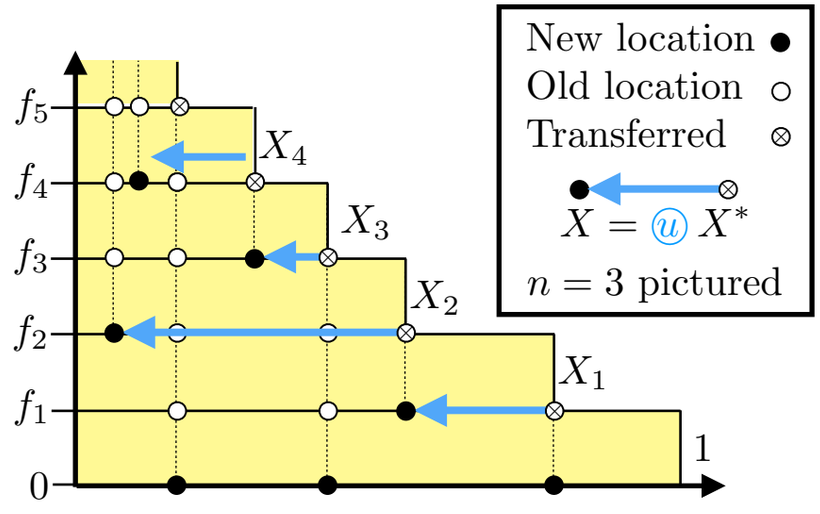
$$\left.\begin{aligned} X_1 &= \gamma_1 \,, \\ X_2 &= \gamma_1\gamma_2 \,, \\ X_3 &= \gamma_1\gamma_2\gamma_3 \,, \\ &\ldots \end{aligned}\right\}$$

$\Pr(\gamma) = n\gamma^{n-1}\,, \quad \gamma = ⓤ^{1/n}$ (outermost of n)

$ⓤ \equiv \texttt{Uniform}(0,1)$

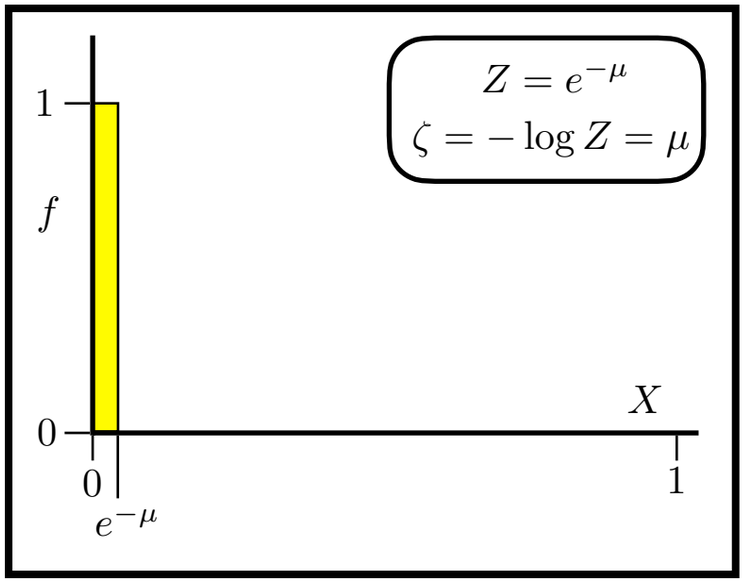
Geometrical compression, k steps reach $X \sim e^{-k/n}$.

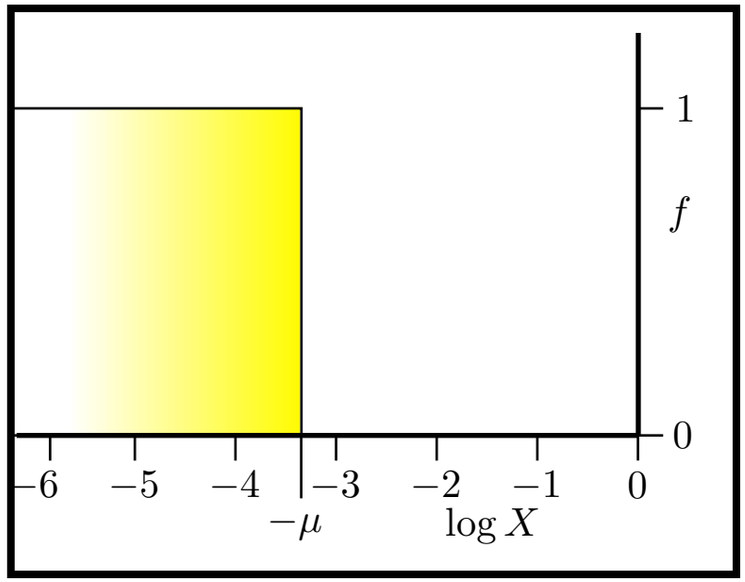




The basic example

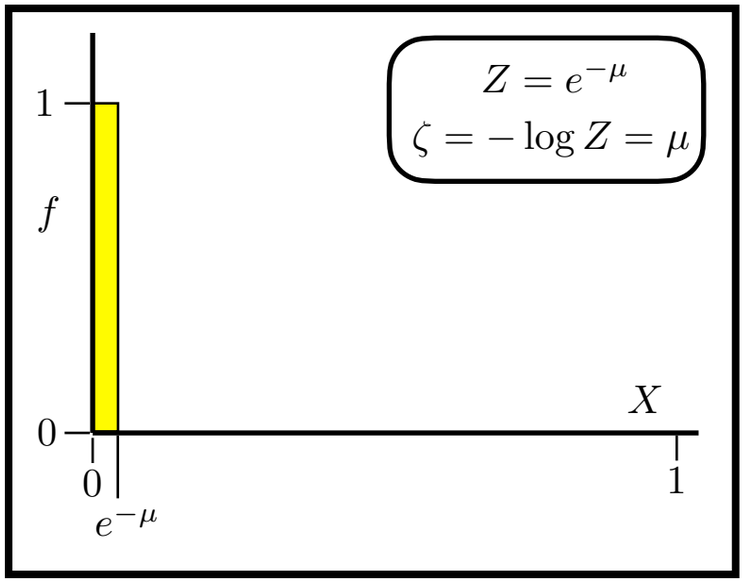
$$f(X) = \left\{ \begin{array}{ll} 1 & \text{for } X < e^{-\mu} \\ 0 & \text{otherwise} \end{array} \right\} \text{ with tiny downward slope to make } f \searrow.$$

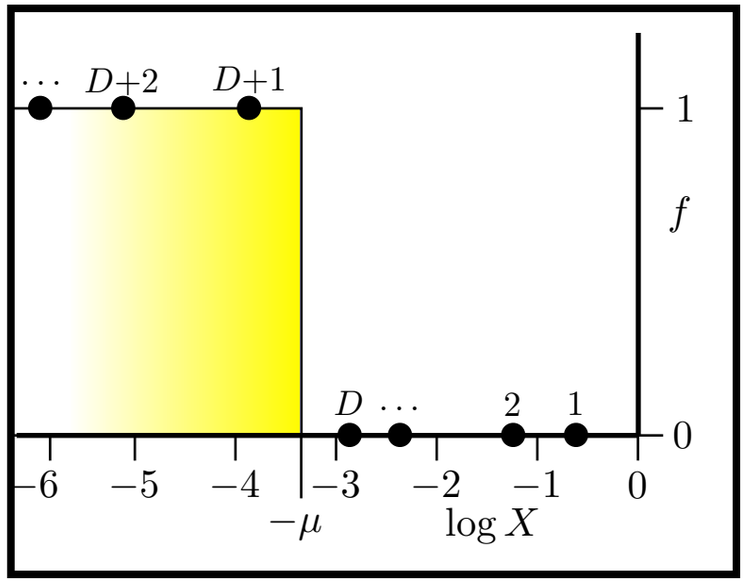




The basic example

$$f(X) = \left\{ \begin{array}{ll} 1 & \text{for } X < e^{-\mu} \\ 0 & \text{otherwise} \end{array} \right\} \text{ with tiny downward slope to make } f \searrow.$$





Run with $n = 1$.
Get dataset $\{\underbrace{0, 0, \ldots, 0}_{D \text{ zeros}}, \underbrace{1, 1, 1, 1, \ldots}_{\text{all 1's}}\}$.

Compressions are $\gamma \sim \texttt{Uniform}(0, 1)$, so $\log \gamma \sim -\texttt{Exponential}(1)$

so $\texttt{Frequency}(D \mid \mu) = e^{-\mu}\mu^D/D!$ [Poisson] $[\, D = \mu \pm \sqrt{\mu}\,]$

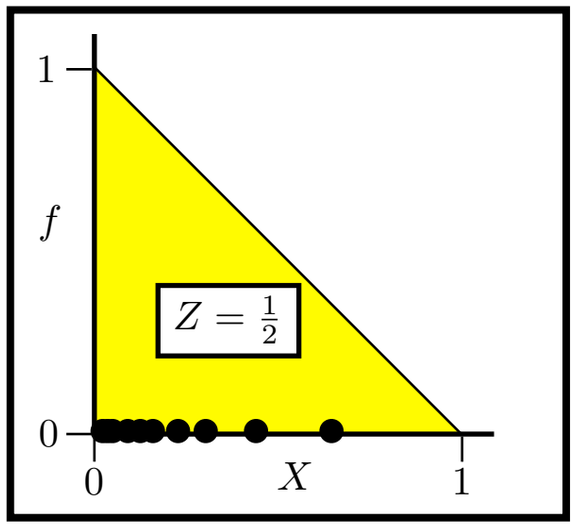
Given D, inference of $\zeta = -\log Z$ where $Z = f_1 + (f_2{-}f_1)X_1 + (f_3{-}f_2)X_2 + \cdots = X_D = \gamma_1\gamma_2 \ldots \gamma_D$

is $\texttt{Inference}(\zeta \mid D) = e^{-\zeta}\zeta^{D-1}/(D{-}1)!$ [Gamma] $[\, \zeta = D \pm \sqrt{D}\,]$

Expectation recovery of μ is $\langle \Pr(\zeta \mid \mu) \rangle = \sum_{D=0}^{\infty} \texttt{Inference}(\zeta \mid D)\, \texttt{Frequency}(D \mid \mu)$ $[\, \zeta = \mu \pm \sqrt{2\mu}\,]$

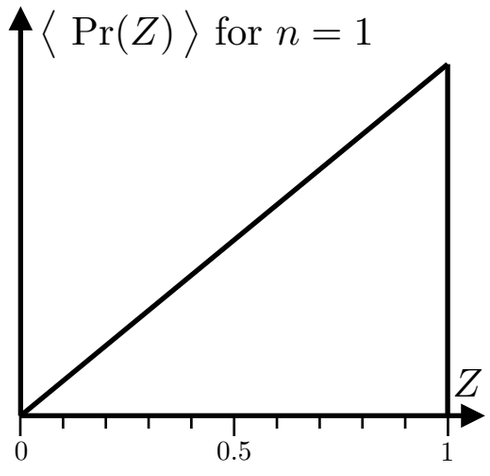
A second example

$f(X) = 1 - X$

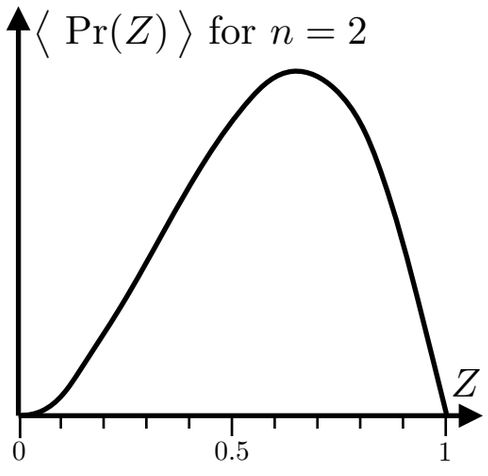


Run with different ensemble sizes n.

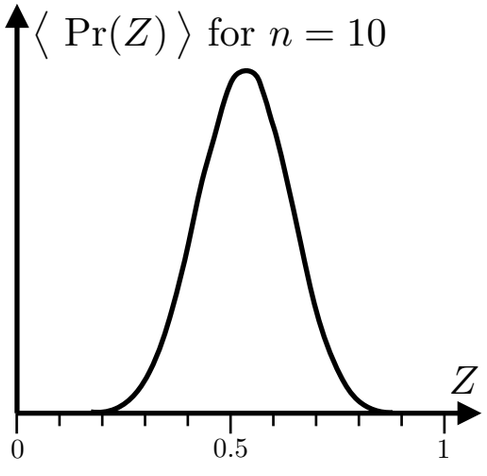
Expectation recoveries of Z over many runs:



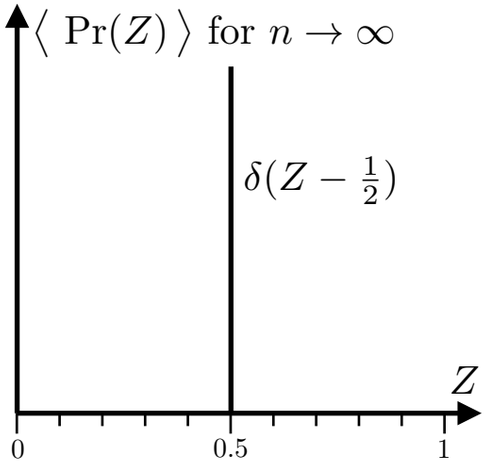
$\langle Z \rangle = 0.6667 \pm 0.2357$



$\langle Z \rangle = 0.6 \pm 0.2$



$\langle Z \rangle = 0.5239 \pm 0.1064$



$\langle Z \rangle = 0.5$

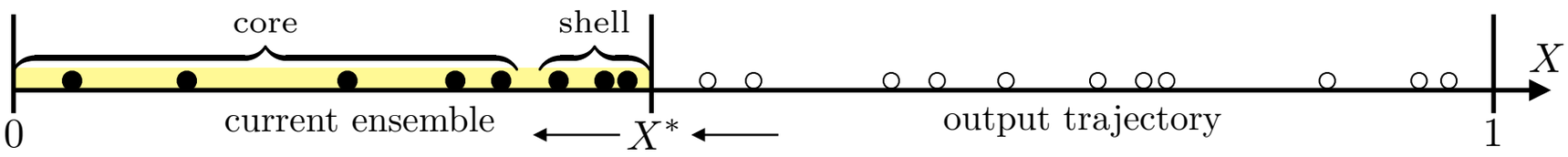
Do not average coarse runs ····· merge them or (more robustly) run a large ensemble directly.

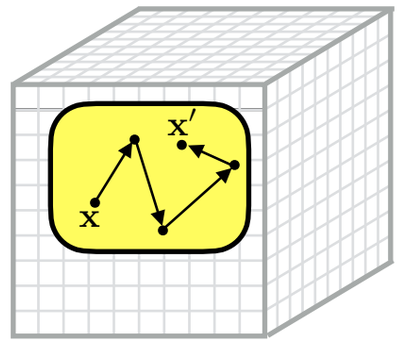
You only ever have one dataset!

Programming

`Nested Sampling` uses n locations in $f(\mathbf{x}) \geq f^*$ $\left\{ \begin{array}{l} c = \text{ number in core, } f > f^* \\ s = \text{ number in shell, } f = f^* \end{array} \right\}$ $n = c+s$

Force up in f by removing shells while keeping the ensemble populated.





You supply procedure to get $\mathbf{x}' = \texttt{Explore}(\mathbf{x})$ randomly in $f(\mathbf{x}') \geq f^*$.

You write this.

Begin with n random $\mathbf{x}$'s with their values $f(\mathbf{x})$, and set $f^* = 0$.

Iterate:

Divide ensemble into core c and shell s.

You control this → While you want more members (usually because c is smaller than you want) ...

Take any $\mathbf{x}$ in $f(\mathbf{x}) \geq f^*$ (suggest random choice from current ensemble).

Get new $\mathbf{x}' = \texttt{Explore}(\mathbf{x})$ and its $f(\mathbf{x}')$.

Add $\mathbf{x}'$ to ensemble and increment either c (if $f' > f^*$) or s (if $f' = f^*$).

While shell remains populated (it may have several members if f has a plateau) ...

Transfer shell members to the output "trajectory" and decrement s.

Monitor progress (*next slide*).

Compress by increasing f^* to the next lowest f.

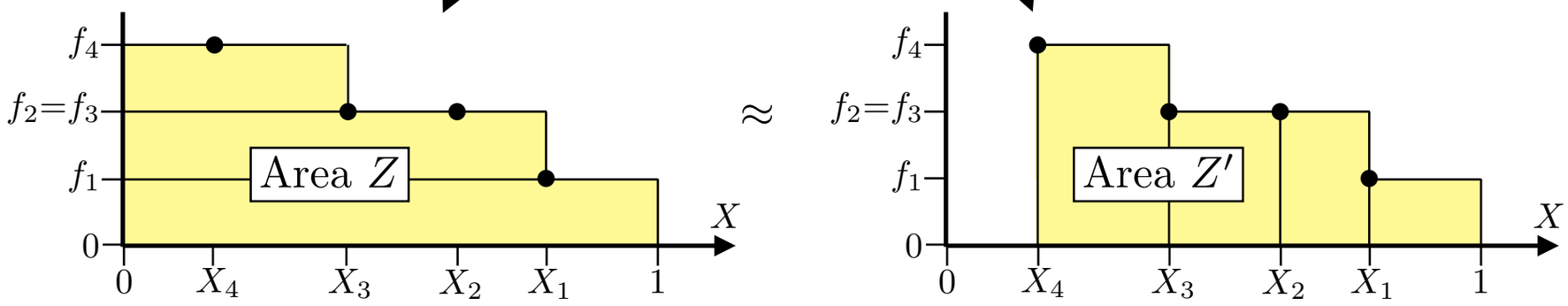
End: empty the ensemble by iterating without generating new members.

Get Quantity $Z = \int f\, dX$ and Information $H = \int \frac{f}{Z} \log \frac{f}{Z}\, dX$

Terminate when Z stops increasing, or (better?) when H saturates as learning stops.

Progress updates

Begin	··· Iterate ···	End
$X_0 = 1$	$X_i = ⓤ^{1/n_i} X_{i-1}$	$X = 0$
$Z_0 = 0$	$Z_i = Z_{i-1} + X_{i-1}(f_i - f_{i-1})$	$\sum X\, \delta f = Z$
$Z'_0 = 0$	$Z'_i = Z'_{i-1} + (X_{i-1} - X_i) f_i$	$\sum f\, \delta X \approx Z$



Z decomposes by value.
Use for termination.

Z' decomposes by volume.
Use for applications.

The compression factors actually were $ⓤ^{1/n}$ for unknown ⓤ, which we simulate with ν samples from $\texttt{Uniform}(0, 1)$ of what ⓤ might have been.

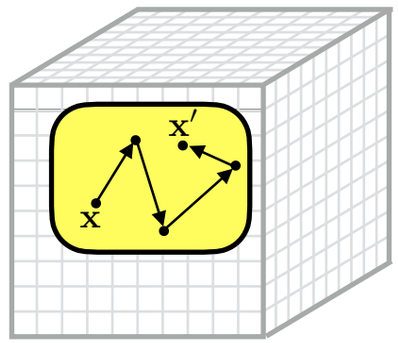
These ν simulated trajectories yield the range of plausible results.

Each record on the trajectory has:

- $\mathbf{x}$ = location
- f = value $f(\mathbf{x})$
- n = ensemble size of which $\mathbf{x}$ was the outermost
- $ⓤ_1 \ldots ⓤ_\nu$ = simulated compression controllers from $\texttt{Uniform}(0, 1)$
- $ⓤ_0$ = central geometric-mean $1/e$ for central trajectory, used to terminate when Z or (better H) saturates.

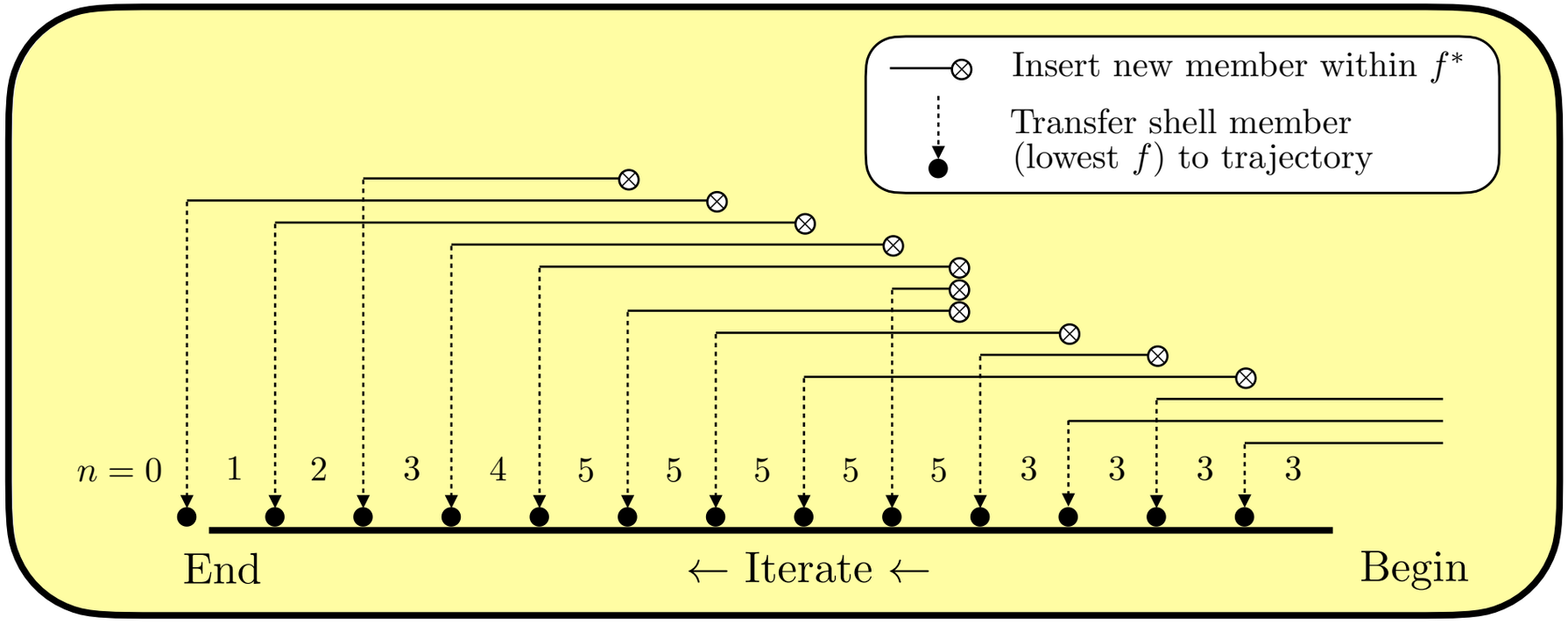
Control

New member is $\mathbf{x}' = \texttt{Explore}(\mathbf{x})$ uniform in $f(\mathbf{x}') \geq f^*$.

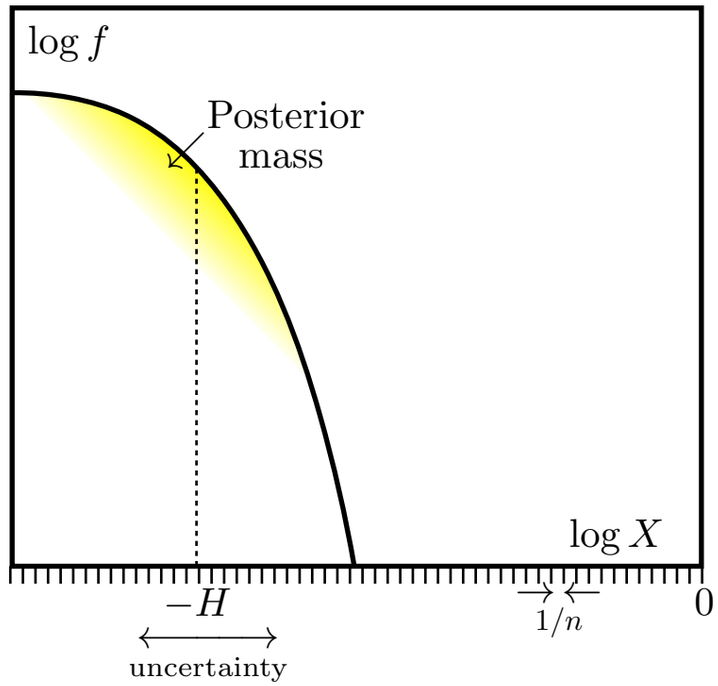


Upon removing a shell (occupancy s, usually 1), insert $\begin{cases} > s & \text{new members to grow ensemble;} \\ = s & \text{new members to preserve membership } n; \\ < s & \text{new members to erode ensemble.} \end{cases}$

You control this



Uncertainty

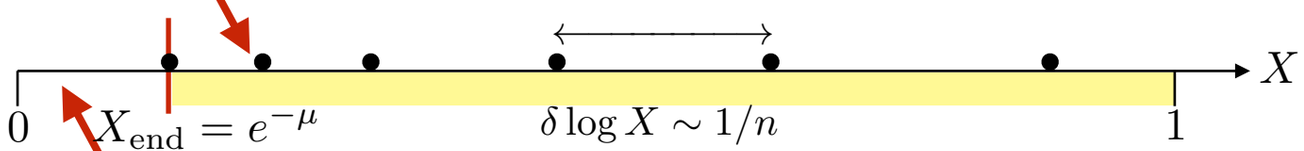


The posterior occupies a volume $X \sim e^{-H}$ (definition of information).
With ensemble size n, this is reached in about $nH \pm \sqrt{nH}$ steps (Poisson).
The uncertainty yields $\delta(\log X) \sim \sqrt{H/n}$, hence $\delta(\log Z) \sim \sqrt{H/n}$ also.

Convergence

Integral $\displaystyle\int_{X_{\text{end}}}^{1} f(X)\,dX$ is defined as $\displaystyle\lim_{\substack{\delta X \to 0 \\ \text{in any way}}} \sum f(X_i)\,\delta X_i$

"In any way" includes geometrical compression with large n and, with probability 1, statistical generation and recovery of such compression (which is nested sampling).



Missing termination proportion is bounded by the (always limited) information content H^* from f.

$$\frac{\Delta Z}{Z} \equiv \frac{1}{Z}\int_{0}^{X_{\text{end}}} f(X)\,dX < \frac{H^*}{\mu}$$ which $\to 0$ as μ increases with iteration count.

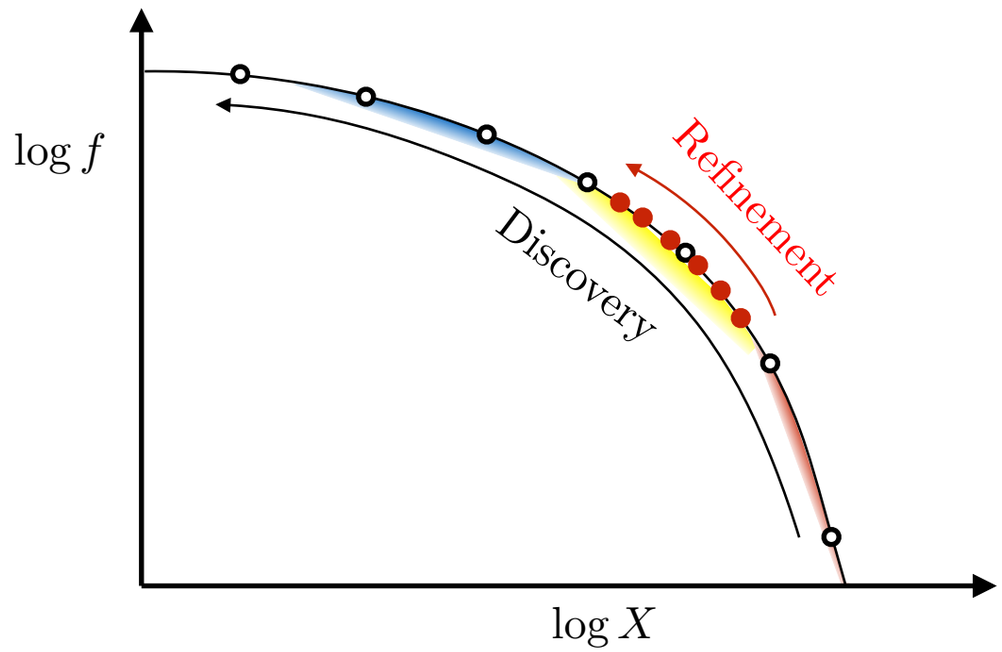
$$\lim_{\substack{\text{large ensemble} \\ \text{large compression}}} (\text{inferred } Z) = \text{true } Z$$

QED ✓

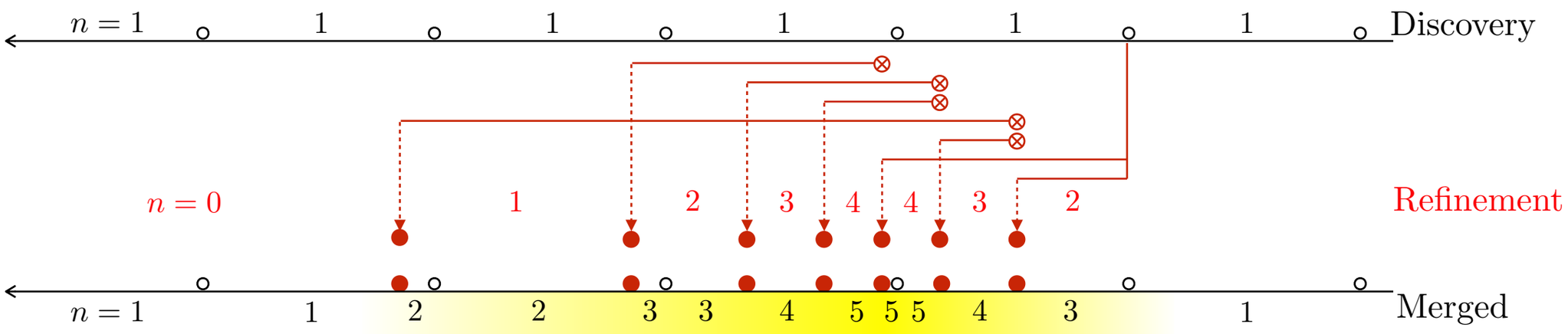
Refinement

Trajectory gives posterior distribution $\Pr(\mathbf{x})$ as a weighted sum $\sum P_i\,\delta(\mathbf{x}-\mathbf{x}_i)$ over locations.
Hence estimate distribution $\Pr\left(\int \phi(\mathbf{x})\,d\mathbf{x}\right) = \sum P_i\phi(\mathbf{x}_i)$ of arbitrary quantity.
To get numerical uncertainty in this, use the ν simulations of what the compressions actually were.

To improve numerical accuracy, get finer sampling of range of interest by sourcing another ensemble.



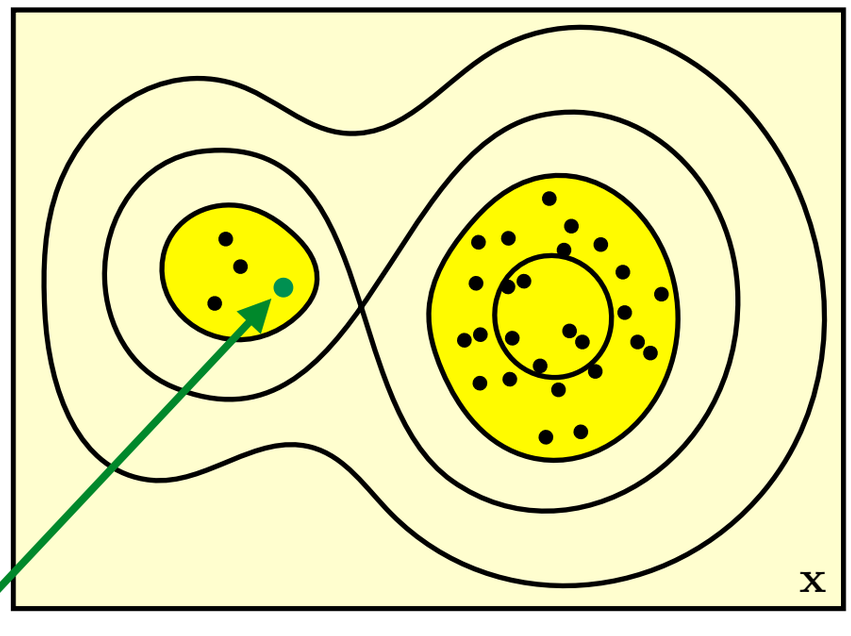
Aim to add enough refinement to get roughly uniform decomposition $f\,\delta X$ of the posterior mass.

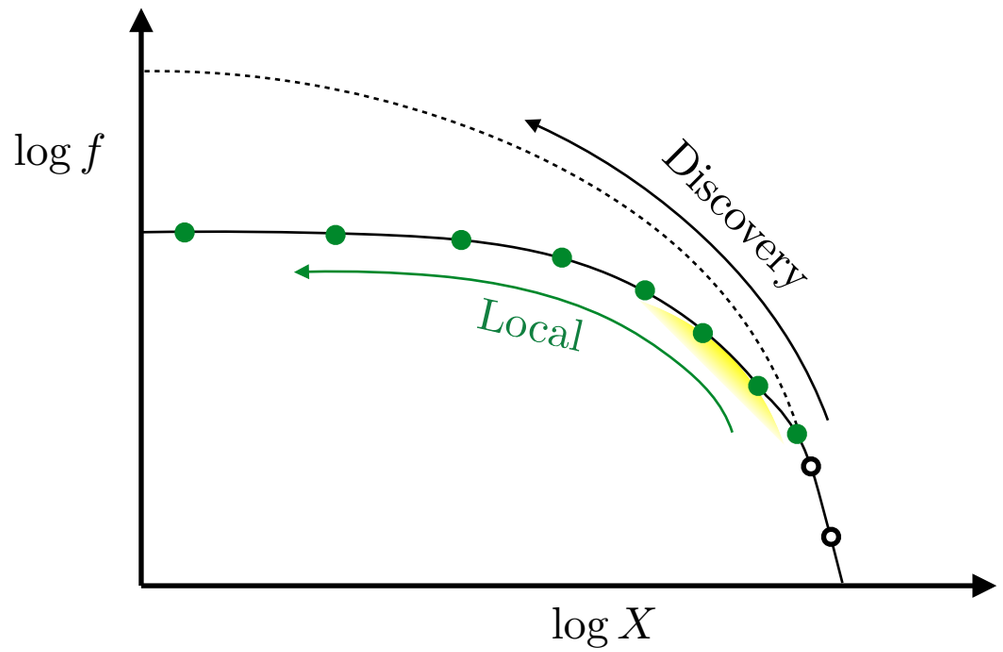


Add the n's in each merged interval, with intervals compressing by about $\exp(-1/n)$.

Multimodality

I define a *separate mode* as an island of locations which your exploration technique (confined by the value constraint) is (at least in practice) *unable to communicate with.*





You suspect that this location is in a separate mode.

Source an ensemble from there to explore the supposedly isolated island.

Get the island's evidence, as compared with the full system, and its posterior distribution.

Applications

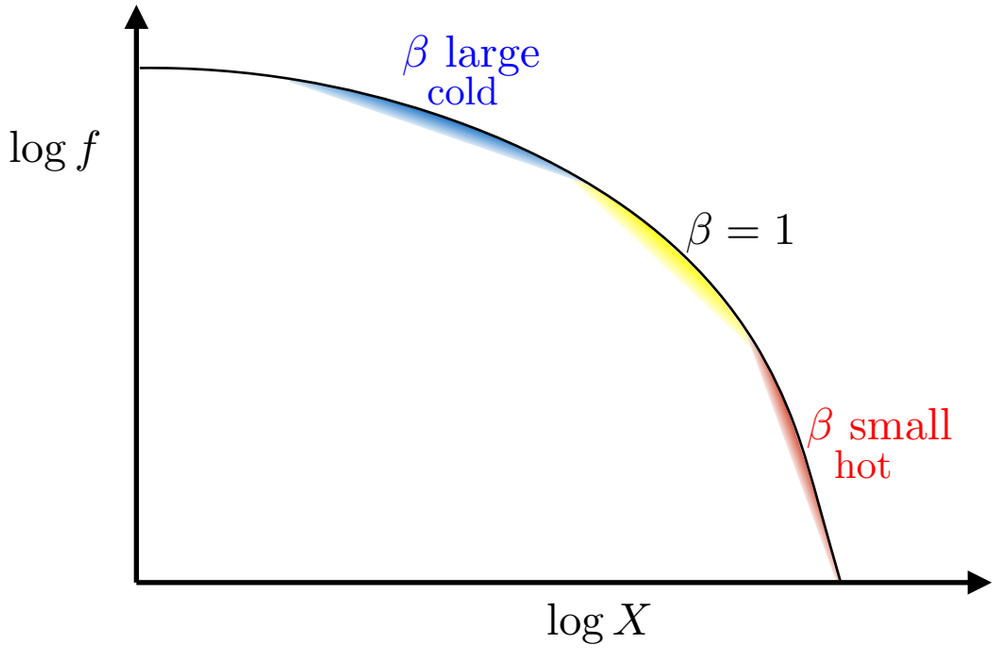
		Quantification	Statistics	Statistical physics
In	f_i	Density	Likelihood	Boltzmann factor
	δX_i	δ volume	δ Prior	number microstates
Out	$w_i = f_i\,\delta X_i$	δ mass	δ Joint	Boltzmann factors
	$Z = \sum f_i\,\delta X_i$	Mass	Evidence	Partition function
	$P_i = w_i/Z$	δ proportion	δ Posterior	δ probability
	$H = \sum P_i \log P_i$	Information	Kullback-Leibler	$-$Entropy

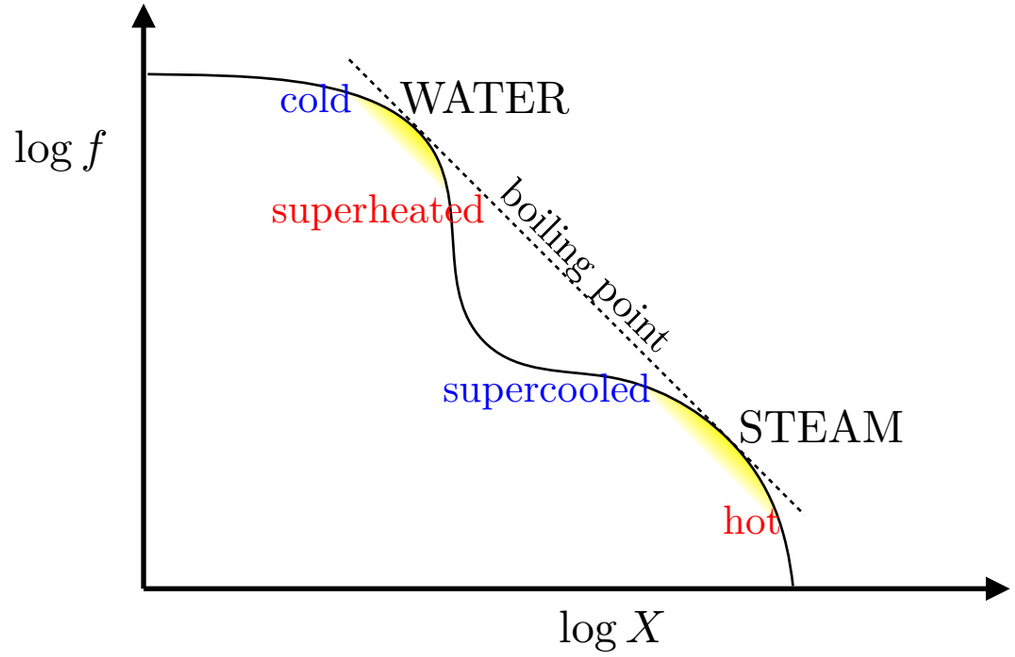
The same trajectory is obtained for *any* monotonic function of f, in particular f^β.

So Z generalises to $Z(\beta) = \sum f_i^\beta\,\delta X_i$ at no cost, smoothly because the δX's are fixed.

In statistical physics, $f^\beta = e^{-\beta E}$ where E = energy (= $-$logLikelihood) and $\beta = 1/kT$ = coolness.

We can plot $Z(\beta)$ and also internal energy $U = -\dfrac{\partial \log Z}{\partial \beta}$ and specific heat $C = \dfrac{\partial U}{\partial \beta}$ as smooth functions.





And we can deal with phase changes, where U has a discontinuity (latent heat) and C goes singular. (Annealing fails.)

That's it! That's how to add up!

I am a *craftsman* — I make tools, and nested sampling is my finest.

Good workmen know their tools.

Enjoy!

John Skilling, Auckland, December 2025