Everything I Know about Diffusion & Flows & related things

Clark Miyamoto (cm6627@nyu.edu)

January 4, 2026

Contents

Part I
Sampling

In problems in scientific computing, you are tasked with generating samples $x \sim p(x)$. For example

- In Bayesian inference, one has a forward model $p(x|\theta)$ of the observed data x given some parameters θ. By Bayes' rule, you can ask what's the probability of seeing these parameters *given* the data.

  $$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

  In this context, querying $p(x|\theta)$ is computationally expensive, and the probability is usually unnormalized.

1 Importance Sampling

Part II
Stochastic Processes

Definition 1 *Let (Ω, A, P) be a probability space and let $T \subset \mathbb{R}$ be time. A collection of random variables $X_t, t \in T$ with values in $\mathbb{R}$ is called a* ***stochastic process****.*

Definition 2 *A stochastic process is called* ***measurable****, if $X : T \times \Omega \to S$ is measurable w.r.t. the sigma algebra the $\mathcal{B}(T) \times \mathcal{A}$*

Definition 3 (Ito Process) *A stochastic process of the following form*

$$X_t = X_0 + \int_0^t \mu_s \, ds + \int_0^t B_s \, dW_s \tag{1.1}$$

is called an ***Ito process****. In particular it solves the SDE*

$$dX_t = \mu_t \, dt + B_t \, dW_t \tag{1.2}$$

Theorem 1 (Ito's Lemma) *Consider the process $dX_t = \mu(X_t, t) \, dt + \sigma(X_t, t) \, dW_t$. Consider a process related by a function $Y_t = f(X_t)$. Then*

$$dY_t = \left(\partial_t u(X_t, t) \, + \frac{1}{2} \partial_{xx} u(X_t, t) \, \sigma^2(X_t, t) \right) dt \tag{1.3}$$

2 Fokker-Planck Equation

There are a couple forms of a stochastic differential equation. Here I'll assume X_t solves Geometric Brownian Motion for illustration

- Dynamics, an SDE describes how a particle at position X_t should move under a noisy force dW_t.

$$dX_t = X_t(\mu\, dt + \sigma\, dW_t) \tag{2.1}$$

- Random Variable, upon solving the SDE you can sample the solution at any point

$$X_t = X_0 \exp\left((\mu - \frac{1}{2}\sigma^2)t + \sigma Z_t\right) \tag{2.2}$$

  where $Z_t \sim \mathcal{N}(0, \sqrt{t})$.

- Distribution, any random variable has a probability distribution representation as well

$$p_t(x) = \frac{1}{\sqrt{2\pi t}}\frac{1}{\sigma x}\exp\left(-\frac{[\log x - (\mu - \sigma^2/2)x]^2}{2\sigma^2 t}\right) \tag{2.3}$$

  where $\text{Law}(X_t) = p_t$.

Because you have a differential equation which describes the random variable, you should have a differential equation which describes the time evolution on the distribution $p_t(x)$. We'll describe this relationship here.

Theorem 2 (Fokker-Planck Equation) *Consider a stochastic process X_t*

$$dX_t = \mu(X_t, t)\, dt + \sigma(X_t, t)\, dW_t \tag{2.4}$$

where μ, σ are deterministic functions. The probability density $p_t = Law(X_t)$ satisfies the corresponding partial differential equation

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}(\mu(x,t)\, p) + \frac{1}{2}\frac{\partial^2}{\partial x^2}(\sigma^2(x,t)\, p) \tag{2.5}$$

Proof 1 *Consider the stochastic process $Y_t = f(X_t)$, where f is a test function (for distributions). Using Ito's Lemma, we can find dY_t. Prior to that calculation, note that $dX_t^2 = \sigma^2(X_t, t)\, dt + \mathcal{O}(dt^{3/2})$*

$$df = \partial_x f(X_t)\, dX_t + \frac{1}{2}\partial_{xx} f(X_t) dX^2 \tag{2.6}$$

$$= \partial_x f(X_t)(\mu(X_t, t)\, dt + \sigma(X_t, t)\, dW_t) + \frac{1}{2}\partial_{xx} f(X_t)\sigma^2(X_t, t)\, dt \tag{2.7}$$

$$= \Big(\partial_x f(X_t)\mu(X_t, t) + \frac{1}{2}\partial_{xx} f(X_t)\sigma^2(X_t, t)\Big)dt + \partial_x f(X_t)\sigma(X_t, t) dW_t \tag{2.8}$$

We can now compute $\mathbb{E}[df]$, note that $\mathbb{E}[dW_t] = 0$.

$$\frac{d}{dt}\mathbb{E}[f] = \mathbb{E}\left[\partial_x f(X_t)\mu(X_t, t) + \frac{1}{2}\partial_{xx} f(X_t)\sigma^2(X_t, t)\right] \tag{2.9}$$

$$\int f(x)\,\frac{dp(x,t)}{dt}\,dx = \int \left(\partial_x f(x)\mu(x,t) + \frac{1}{2}\partial_{xx} f(x)\sigma^2(x,t)\right) p(x,t)\, dx \tag{2.10}$$

Now if you can show something holds for all functions f, then you can extract an equation

$$\int f(x)\frac{dp(x,t)}{dt}\, dt = \int f(x)\left(-\frac{\partial}{\partial x}(\mu(x,t)p(x,t)) + \frac{1}{2}\frac{\partial^2}{\partial x^2}(\sigma^2(x,t)p(x,t))\right) dx \tag{2.11}$$

$$\implies \frac{dp}{dt} = -\frac{\partial}{\partial x}(\mu(x,t)\, p) + \frac{1}{2}\frac{\partial^2}{\partial x^2}(\sigma^2(x,t)\, p) \tag{2.12}$$

QED

2.1 Time Reversibility & Backwards Equation

In the forwards equation, it was assume that you knew $\text{Law}(X_0) = p_0$. What if you have a situation in the opposite context, that is you only knew $\text{Law}(X_T) = p_T$?

Theorem 3 (Reverse Fokker-Planck Equation) *Consider the parameterization $\tau = T - t$.*

3 Feynman-Kac

3.1 Aside on Solving Path Integrals

4 Change of Measure

4.1 Radon-Nikodym Derivative

Definition 4 *Consider a probability space $(\Omega, \mathcal{F})$ and two measures $dP(\omega)$ and $dQ(\omega)$. We'll say the **Radon-Nikodym derivative** of dP w.r.t. dQ (denoted as $\frac{dP}{dQ}(\omega)$) if*

$$P(A) = \int_{\omega \in A} L(\omega) dQ(\omega) \tag{4.1}$$

An equivalent definition is L is the RN-derivative if for any test function V

$$\int V(\omega) dP(\omega) = \int V(\omega) L(\omega) dQ(\omega) \tag{4.2}$$

Ok this might look all that fancy, but this massively simplifies if the two probabilities have densities. Say the two measures have densities $dP(x) = p(x)\,dx$ and $dQ(x) = q(x)\,dx$. Checking the RN-derivative definition

$$\int V(x)p(x)dx = \int V(x)L(x)q(x)dx \implies L(x) = \frac{p(x)}{q(x)} \tag{4.3}$$

In this case, the Radon-Nikodym derivative is just the ratio of densities. Statisticians call this the "likelihood ratio", which is also why we gave it the L notation.

4.2 Girsinov's Theorem

However this was just defined for random variables, how can we do a change of measure for stochastic processes?

Part III
Generative Models

The problem setup for generative models is as follows

- You only have access to samples $\mathcal{D} = \{x_i\}_i$, which are realizations $x_i \sim_{iid} \pi$ of some target distribution. Often x_i is a high dimensional object.
- You want to construct $p_\theta \approx \pi$.

5 (Variational) Autoencoders

Consider a machine learning model with a bottleneck. That is your model is the function

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^d, \text{ s.t. } f_\theta(x) = \phi_\theta(\psi_\theta(x)) \tag{5.1}$$

$$\text{where } \psi_\theta : \mathbb{R}^d \to \mathbb{R}^n, \ \phi_\theta : \mathbb{R}^n \to \mathbb{R}^d, \ n < d \tag{5.2}$$

If you construct a loss function like

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim p_{data}}[\|x - f_\theta(x)\|^2] \tag{5.3}$$

you're kinda forcing the model to learn a low dimensional representation of the data. This is called an **autoencoder**. Often times asking for a deterministic reconstruction is too difficult? What if we ask for a reconstruction in distribution? That is $p_\theta \equiv \text{Law}(f_\theta(x)) \approx \pi$... This is called a **variational autoencoder**.

One method to achieve this is by performing gradient descent on distance metrics between probability distributions. One such is the KL divergence.

$$D_{\text{KL}}(p \,\|\, \pi) \equiv \mathbb{E}_{x \sim p}\left[\log \frac{p(x)}{\pi(x)}\right] \tag{5.4}$$

Note, KL is not symmetric $D_{\text{KL}}(p_\theta \,\|\, \pi) \neq D_{\text{KL}}(\pi \,\|\, p_\theta)$, so it's worth considering which one to make first. The question I ask myself, is it easier to sample p_θ or π? In the case of generative models, π is a empirical distribution (i.e. a bunch of photos of cat), so it's easy to sample that.

$$D_{\text{KL}}(\pi \,\|\, p_\theta) \equiv \mathbb{E}_{x \sim \pi}\left[\log \frac{\pi(x)}{p_\theta(x)}\right] \tag{5.5}$$

$$\nabla_\theta D_{\text{KL}}(\pi \,\|\, p_\theta) = \mathbb{E}_{x \sim \pi}\left[-\frac{p_\theta(x)}{\pi(x)} \frac{\pi(x)}{p_\theta(x)^2} \nabla_\theta p_\theta(x)\right] \tag{5.6}$$

$$= -\mathbb{E}_{x \sim p_{data}}\left[\nabla_\theta \log p_\theta(x)\right] \tag{5.7}$$

6 Denoising Diffusion Probabilistic Models (DDPM)

In DDPM your neural network attempts to learn how to add white-noise to the data (iteratively). We can mathematically state this as

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t} z_t \tag{6.1}$$

where $x_0 \sim p_{data}$ and $z_t \sim^{iid} \mathcal{N}(0, \mathbb{I})$. However, when evaluating this, we don't have to go recursively

$$\begin{aligned}
x_t &= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_{t-2}} z_{t-1}) + \sqrt{1-\alpha_t} z_t && (6.2)\\
&= \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t \alpha_{t-1}} z_{t-1} && (6.3)\\
&= \sqrt{\prod_{i=1}^{t} \alpha_i}\, x_0 + \sqrt{1-\prod_{i=1}^{t} \alpha_i}\, z_0 && (6.4)\\
&\sim \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t)\mathbb{I}) \qquad \text{where } \bar{\alpha}_t \equiv \prod_{i=1}^{t} \alpha_i && (6.5)
\end{aligned}$$

where we've used the fact that the sum of two iid Gaussians is still a Gaussian.

7 Score Based Diffusion (SBD)

In score based diffusion, you learn the score function

$$s(x;t) = \nabla_x \log p_t(x)$$

Intuitively, the score says which direction will yields samples of high probability, so this has to be an important quantity in generative modeling.

> This quantity also has some nice properties. For example, it no longer is concerned about normalization. For example, consider an exponential family distribution
>
> $$\pi(x) = e^{-U(x)}/Z \implies \nabla_x \log \pi(x) = -\nabla_x U(x)$$
>
> The score is not dependent on the normalization Z.

The trade off is that now we must figure out ways to sample when only having access to gradient information (not log probability).

In DDPM, we iteratively noised the

7.0.1 Learning (Score Matching)

When learning distributions, we use a KL divergence as the loss function

$$D_{KL}(q\|p) \equiv \mathbb{E}_{x\sim q}\left[\log q - \log p\right] \tag{7.1}$$

However since we're learning the score $s_\theta(x;t) \approx \nabla_x \log p_t(x)$, we can use the Fisher Divergence

$$D_F(p\|q_\theta) \equiv \mathbb{E}_{x\sim p}[(\nabla_x \log p(x) - \nabla_x \log q_\theta(x))^2] \tag{7.2}$$

Similarly to the KL divergence, $D_F \geq 0$ and $D_F = 0$ if and only if $p = q_\theta$. Since we're using this as loss function, let's inspect the gradient w.r.t. the model parameters θ.

$$\begin{aligned}
\nabla_\theta D_F(p\|q_\theta) &= \nabla_\theta \mathbb{E}_{x\sim p}[(\nabla_x \log p(x) - \nabla_x \log q_\theta(x))^2] && (7.3)\\
&= \nabla_\theta \mathbb{E}_{x\sim p}[(\nabla_x \log p(x))^2 + (\nabla_x \log q_\theta(x))^2 - 2\nabla_x \log p(x)\cdot \nabla_x \log q_\theta(x)] && (7.4)\\
&= \nabla_\theta \mathbb{E}_{x\sim p}[(\nabla_x \log q_\theta(x))^2 - 2\nabla_x \log p(x)\cdot \nabla_x \log q_\theta(x)] && (7.5)\\
&= \nabla_\theta \mathbb{E}_{x\sim p}[(\nabla_x \log q_\theta(x))^2 + 2\,\nabla_x \cdot \nabla_x \log q_\theta(x)] && (7.6)
\end{aligned}$$

Using the notation $s_\theta(x) = \log q_\theta(x)$, we arrive at the **score matching (SM) objective**

$$\mathcal{L}_{\text{SM}}(\theta) = \mathbb{E}_{x\sim p}[s_\theta(x)^2 + 2\nabla_x \cdot s_\theta(x)] + \text{Constants} \tag{7.7}$$

Remember in practice this is done for various levels of noise p_t, so you'll use

$$\boxed{\mathcal{L}_{\text{SM}}(\theta) = \mathbb{E}_{t\sim[0,T]}\mathbb{E}_{x\sim p_t}[s_\theta(x;t)^2 + 2\nabla_x \cdot s_\theta(x;t)]} + \text{Constants} \tag{7.8}$$

7.0.2 Inference

Since you're learning the score, you can try overdamped Langevin dynamics.

$$dX_t = -\nabla_x \log p_{t=1}(x)\,dt + \sqrt{2}\,dW_t \tag{7.9}$$

However because the distribution your'e sampling is highly multimodal (or at least for images it seems to be), the mixing time would be incredibly slow. Another thing is that in SBD you've learned a path between p_0 and p_{data} (given by $\nabla_x \log p_t$), so you should take advantage of that extra information.

A natural solution is to reverse the SDE of your noising process. Say you noised the data using

$$dX_t = \mu(X_t, t)\, dt + \sigma(X_t, t) dW_t \iff \partial_t p = \partial_x(\mu\, p) + \frac{1}{2}\partial_{xx}(\sigma^2\, p) \tag{7.10}$$

for times $t \in [0, T]$. The boundary condition is that you asserted $p_{t=0} = p_{data}$, where μ, σ are chosen s.t. $p_{t=T} = p_{prior}$ (which is hopefully something easy to sample). You can flow back in time $\tau = T - t \in [0, T]$ according the reverse Fokker Planck

$$\partial_\tau p_\tau = -\partial_x(\mu\, p_\tau) + \frac{1}{2}\partial_{xx}(\sigma^2\, p_\tau) \tag{7.11}$$

$$\iff d\tilde{X}_\tau = [-\mu(X_\tau, \tau) + \sigma^2(X_\tau, \tau)\nabla_x \log p_\tau(x)]\, d\tau + \sigma(X_\tau, \tau)\, dW_\tau \tag{7.12}$$

where you simulate with the time increments backwards $T, T - \epsilon, T - 2\epsilon, ..., 0$. The boundary conditions are now $p_{\tau=0} = p_{prior}$, and hopefully this yields samples from $p_{\tau=T} = p_{data}$.

8 Flow Matching

Definition 5 (Flow Model) *A flow model is described by the ODE*

$$X_0 \sim p_{init} \tag{8.1}$$

$$\frac{d}{dt}X_t = u_t^\theta(X_t) \tag{8.2}$$

where $u_t^\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is a vector field (constructed with a neural network w/ parameters θ). We hope to construct u_t^θ s.t.

$$X_1 \sim p_{data} \iff \psi_1^\theta(X_0) = p_{data} \tag{8.3}$$

where ψ_t^θ is the induced flow due to the drift field u_t^θ.

In intermediate times, your transported samples $X_t \sim p_t$ will have some path in probability space p_t. In flow models, you specify the probability path p_t, and then derive the target drift field u_t^{target} (this is your training objective, in the sense you'll optimize $\mathcal{L}(\theta) = \|u_t^\theta - u_t^{target}\|$). As a user, you have a lot of possible algorithms to choose from, while other algorithms (e.g. normalizing flows, DDPM) ! An alternative method (Stochastic Interpolants) will specify the path at the level of the samples.

Definition 6 (Probability Path) *Consider the **conditional probability path**, which is a family of distributions $\{p_t(\cdot|z)\}_{0\le t\le 1}$ such that*

$$p_0(x|z) = p_{init}(x),\ p_1(x|z) = \delta_z(x),\ \forall z \in \mathbb{R}^d \tag{8.4}$$

where δ_z is a Dirac delta distribution which returns z.

*A **marginal probability path** is a familiy of distributions $\{p_t(\cdot)\}_{0\le t\le 1}$ such that*

$$p_t(x) = \int p_t(x|z)p_{data}(z)\,dz \tag{8.5}$$

Notice the conditions on the conditional probability path imply $p_0 = p_{init}$ and $p_1 = p_{data}$.

Every conditional probability path gives a marginal probability path which satisfies the objectives of the flow model. So if you have a probability distribution which has an easy to sample limit and a Dirac delta limit, it would make for a good candidate for a conditional probability path. An example would be a *Gaussian probability path*, which is used by DDPMs.

$$p_t(\cdot|z) = \mathcal{N}(\alpha_t z, \beta_t^2 \mathbb{I}) \tag{8.6}$$

Where the boundary conditions $\alpha_0 = \beta_1 = 0$ and $\alpha_1 = \beta_0 = 1$, ensure that $p_1 = p_{data}$ and $p_0 = \mathcal{N}(0, \mathbb{I})$.

Theorem 4 *For every $z \in \mathbb{R}^d$, let $u_t^{target}(\cdot|z)$ denote a conditional vector field, defined s.t. the corresponding ODE yields the conditional probability path $p_t(\cdot|z)$*

$$\frac{d}{dt}X_t = u_t^{target}(X_t|z). \tag{8.7}$$

The marginal vector field $u_t^{target}(x)$ is given by

$$u_t^{target}(x) = \int u_t^{target}(x|z)\frac{p_t(x|z)p_{data}(z)}{p_t(x)}\,dz, \tag{8.8}$$

follows the marginal probability path

$$X_0 \sim p_{init},\ \frac{d}{dt}X_t = u_t^{target}(X_t) \implies X_t \sim p_t,\ t \in [0,1] \tag{8.9}$$

In particular, $X_1 \sim p_{data}$ for this ODE. So we might say "u^{target} converts p_{init} to p_{data}".

Proof 2

So to summarize

1. Specify a conditional probability path $p_t(\cdot|z)$, where $p_0 = p_{init}$ and $p_1 = \delta_z$.
2. Compute the corresponding u_t^{target} using

8.1 Training

9 Stochastic Interpolants

Definition 7 (Stochastic Interpolant) *A stochastic interpolant is a stochastic process* x_t

$$x_t = I(t, x_0, x_1) + \gamma(t)z \tag{9.1}$$

with the following properties

- *Boundary conditions of interpolant:* $I(t = 0, x_0, x_1) = x_0$ *and* $I(t = 1, x_0, x_1) = x_1$.
- *Coupling of* x_0, x_1*: These are random variables s.t.* $(x_0, x_1) \sim \nu(x_0, x_1)$. *The coupling must be marginalize to the original distributions.*

$$\int \nu(x_0, x_1)\, dx_1 = p_0(x_0) \qquad \int \nu(x_0, x_1)\, dx_0 = p_1(x_1) \tag{9.2}$$

- $z \sim \mathcal{N}(0, \mathbb{I})$ *is Gaussian noise, and it's independent to* x_0, x_1. *That is* $z \perp x_0, x_1$.

To understand this notation, let's compute the mean and variance of x_t

$$\mathbb{E}[x_t] = \mathbb{E}_{(x_0,x_1)\sim\nu}[I(t, x_0, x_1)] \tag{9.3}$$
$$\mathrm{Cov}[x_t] = \mathrm{Cov}_{(x_0,x_1)\sim\nu}[I(t, x_0, x_1)] + \gamma^2(t)\mathbb{I} \tag{9.4}$$

So instead of defining the "flow map" (which defines a transport at the level of the distribution) or relying on Langevin dynamics to define the measure transport, we can talk about the transport at the level of the samples.

Theorem 5 *Consider the interpolant* x_t *defined in (9.1), it has Law*$(x_t) = p(t, x)$ *for times* $t \in [0, 1]$ *s.t.* $p(t = 0, x) = p_0(x)$ *and* $p(t = 1, x) = p_1(x)$. *In addition it solves the transport equation*

$$\frac{\partial p}{\partial t} = \nabla \cdot (b\, p) \tag{9.5}$$

where $b(t, x) = \mathbb{E}[\dot{x}_t | x_t = x] = \mathbb{E}[\partial_t I(t, x_0, x_1) + \dot{\gamma}(t)z | x_t = x]$.

Proof 3 *Consider the Fourier transform of the transport equation*

$$\partial_t \tilde{p} = (-ik) \cdot \tilde{b}\tilde{p} \tag{9.6}$$

All one needs to do is to calculate the time derivative of $\tilde{p}$ *(Fourier transform of* p*), and you'l find the* $\tilde{b}$.

$$\tilde{p} = \mathbb{E}_{x_t\sim p_t(x)}[e^{ik\cdot x_t}] \tag{9.7}$$
$$= \mathbb{E}_{x_t\sim p_t}[\exp(ik \cdot I_t(x_0, x_1) + i\gamma(t)k \cdot z)] \qquad \textit{Definition} \tag{9.8}$$
$$= \mathbb{E}[e^{ik\cdot I_t}]e^{-\frac{1}{2}\gamma^2(t)|k|^2} \tag{9.9}$$

$$\partial_t \tilde{p} = ik\, \mathbb{E}[\dot{I}_t e^{ikI_t}]e^{-\frac{1}{2}\gamma^2(t)|k|^2} - \gamma(t)\dot{\gamma}(t)|k|^2 e^{-\frac{1}{2}\gamma^2(t)|k|^2} \tag{9.10}$$
$$= (ik\, \mathbb{E}[\dot{I}_t e^{ikI_t}] - \gamma(t)\dot{\gamma}(t)|k|^2)e^{-\frac{1}{2}\gamma^2(t)|k|^2} \tag{9.11}$$

9.0.1 Learning

9.0.2 Inference

In Theorem 5, we construct a transport equation which tells you how to construct a drift field b s.t. the stochastic interpolant has $\text{Law}(x_t) = p(t, x)$. However solving a high dimensional PDE is difficult, we can instead using the Fokker Planck relationship to perform inference at the level of x_t.

Part IV
Formula Sheet

Information Theory Bounds

Data Processing Inequality: Consider the markov chain $X \to Y \to Z$.

$$I(X;Y) \geq I(X;Z) \tag{9.12}$$

Pinsker's Inequality

$$\mathrm{TV}(p,q) \leq \sqrt{\frac{1}{2}\mathrm{KL}(p\|q)} \tag{9.13}$$