Lecture 18: Langevin Dynamics and Diffusion models.

Last class: sampling over discrete domains.

This class: sampling over continuous domains.

# of pixels

ex. distributions over images are more naturally thought of as distributions over $\mathbb{R}^d$

(Super) crash course on continuous probability.

discrete distribution over $\{1, \dots, n\}$ is specified by probability mass function

$p(i) = \Pr_{X \sim p}[X = i]$. An event $E \subseteq \{1, \dots, n\}$ has probability

$p(i) \geq 0, \quad \sum_{i=1}^{n} p(i) = 1.$

$\Pr_{X \sim p}[X \in E] = \sum p(i).$

A continuous distribution over $\mathbb{R}$ (or $\mathbb{R}^d$) is specified by a probability density function (pdf)

$p : \mathbb{R}^d \to \mathbb{R}_{\geq 0}.$

$p(x) \geq 0, \quad \int p(x)\,dx = 1.$

For any event $E \subseteq \mathbb{R}^d$, we have

$\Pr_{X \sim p}[X \in E] = \int_E p(x)\,dx.$

↳ usually doesn't make much sense to talk about

$\Pr_{X \sim p}[X = x]$ ← will be 0

For any function $f : [n] \to \mathbb{R}$, its expectation is

$$\mathbb{E}_{X \sim p}[f(X)] = \sum_{i=1}^{n} f(i)\,p(i)$$

For any function $f : \mathbb{R}^d \to \mathbb{R}$, its expectation is

$$\mathbb{E}_{X \sim p}[f(X)] = \int_{\mathbb{R}^d} f(x)\,p(x)\,dx$$

What are "nice" distributions over continuous domains?

Stereotypical: Gaussian distribution.

In 1-D:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

mean μ $\quad \mathbb{E}[X] = \mu$

variance σ^2 $\quad \mathbb{E}[(X-\mu)^2] = \sigma^2.$

In d dimensions:

$$p(x) = \frac{1}{(2\pi)^{k/2}\det(\Sigma)^{1/2}} \exp(-(x-\mu)^T \Sigma (x-\mu)).$$

mean μ: $\mathbb{E}[X] = \mu$

covariance matrix Σ: $\mathbb{E}[(X-\mu)(X-\mu)^T] = \Sigma.$

If $\Sigma = I \rightarrow$ d independent univariate Gaussians.

General $\Sigma \rightarrow$ corresponds to $\Sigma^{1/2} \cdot X$, $X \sim \mathcal{N}(\mu, I)$.

Key point: the pdf has the form

$$p \propto \exp(f(x)), \quad p = \frac{1}{Z} \exp(f(x))$$

$f(x) = -(x-\mu)^T \Sigma (x-\mu)$.

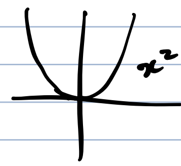
partition function.

Key property of f: it is <u>concave</u>!

e.g. think about when $\mu = 0$, $\Sigma = I$.

$f(x) = -\|x\|^2$.

$\|x\|^2$ is convex.



so $-\|x\|^2$ is concave.

<u>Def:</u> We say a distribution $p \propto \exp(f(x))$ is <u>log-concave</u> if $f(x)$ is concave.

<u>Note:</u> Most distributions are <u>not</u> log concave. The theory works best for log concave, but in practice things work more generally (sometimes...).

Several questions:

1 Given access to $f(x)$ and/or $\nabla f(x)$, can you efficiently sample from $p(x) \propto \exp(f(x))$?

2. Given samples $\underbrace{X_1, \dots, X_n \sim p(x), \text{ can you efficiently sample from } p?}_{\text{generative modeling / distribution learning.}}$

Langevin dynamics.

connection between optimization $\Longleftrightarrow$ sampling.

Let $\beta > 0$, and consider

"Gibbs distribution with inverse temperature β".

$$p_\beta \propto \exp(-\beta f(x))$$

<u>Claim:</u> As $\beta \to \infty$, almost all mass of p_β is at minimizer of f, if f is "nice"

easy to show in discrete case: $f: [n] \to \mathbb{R}$, we take

$$p_\beta(i) \propto \exp(-\beta f(i)).$$

so $$\Pr_{X \sim p_\beta}[X = i] = \frac{\exp(-\beta f(i))}{\sum_{i=1}^{n} \exp(-\beta f(i))}$$

Let $B_\varepsilon = \{ i : f(i) \le (1+\varepsilon) f_{min} \}$, $f_{min} = \min_i f(i)$.

Then $$\Pr_{X \sim p_\beta}[X \notin B_\varepsilon] = \frac{\sum_{i \notin B_\varepsilon} \exp(-\beta f(i))}{\sum_{i=1}^{n} \exp(-\beta f(i))} \quad (*)$$

But we have:

$$\sum_{i \notin B_\varepsilon} \exp(-\beta f(i)) \le n \cdot \exp(-\beta(1+\varepsilon) f_{min})$$

$$\sum_{i=1}^{n} \exp(-\beta f(i)) \ge \exp(-\beta \cdot f_{min}).$$

so $$(*) \le \frac{n \cdot \exp(-\beta(1+\varepsilon) f_{min})}{\exp(-\beta f_{min})}$$

$$= n \cdot \exp(-\varepsilon \beta f_{min}),$$ so if $\beta \ge \frac{\log n + \log 1/\delta}{\varepsilon f_{min}}$, then

$$\le \cancel{n \cdot} \exp(-(\cancel{\log n} + \log 1/\delta))$$

$$\le \delta.$$

As $\beta \to \infty$, i.e. as temperature drops, our distribution puts more and more mass around minimizer.

For $\beta < \infty$, it's like minimizing, but also need to keep some mass overall.

For minimization / optimization, a standard method is <u>gradient descent</u>

$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

step size.

For "nice" f, this converges to minima.

Langevin dynamics:

$$x_{t+1} = x_t - \frac{\varepsilon}{2} \nabla f(x_t) + \sqrt{\varepsilon}\, Z_t, \quad Z_t \sim N(0, I).$$

this scaling is important

A perspective from stochastic calculus: this is really a discretization of a <u>continuous-time</u> process.

$x_1, \dots, x_T$

$x_0, x_{0+\Delta t}, x_{2\Delta t}, \dots, x_1$

Consider gradient descent, and reparametrize so that timesteps are small.

$$x_{t+\Delta t} = x_t - \eta \nabla f(x_t).$$

$$\frac{x_{t+\Delta t} - x_t}{\Delta t} = -\eta \nabla f(x_t)$$

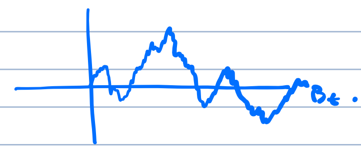
$$x_{t+\Delta t} - x_t = -\eta \nabla f(x_t)\Delta t.$$

$$dx_t = -\eta \nabla f(x_t) dt \quad \text{as } \Delta t \to 0.$$

PDE in x, t. This is called the "gradient flow"

Langevin: $dx_t = -\eta \nabla f(x_t) dt + \sqrt{2}\, dB_t$

"Brownian motion"



This is called a stochastic differential equation

<u>Thm:</u> If f is (strongly) logconcave, then Langevin dynamics converge to p.

i.e. $d_{TV}(\mathrm{law}(X_t), p) \to 0$ as $t \to \infty$.

$$d_{TV}(\pi, \sigma) = \int |\pi(x) - \sigma(x)|\, dx.$$

One can also show that discretized Langevin converges.

Diffusion models : sampling via data.

given samples $X_1, \dots, X_n \sim p$, how can you generate fresh samples from p?

<u>Ornstein-Uhlenbeck</u> (OU) process: Given a distribution $p = p_0$, the OU-process specifies a sequence of distributions p_t, where $p_t = \mathrm{law}(X_t)$,

$$X_t = \underbrace{\exp(-t)}_{\alpha_t} X_0 + \underbrace{\sqrt{1 - \exp(-2t)}}_{\beta_t = \sqrt{1 - \alpha_t^2}} Z_t, \quad X_0 \sim p_0, \; Z_t \sim \mathcal{N}(0, I).$$

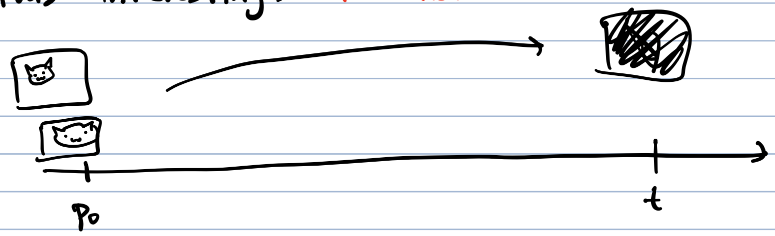
$$\alpha_t^2 + \beta_t^2 = 1.$$

The "right" way to think about OU is as an SDE:

$$dX_t = -X_t dt + \sqrt{2}\, dB_t.$$

Key point: OU takes a data distribution p_0 and turns it smoothly into noise.

Why is this interesting? It's not!



But: this <u>process can be reversed</u>

Fix some time T, define the reverse process $\overleftarrow{X}_t = X_{T-t}$ for $t = [0,T]$.

Then this reverse process satisfies:

$$d\overleftarrow{X}_t = (\overleftarrow{X}_t + 2\nabla \log p_{T-t}(\overleftarrow{X}_t))dt + \sqrt{2}\, dB_t$$

this takes noise $\to$ data. This is very interesting!

Two main problems:

1. This is still some continuous-time mumbo-jumbo.

   Can discretize this SDE, just like how you can for Langevin.

   discretize time $[0,T]$ into chunks of length Δt

   each step is of the form:

$$\overleftarrow{X}_{t+\Delta t} = \alpha \overleftarrow{X}_t + \alpha(\beta \cdot \nabla \log p_{T-t}(\overleftarrow{X}_t) + \gamma Z_t),$$

$$Z_t \sim \mathcal{N}(0, I).$$

2. This depends on $\nabla \log p_{T-t} := s_{T-t}$, which we don't know.

   Call this function the <u>score function</u>.

<u>Score matching</u> [Hyvärinen '05]

Suppose I want to find the best match to the score function

from some family of functions $\mathcal{F}$:

$$\arg\min_{s \in \mathcal{F}} \mathbb{E}_{x \sim p_t}[\|s(x) - \nabla \log p_t(x)\|^2]$$

the minimizer of this is the same as:

$$\arg\min_{s \in \mathcal{F}} \mathbb{E}_{\substack{x \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \left[\left\| s(X_t) + \frac{1}{\sqrt{1-\exp(-2t)}} Z_t \right\|_2^2\right], \quad X_t = \exp(-t)X_0 + \sqrt{1-\exp(-2t)}\, Z_t$$

reparametrize: let $\hat{s}(x) = -s(x)\sqrt{1-\exp(-2t)}$

$$\arg\min_{\hat{s}} \mathbb{E}_{\substack{x \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \left[\| z - \hat{s}(x_t) \|_2^2 \right] \quad (**)$$

local denoising function: given noisy sample, predict what part of it is noise.

we can optimize this objective given samples from p_0!

Given $x_1, \cdots, x_n \sim p_0$, let $z_1, \cdots, z_n \sim \mathcal{N}(0, I)$,

form $y_i = \exp(-t) x_i + \sqrt{1 - \exp(-2t)}\, z_i$

then $(**) \approx \frac{1}{n} \sum_{i=1}^{n} \| z_i - \hat{s}(y_i) \|_2^2$

this is a regression problem!

In practice: take $\hat{s}$ to be a large neural network.

> 12k citations!

Denoising Diffusion Probabilistic Models (DDPM): [Ho-Abbeel-Jain '20]

1. learn score function from data.
2. plug it into discretized reverse process
3. ???
4. profit (literally, see e.g. stable diffusion).

The backbone of modern generative models!

Many variants now: DDIM, latent diffusion, consistency models, etc.

Thm [CCLLSZ'23]: If the neural network learns the score effectively, then DDPM output is provably close to p_0!