Particle Distributions & Particle Filters

In general in computer science RANDOMNESS does not exist → all deterministic!

↓

BUT, programs to generate samples from uniform distribution (uniform between 0 and 1)

↳ state machine ⟹ initialized with a seed → by altering the initial state/seed, we can decide how to play the random machine

↳ example : <stdlib.h> → drand48()

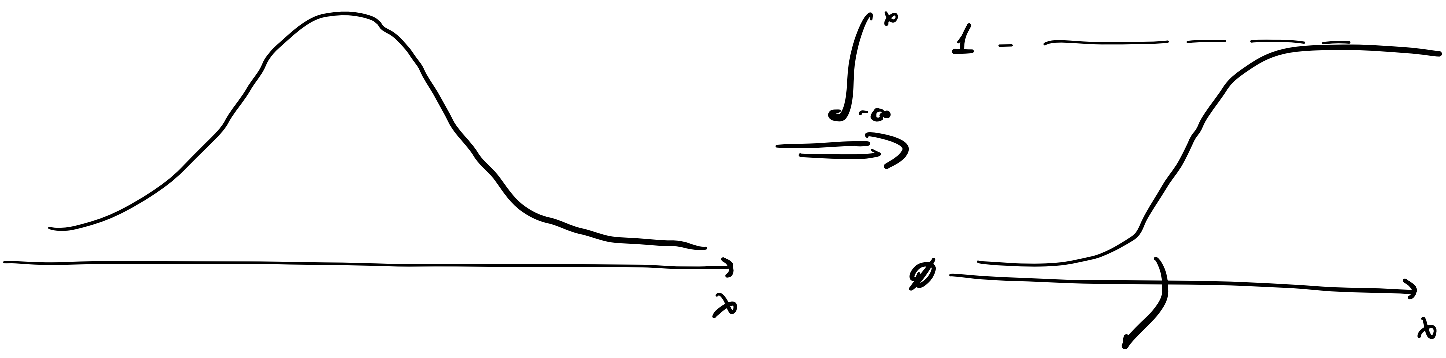
⇓

our goal: draw random samples from this distribution!

Cumulative density : $P(x) = \int_{-\infty}^{x} p(x')\,dx'$

|||

- monotonic growing function
- probability of an event that a certain sample falls in the event space between $[-\infty, x]$

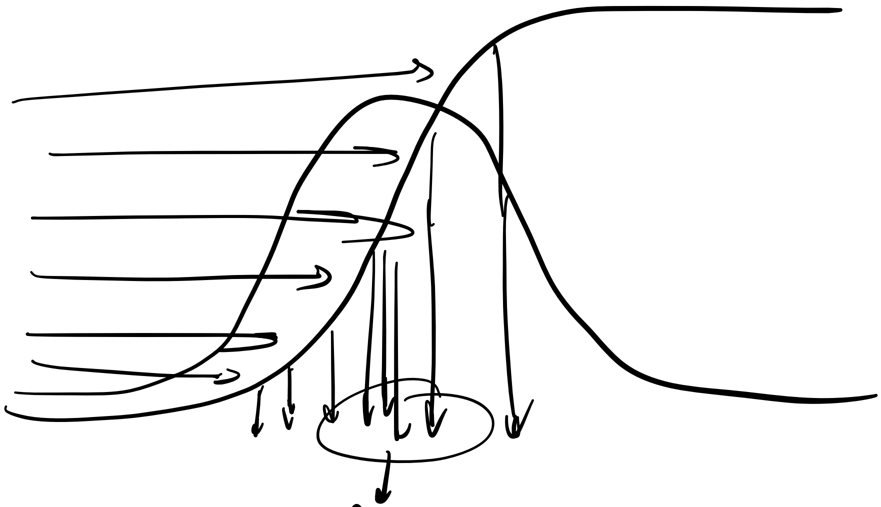


$y^{(i)} \sim U(0,1) \longrightarrow x^{(i)} = P^{-1}(y^{(i)})$

↓

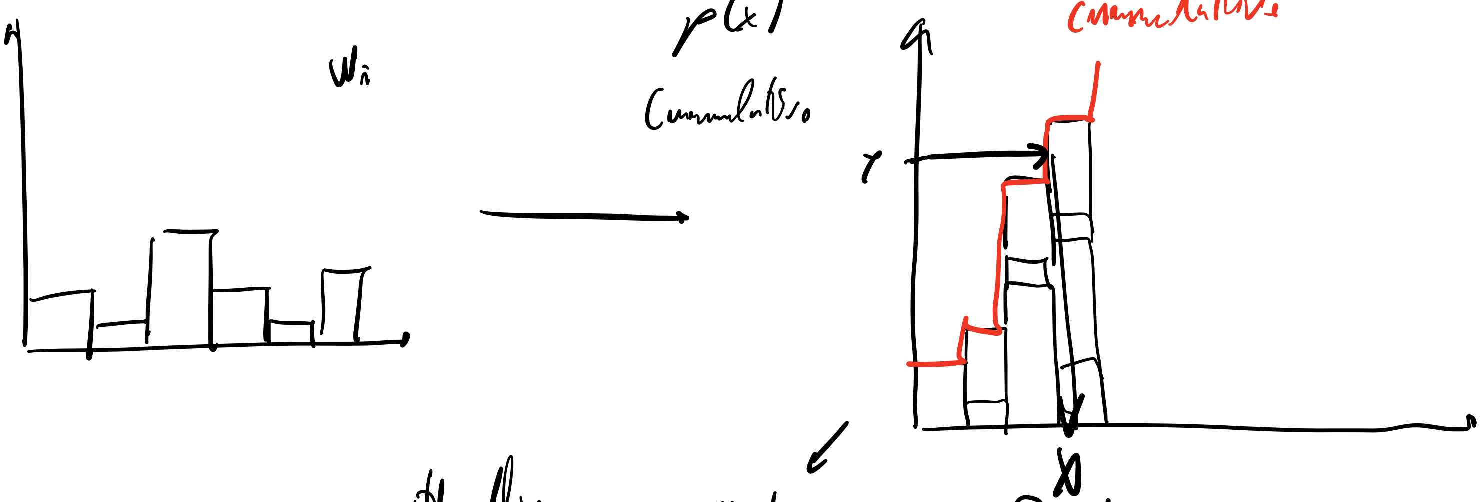
uniform distribution

$\uparrow \dfrac{dP(x)}{dx} \Rightarrow \uparrow p(x)$



dense mean the higher probability of $p(x)$

Cumulative



with this way, we are at least able to evaluate $P^{-1}(y^{(i)}) \to x^{(i)}$

However,

computation problem: sampling strategy from known distributions

computation ~ millions of samples

⇓

INEFFICIENT!

Uniform Sampling: 1. Generate N samples drawn from discrete distribution at random by invoking 1 / once the random generator

$y^{(0)} \sim U(0, 1/N)$

very first time I initialize the

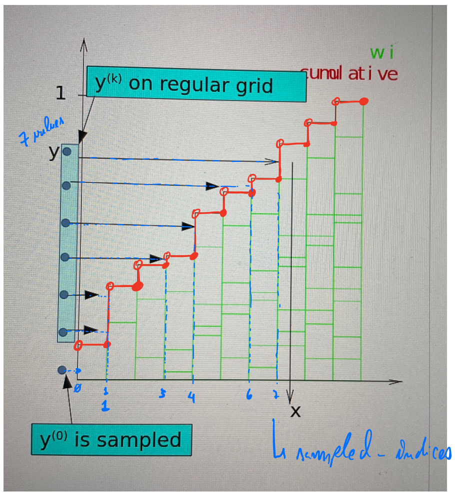
2. generate other element in regular sample:

$y^{(1)} = y^{(0)} + 1/N$

$y^{(2)} = y^{(0)} + 2/N$

....

doing the subsampling, I am only moving forward in reading the cumulative distribution (instead of $\log_2 N$ each time)



(resemble each other)

we want to sample from this, not able to formulate a close form, but able to evaluate it POINT-WISE

1. $\tilde{\pi}(x) \simeq p(x)$ → IMPORTANCE SAMPLING

do not know it in a close equation!

[but through a "complicated" process, we can approximate it ....
$\tilde{\pi}(x)$ supports samples (normal form, gaussian, ...)]

$X^{(i)} \sim \tilde{\pi}(x)$

PROPOSAL DISTRIBUTION

$\tilde{\pi}(x)$ must → be non-zero in the area $p(x)$ is non-zero!

2. Compute weight

$$w^{(i)} = \frac{p(x^{(i)})}{\pi(x^{(i)})}$$

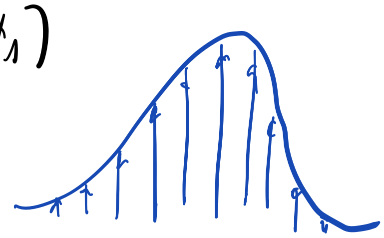
target (numerator) — proposal (denominator)

⇓ goal is to entirely cover the full domain of the target distribution

↳ if proposal $\pi(x^{(i)}) = p(x^{(i)})$ ⇒ ALL THE WEIGHTS WOULD BE UNIFORM !

Resampling

L samples ⊕ weights = discrete distribution (w/ normalized weights)

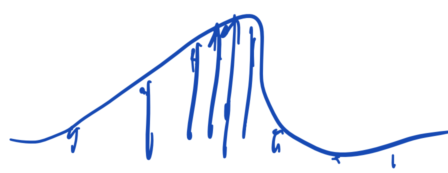
$\{\langle w^{(i)}, x^{(i)}\rangle\}$ ≡ 

We may be over-representing irrelevant samples

⇓

RESAMPLING

⇓



$\{j^{(i)}\} = \text{uniformSample}(\{w^{(i)}\})$

$\tilde{x}^{(i)} = x^{(j^{(i)})}$

↑ new sample in position i is the old sample in position j[i]

Particle filter → $\boxed{\text{particle densities!}}$ → ANOTHER WAY TO REPRESENT PROBABILITY DISTRIBUTIONS

(each sample ||| discrete event) — set of samples from dist → does not restrict the distribution to a Gaussian e.g.

$\Downarrow$

$$x^{[i]} \sim p(x)$$

$$p(x) = \sum_i w^{(i)} \delta(x - x^{(i)})$$

$$\int_E p(x)\,dx = \sum_{x^{(i)} \in E} w^{(i)}$$

probability that x falls in region E can be modeled by summing the weights in the region

Chain Rule

for each of x_a samples, put on $x_b \mid x_a$ to produce new samples

$\Downarrow$

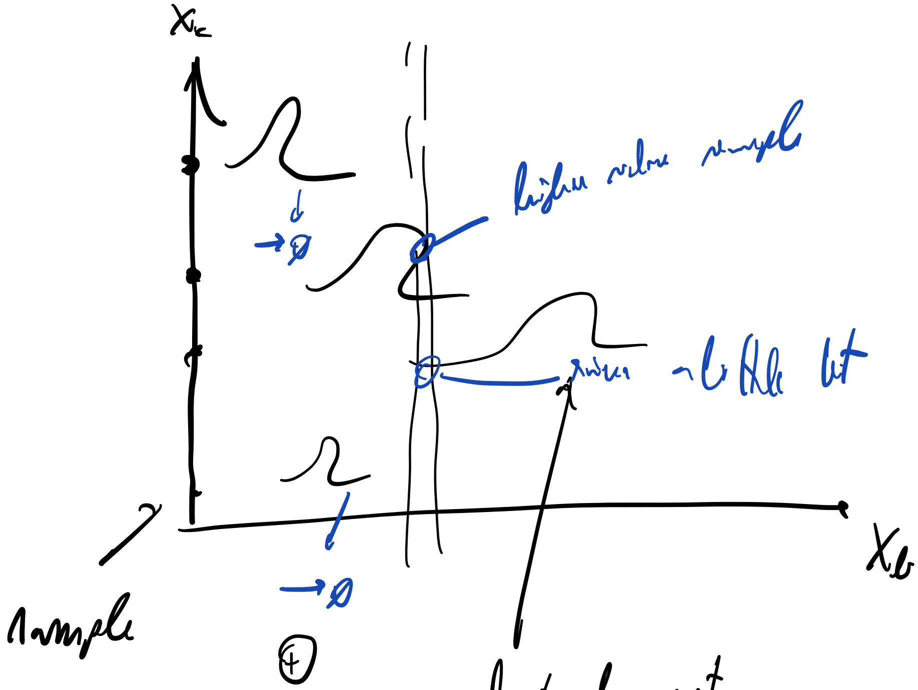
$$p(x_a \mid x_b) = \frac{p(x_a, x_b)}{p(x_b)}$$

CONDITIONING $\sim$ do not like to be sliced

(similar to sigma points) $\Longrightarrow$ SAMPLES ARE A SPARSE REPRESENTATION

$\Downarrow$

RELAX THE PROBLEM



analytical part

$x_b \mid x_a^{(i)}$

$$p(x_a, x_b) \simeq \sum_i w^{(i)} \delta(x_a - x_a^{(i)}) \, p(x_b \mid x_a^{(i)})$$

samples

analytical part (assuming we are able to sampling the distribution $x_b \mid x_a^{(i)}$)

Note:

$$\int \left[\delta(x_a - x_a^{(i)}) \, dx_a \right] = \underline{1}$$

because it's a delta!

$$\left(\int \delta(\cdot) = 1 \right)$$

NOTE: $p(x_b \mid x_a^{(i)})$ is only a number ⊕ η is only a normalizer

$\Downarrow$

CONDITIONING ONLY AFFECTS THE WEIGHTS

$$w_{a|b}^{(i)} \propto w_a^{(i)} \cdot p(x_b \mid x_a^{(i)})$$

$$\boxed{\text{PARTICLE FILTER}}$$

$x_t = f(x_{t-1}, u_{t-1})$ ⟶ know the transition function

⊕

$p(x) \simeq \sum_i w^{(i)} \delta(x - x^{(i)})$ ⟶ assume that the state probability density distribution is represented by a set of samples

⊕

$u_t \sim p(u_t)$ ⟶ additive noise $\sim p(u_t)$ that we can sample from

⊕

$p(z_t \mid x_t)$ ⟶ we can evaluate POINT-WISE / sample from the observation model!

↳ PREDICTION

↳ UPDATE ⟶ Resample a new generation to focus computation on likely regions of the state space

(SUPPRESS NON MEANINGFUL SAMPLES)