Best customized Beamer Themes

Beautiful title slides: Model 1

LaTeX-beamer.com

September 29, 2024

Euler Method for Solving ODEs

Given ODE:

$$\frac{dX_t}{dt} = \mu(t, X_t) \quad \text{with} \quad X_0 = x_0$$

- X_t: State at time t
- $\mu(t, X_t)$: Drift function (rate of change)
- $X_0 = x_0$: Initial condition

Euler Method Approximation:

- Discretize time: $\delta \ll 1$
- Set initial estimate $\hat{X}_0 = x_0$
- Iterate using:

$$\hat{X}_{t+\delta} = \hat{X}_t + \delta\mu(t, \hat{X}_t)$$

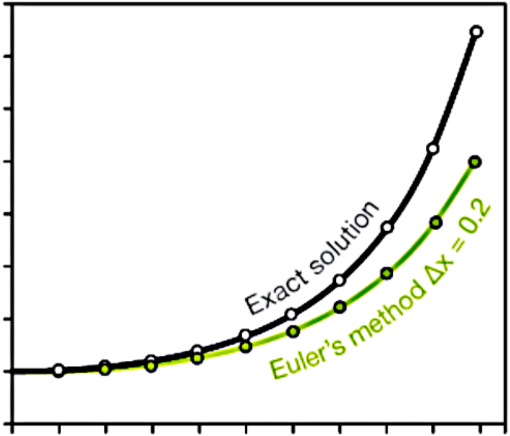
Procedure:

- Break continuous time into small discrete steps
- Update state X_t using drift at each step
- Approximate solution accumulates over time

Applications of ODEs

Ordinary Differential Equations (ODEs) appear frequently in real-world scenarios:

- **Population Growth:** Predicting how populations evolve over time.
- **Physics:** Describing motion, e.g., Newton's second law $F = ma$.
- **Finance:** Used in models such as the Black-Scholes equation for option pricing.



Brownian Motion

Stochastic Differential Equation:

$$\frac{dX_t}{dt} = \mu(t, X_t) + (noise)$$

- Describes random motion, e.g., particles moving in a fluid.
- Trajectories are uncertain due to added noise.

Figure: Brownian Motion

Brownian motion (cont)

Equation with Noise:

$$dX_t = \mu(t, X_t)dt + \sigma dW$$

- σ quantifies the noise intensity.
- $dW = W_{t+dt} - W_t$ represents a Wiener Process.

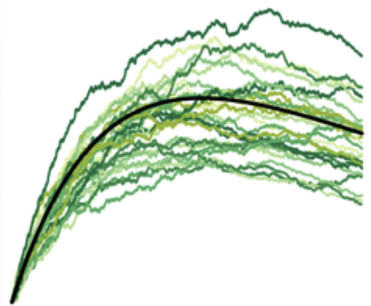


Figure: Stochastic Trajectories

Properties of Brownian Motion

Key Properties:

- Markov property: Independent increments.
- Gaussian distribution for changes:

$$E[W_{t+\delta} - W_t] = 0, \quad Cov(W_{t+\delta} - W_t) = \delta I$$

- The covariance matrix indicates linear variance growth.

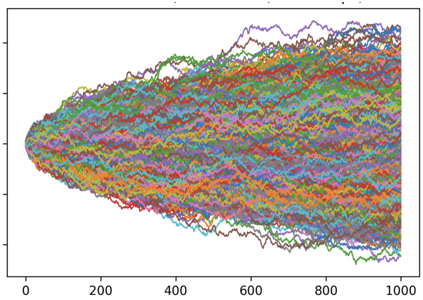


Figure: Gaussian Properties of Increments

Simulating Brownian Motion

Simulation:

$$W_{t+\delta} = W_t + \sqrt{\delta}\mathcal{N}(0, I)$$

- Easy to simulate Gaussian-distributed increments.
- Visualize random trajectories over time.

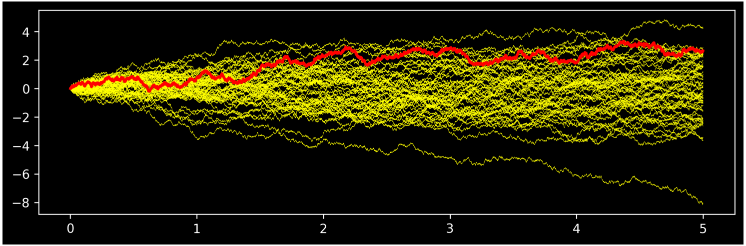


Figure: Simulated Brownian Motion

Stochastic Differential Equation

SDE Formulation:

$$dX_t = \mu(t, X_t)dt + \sigma dW_t$$

- Combines both deterministic (drift) and stochastic (random noise) components.
- Brownian motion introduces uncertainty into the paths.

Example - Stock Price SDE

SDE for Stock Prices:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

- $\mu S_t dt$: Expected growth (deterministic).
- $\sigma S_t dW_t$: Market volatility (stochastic).

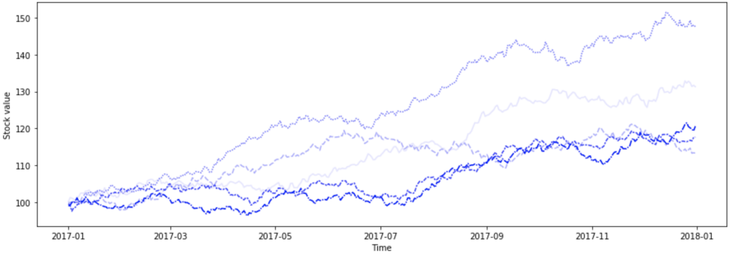


Figure: Possible growth scenarios of a (fictional) stock value

Simulating SDE - Euler-Maruyama Method

Euler-Maruyama Method:

$$\hat{X}_{t+\delta} = \hat{X}_t + \mu(t, \hat{X}_t)\delta + \sigma\sqrt{\delta}\mathcal{N}(0, 1)$$

- Discrete approximation of continuous SDE.
- Combine drift and Brownian motion steps.

Specific SDE Example

Consider the following SDE:

$$dX_t = -5(X_t - \sin(5t))dt + dW_t$$

- $-5(X_t - \sin(5t))dt$: Drift pulls X_t towards $\sin(5t)$.
- dW_t: Adds Brownian motion.

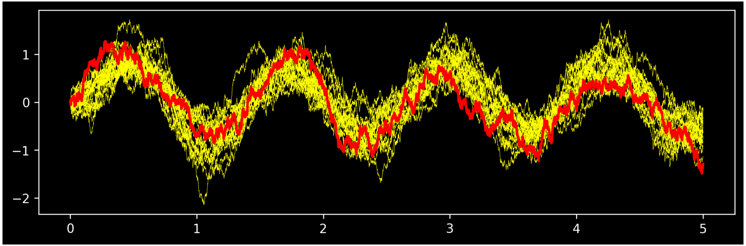


Figure: Simulated SDE

Ornstein-Uhlenbeck (OU) Process - Overview

A classical, well-controlled Gaussian stochastic process used in fields like physics, finance, and biology.

$$dX_t = -\frac{1}{2}\frac{X_t}{\sigma^2}dt + dW_t$$

- $-\frac{1}{2}\frac{X_t}{\sigma^2}dt$: This term is proportional to X_t and pulls X_t towards zero. This property is known as **mean reversion**.
- dW_t: Represents the random noise component.

Ornstein-Uhlenbeck (OU) Process - Simulation

The OU process provides an exact transition probability equation:

$$X_{t+\delta} \mid (X_t = x) \sim \mathcal{N}\left(e^{-\frac{\delta}{2\sigma^2}}x, \sigma^2\left(1 - e^{-\frac{\delta}{\sigma^2}}\right)I\right)$$

- The mean $e^{-\frac{\delta}{2\sigma^2}}x$ pulls the process back to zero. - The variance $\sigma^2\left(1 - e^{-\frac{\delta}{\sigma^2}}\right)$ grows with time δ, but is bounded.

To simulate:

$$X_{t+\delta} = \alpha x + \sqrt{1-\alpha^2}\mathcal{N}(0, \sigma^2 I)$$

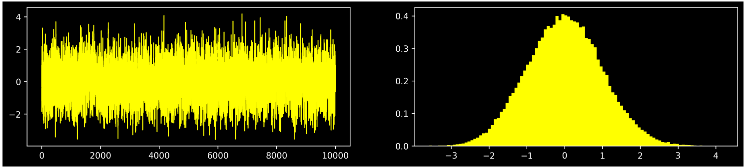


Figure: Simulated OU process

Langevin Diffusion - Introduction

Langevin diffusion is a stochastic process often used for sampling from a distribution $\pi(dx)$. The process is designed so that $\pi(dx)$ becomes the invariant distribution, meaning:

- **Invariant Distribution**: If $X_t \sim \pi(dx)$, then $X_T \sim \pi(dx)$ for all $T \geq t$.
- **Convergence**: Starting from any initial distribution, we obtain:

$$X_t \to \pi(dx) \quad \text{as} \quad t \to \infty.$$

Langevin Diffusion - Equation and Example

The Langevin equation is given by:

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\, dt + dW_t$$

If $\pi = \mathcal{N}(\mu, \Gamma)$ (normal distribution), Langevin diffusion is a form of the Ornstein-Uhlenbeck process:

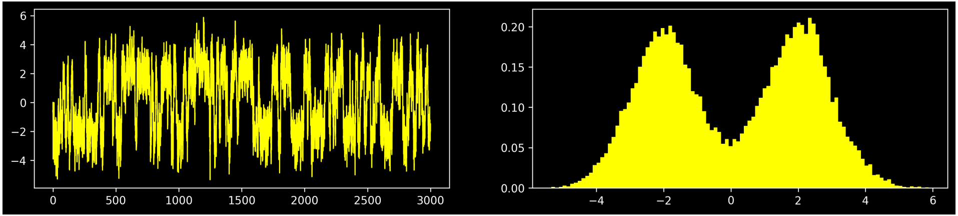


Figure: Simulated Langevin

Taylor Expansion for Stochastic Processes

If X_t follows a diffusion process described by:

$$dX_t = \mu(t, X_t)dt + \sigma dW_t,$$

we can apply the Taylor expansion to approximate the value of a function $F(X_t)$ at time $t + \delta$. The expansion becomes:

$$F(X_{t+\delta}) \approx F(X_t) + (LF)(t, X_t)\delta + (\text{Noise of order } \delta^{1/2}).$$

- $F(X_t)$: value of the function at time t
- $(LF)(t, X_t)\delta$: **Generator** of the diffusion process
- Noise term $\delta^{1/2}$: Represents the random component due to noise

Ito's Lemma and Generator of Diffusion

The generator L of the diffusion process is calculated as:

$$LF(t, X_t) = \mu(t, X_t)\nabla F(X_t) + \frac{\sigma^2}{2}\Delta F(X_t).$$

- **Drift term**: $\mu(t, X_t)\nabla F(X_t)$
  - $\mu(t, X_t)$: Drift function describing the deterministic change
  - $\nabla F(X_t)$: Gradient of F, describing how $F(X_t)$ changes with X_t
- **Diffusion term**: $\frac{\sigma^2}{2}\Delta F(X_t)$
  - $\Delta F(X_t)$: Laplacian, describing the second-order variation in F
  - The factor $\frac{\sigma^2}{2}$ accounts for the properties of Brownian motion.
- **Noise**: Proportional to $\delta^{1/2}$, representing the random fluctuation due to the stochastic process.

Fokker-Planck Equation

Given the SDE:

$$dX_t = \mu(t, X_t)dt + \sigma dW,$$

where X_0 follows the distribution $\pi_0(dx)$, the goal is to find the distribution $\pi_t(dx)$ of X_t at time t.

To describe $\pi_t(dx)$, we utilize:

- **Test function** $\phi(x)$
- **Taylor expansion**
- **Integration by parts**

The Taylor expansion yields:

$$\phi(X_{t+\delta}) \approx \phi(X_t) + L\phi(X_t) \cdot (X_{t+\delta} - X_t) + \text{higher-order terms.}$$

Expectation and Fokker-Planck Equation

Taking expectation:

$$E_{X_{t+\delta} \sim \pi_{t+\delta}}[\phi(X_{t+\delta})] \approx E_{X_t \sim \pi_t}[\phi(X_t) + L\phi(X_t)(X_{t+\delta} - X_t) + \text{higher-order terms}].$$

Ignoring higher-order terms gives:

$$\mathbb{E}_{X_{t+\delta} \sim \pi_{t+\delta}}\left[\phi(X_{t+\delta})\right] \approx \mathbb{E}_{X_t \sim \pi_t}\left[\phi(X_t) + L\phi(X_t)\delta\right].$$

By integrating by parts, we obtain the Fokker-Planck equation:

$$\frac{\partial}{\partial t}\pi_t = -\nabla \cdot (\mu \pi_t) + \frac{\sigma^2}{2}\Delta \pi_t.$$

- $\partial_t \pi_t$: Change of the probability distribution π_t over time.
- $\nabla \cdot (\mu \pi_t)$: Drift term, describing the movement of the distribution under the influence of the drift function.
- $\frac{\sigma^2}{2}\Delta \pi_t$: Diffusion term, describing the spreading of the probability distribution due to random noise.

them gif

Kullback-Leibler Divergence

KL divergence measures the difference between two probability distributions $p(dx)$ and $q(dx)$:

$$\mathrm{KL}(p \parallel q) = \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx.$$

- $\int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$: Expectation of $\log \left(\frac{p(x)}{q(x)} \right)$ with respect to $p(x)$, indicating how much information is lost when using $q(x)$ instead of $p(x)$.
- KL divergence is non-negative and equals 0 only when $p(x) = q(x)$; they are identical.
- KL divergence is not symmetric: $\mathrm{KL}(p \parallel q) \neq \mathrm{KL}(q \parallel p)$.

KL Divergence in Variational Inference

Variational inference (VI) is a method for approximating Bayesian inference, where we aim to approximate the distribution $\pi(x)$ with a simpler distribution $q_\lambda(x)$, parameterized by λ.
The goal of VI is to find $q_\lambda(x)$ that is as close as possible to $\pi(x)$:

$$\lambda^* = \arg\min_{\lambda} \mathrm{KL}(q_\lambda \parallel \pi).$$

In the reverse diffusion process, we aim to reconstruct the original data from noisy data. This involves estimating the data distribution at each step in the reverse diffusion process, thus needing to approximate the true data distribution at each random step. KL divergence helps evaluate the accuracy of the distribution $q_\lambda(x)$ (the distribution predicted by the model) compared to the true data distribution $\pi(x)$.

Denoising Diffusion Probabilistic Models (DDPM)

DDPM is a data generation method based on the diffusion process, where the original data undergoes a gradual noise addition until it becomes pure noise. The model then learns to reverse this process to reconstruct the original data from the noise.

- We have N samples $\mathcal{D} \equiv \{x_i\}_{i=1}^{N} \in \mathbb{R}^D$ from an unknown distribution $\pi_{\text{data}}(dx)$.
- Consider a diffusion process $\{X_t\}_{t=0}^{T}$:
  - Starts from the data distribution: $X_0 \sim p_0(dx) \equiv \pi_{\text{data}}(dx)$.
  - Ends near the desired distribution: $X_T \sim p_T(dx) \equiv \pi_{\text{ref}}(dx)$.
- Typically, $\pi_{\text{rev}}(dx) \sim \mathcal{N}(0, I)$, and the diffusion process is modeled as an Ornstein-Uhlenbeck (OU) process.
- Once there is completely noisy data, this dispersion process is reversed

Noising OU Process

The Ornstein-Uhlenbeck (OU) process is used in the noise addition process:

$$dX_t = -\frac{1}{2} X_t \, dt + dW \tag{1}$$

With $X_0 \sim \pi_{\text{data}}(dx)$, it converges to the normal distribution $\pi_{\text{ref}}(dx) = \mathcal{N}(0, I)$, making X_t a Gaussian noise after sufficient time.

- Transition densities:

$$X_{t+s} \mid (X_t = x_t) = \alpha_s x_t + \sqrt{1 - \alpha_s^2}\, \mathbf{n} \tag{2}$$

- Where:
  - $\alpha_s = \exp(-s/2)$
  - $\mathbf{n} \sim \mathcal{N}(0, I)$ is an isotropic Gaussian.

Efficient Simulation and Marginal Distribution

The OU process can be directly simulated for X_{t+s} without relying on Euler's method, simplifying the computation of the noise addition in diffusion models.

- The marginal distribution at time $0 \leq t \leq T$ is given by:

$$X_t \sim p_t(dx) = \int P(X_t \in dx \mid X_0 = x_0)\pi_{\mathrm{data}}(dx_0) \tag{3}$$

How to Reverse a Diffusion?

Consider a diffusion equation:

$$dX_t = \mu(X_t)\,dt + \sigma\,dW \tag{4}$$

with $0 \leq t \leq T$. The goal is to reverse this process, moving from time $t = T$ back to $t = 0$.

Definition of reverse diffusion:

$$X_s^{\text{rev}} = X_{T-s} \tag{5}$$

Since this process runs backward in time, we have the initial condition:

$$X_{\text{rev},0} \sim p_T(dx) \quad \text{and} \quad X_{\text{rev},T} \sim p_0(dx) \tag{6}$$

At $s = 0$, the reverse process starts from the distribution at time $t = T$ and gradually returns to the distribution at time $t = 0$.

Reverse Diffusion Equation

From the above definitions, we derive the reverse diffusion equation:

$$dX_s^{\text{rev}} = -\mu(X_s^{\text{rev}})\, ds + \sigma^2 \nabla \log p_{T-s}(X_s^{\text{rev}})\, ds + \sigma\, dB \tag{7}$$

- $\mu(X_s^{\text{rev}})$: drift term.
- $\nabla \log p_{T-s}(X_s^{\text{rev}})$: gradient of the log distribution at time $T - s$.
- σ: noise coefficient, similar to that in the forward process.
- dB_s: Brownian motion.

Reverse Diffusion Process

In the forward process, the diffusion equation is:

$$dX_t = \mu(X_t)\, dt + \sigma\, dW \tag{8}$$

When reversing the process, the biggest challenge is estimating the term:

$$\sigma^2 \nabla \log p_{T-s}(X_s^{\text{rev}}) \tag{9}$$

This estimation can be approached through the Fokker-Planck equation.

Fokker-Planck Equation

The Fokker-Planck equation describes the time evolution of the probability distribution $p_t(x)$:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (\mu p_t) + \frac{\sigma^2}{2} \Delta p_t \tag{10}$$

In the reverse process, define $\tilde{p}_t(x) \equiv p_{T-t}(x)$, then:

$$\frac{\partial \tilde{p}_t}{\partial t} = -\nabla \cdot (\tilde{\mu} \tilde{p}_t) + \frac{\sigma^2}{2} \Delta \tilde{p}_t \tag{11}$$

where $\tilde{\mu} = -\mu + \sigma^2 \nabla \log p_{T-t}$.

Limitations of Fokker-Planck and Marginals

Although the Fokker-Planck equation describes the change of the distribution in the diffusion process, it only provides the marginal distributions:

$$p_t(x) \text{ describes marginals at each time step, not the full process.} \tag{12}$$

To investigate the reverse process, we need to examine the reverse transition density:

$$p(x_t \mid x_{t+\delta}) \quad \text{with} \quad \delta \ll 1 \tag{13}$$

Estimating Reverse Transition Densities

The reverse transition density is given by:

$$p(x_t \mid x_{t+\delta}) = \frac{p(x_{t+\delta})p(x_{t+\delta} \mid x_t)}{p(x_t)} \tag{14}$$

Using Taylor expansion for $p(x_{t+\delta} \mid x_t)$, and the approximation:

$$\log p(x_{t+\delta}) \approx \log p(x_t) + \nabla \log p(x_t) \cdot (x_{t+\delta} - x_t) \tag{15}$$

allows us to analytically derive the reverse process.

Intuition for the Reverse Dynamics

Forward Process:

$$dX = \mu(X)\,dt + \sigma\,dW$$

Naive Reverse Process:

$$dX = -\mu(X)\,dt + \sigma\,dB$$

Problem: Reversing the drift term isn't enough to accurately reverse the diffusion process.

Corrective Term in Reverse Diffusion

The Necessary Correction:

$$dX_s^{\text{rev}} = -\mu(X_s^{\text{rev}})\, ds + \sigma^2 \nabla \log p_{T-s}(X_s^{\text{rev}})\, ds + \sigma\, dB_s$$

Intuition:

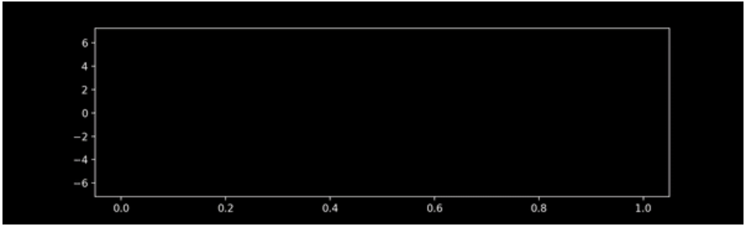
- The term $\sigma^2 \nabla \log p_{T-s}(X_s^{\text{rev}})$ acts like a "guiding force" that pulls the reversed trajectory back toward the data distribution.
- It gathers scattered diffusion and contracts it back along the correct path.

Naive Reverse Dynamics

- Consider the process $dX = \mu(X)dt + \sigma dW$.
- A "naive" reverse dynamics approach:

$$dX = -\mu(X)dt + \sigma dB$$

- The term $\sigma^2 \nabla \log p_{T-s}(X_s^{rev})ds$ acts as a force pulling the process back onto the correct trajectory.



Tweedie Formula

The Tweedie formula estimates the expected value of X_0 based on the observation y during the diffusion process.
Consider a Brownian motion $dX_t = \sigma dW$ with $X_0 \sim p_0(dx)$.
The reverse process equation is:

$$dX_s^{\text{rev}} = \sigma^2 \nabla \log p_{T-s}(X_s^{\text{rev}}) ds + \sigma dB$$

Assume $y = X_0 + \mathcal{N}(0, \sigma^2 \delta)$, where y is observed from X_0 plus Gaussian noise. The reverse process helps estimate the expected value of X_0 as:

$$\mathbb{E}[X_0 \mid y] \approx y + \sigma^2 \delta \nabla \log p_\delta(y)$$

Maurice Tweedie showed the formula holds exactly:

$$\mathbb{E}[X_0 \mid y] = y + \sigma^2 \delta \nabla \log p_\delta(y)$$

Tweedie Formula

The Tweedie formula estimates the expected value of X_0 based on the observation y during the diffusion process.
Consider a Brownian motion $dX_t = \sigma dW$ with $X_0 \sim p_0(dx)$.
The reverse process equation is:

$$dX_s^{\text{rev}} = \sigma^2 \nabla \log p_{T-s}(X_s^{\text{rev}})ds + \sigma dB$$

Assume $y = X_0 + \mathcal{N}(0, \sigma^2\delta)$, where y is observed from X_0 plus Gaussian noise. The reverse process helps estimate the expected value of X_0 as:

$$\mathbb{E}[X_0 \mid y] \approx y + \sigma^2 \delta \nabla \log p_\delta(y)$$

Maurice Tweedie showed the formula holds exactly:

$$\mathbb{E}[X_0 \mid y] = y + \sigma^2 \delta \nabla \log p_\delta(y)$$

Score Estimation

Sampling from $p_t(dx)$ is easy and does not require solving ODEs or SDEs: **simulation-free**

$$X_t = \alpha_t X_0 + \sqrt{1-\alpha_t^2}\mathbf{n}$$

The score function $S(t,x)$ is often parameterized by a neural network and can be represented as:

$$\nabla \log p_t(x) = \nabla \log \int P(X_t \in dx | X_0 = x_0)\pi_{\text{data}}(dx_0)$$

The forward distribution of the OU diffusion process is:

$$P(X_t \in dx | X_0 = x_0) = \mathcal{N}(\alpha_t x_0, (1-\alpha_t^2)I)$$

The score function can be written as:

$$S(t,x) = \nabla_x \log \left(\int P(X_t \in dx | X_0 = x_0)\pi_{\text{data}}(dx_0) \right)$$

Score Function Approximation

This can be expressed as:

$$S(t,x) = \nabla \log \int \exp\left\{-\frac{(x-\alpha_t x_0)^2}{2(1-\alpha_t^2)}\right\} \pi_{\text{data}}(dx_0)$$

Analytically, this results in:

$$S(t,x) = -\frac{\int \frac{(x-\alpha_t x_0)}{(1-\alpha_t^2)} p(X_t = x, X_0 = x_0) dx_0}{\int p(X_t = x, X_0 = x_0) dx_0}$$

This represents a conditional expectation:

$$S(t,x) = -E\left[\frac{x-\alpha_t X_0}{1-\alpha_t^2} \mid X_t = x\right]$$

- The score function is computed by taking the expectation of the deviation between x (the value at time t) and $\alpha_t X_0$ (the original value blurred by α_t). - The variance of the noise, $(1-\alpha_t^2)$, serves as a normalization factor for this deviation.

Regression Problem

Consider a regression problem with two random variables $(X, Y) \in \mathbb{R}^{d_X} \times \mathbb{R}^{d_Y}$.
$F_\star(x) = E[Y \mid X = x]$ is a function $F_\star(x) : \mathbb{R}^{d_X} \to \mathbb{R}^{d_Y}$ where, for each value x, the value of $F_\star(x)$ is the expected value of Y given $X = x$.
The goal of regression is to find a function F that minimizes the mean squared error between Y and $F(X)$:

$$F^\star = \underset{F}{\operatorname{argmin}} E[\|Y - F(X)\|_2]$$

To approximate the conditional expectation, parameterize with a neural network and find the function using stochastic gradient descent.

$$\theta \mapsto E[\|Y - F_\theta(X)\|^2] \approx \frac{1}{N}\sum_{i=1}^{N}\|y_i - F_\theta(x_i)\|^2$$

Building a Sampler from a Denoiser

We can rewrite the score as:

$$\mathcal{S}(t, x) = \nabla_x \log p_t(x) = -\frac{x - \alpha_t \hat{x}_0(t, x)}{1 - \alpha_t^2}$$

where $\hat{x}_0(t, x)$ is the estimate of X_0 given that $X_t = x$:

$$\hat{x}_0(t, x) = \mathbb{E}[X_0 \mid X_t = x]$$

To estimate the score, we need to build a denoising function:

$$\hat{x}_0(\ldots) : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$$

Constructing this function is a straightforward regression problem.

Building the Denoiser

The estimate can be expressed as:

$$\hat{x}_0(t, x) = \mathbb{E}[X_0 \mid X_t = x]$$

It can be easily simulated from (X_0, X_t):

$$x_0 \sim \pi_{data}(dx) \quad \text{and} \quad x_t = \alpha_t x_0 + \sqrt{1 - \alpha_t^2}\mathbf{n}$$

We parameterize $\hat{x}_0 : [0, T] \times \mathbb{R}^d \times \mathbb{R}^d$ with a neural network and minimize:

$$\mathcal{L}(\theta) \equiv \mathbb{E}\left\{\int_{t=0}^{T} \lambda(t)\|X_0 - \hat{x}_0(\tau, X_t)\|^2 dt\right\}$$

- $\hat{x}_0(t, X_t)$ is the model's prediction of the original value X_0 based on X_t.
- $\lambda(t)$ is a weighting function used to adjust the priority at different time points t during the diffusion process.
- $\|X_0 - \hat{x}_0(\tau, X_t)\|^2$ is the squared deviation between the prediction and the actual value.

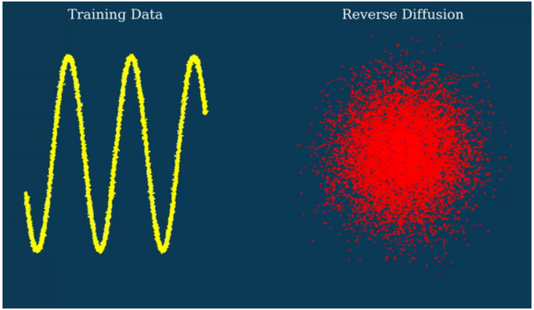
Score Definition

The score is defined as:

$$S(t,x) = -x - \frac{\alpha_t \hat{x}_0(t,x)}{1-\alpha_t^2} = \nabla \log p_t(x)$$

The reverse equation is:

$$dX_s^{rev} = \frac{1}{2} X_s^{rev} ds + S(T-s, X_s^{rev}) ds + dB$$



Other Parametrization

Instead of estimating X_0, we can estimate the noise:

$$x = \mathbb{E}\left[\alpha_t X_0 + \sqrt{1-\alpha_t^2} n \mid X_t = x\right]$$

This can be rewritten as:

$$x = \alpha_t \hat{x}_0(t,x) + \sqrt{1-\alpha_t^2}\hat{\mathbf{n}}(t,x)$$

where $\hat{\mathbf{n}}(t,x) = \mathbb{E}[\mathbf{n} \mid X_t = x]$.
Estimating X_0 or the noise $\mathbf{n}$ is equivalent.

ODE-Diffusion Trick

In addition to conventional diffusion processes, ODEs can be used more effectively for modeling.

Consider a diffusion process:

$$dX = \mu(t, X)dt + dB$$

The evolution of the marginal distribution $p_t(x)$ of X_t follows the Fokker-Planck equation:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (\mu p_t) + \frac{1}{2}\Delta p_t$$

Now, if we consider an ODE:

$$dX_t^{ODE} = \vec{F}(t, X_t^{ODE})dt$$

The evolution of the marginal distribution for the ODE is given by:

$$\frac{\partial p_t^{ODE}}{\partial t} = -\nabla \cdot (\mu p_t^{ODE} \vec{F})$$

We can find $\vec{F}(t, x)$ such that the marginal distribution of the ODE $p_t^{ODE}(x) = p_t(x)$ at all points is:

ODE-Diffusion Trick Introduction

In addition to conventional diffusion processes, ODEs can be used more effectively for modeling.

Consider a diffusion process:

$$dX = \mu(t, X)dt + dB$$

The evolution of the marginal distribution $p_t(x)$ of X_t follows the Fokker-Planck equation:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (\mu p_t) + \frac{1}{2}\Delta p_t$$

Now, if we consider an ODE:

$$dX_t^{ODE} = \vec{F}(t, X_t^{ODE})dt$$

Then the evolution of the marginal distribution is:

$$\frac{\partial p_t^{ODE}}{\partial t} = -\nabla \cdot (\mu p_t^{ODE} \vec{F})$$

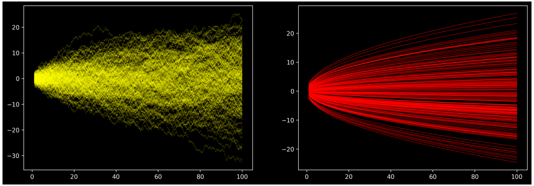
Finding the ODE Function

We can find $\vec{F}(t, x)$ such that the marginal distribution of the ODE $p_t^{ODE}(x) = p_t(x)$ at all points is:

$$\vec{F}(t, x) = \mu(t, x) - \frac{1}{2}\nabla_x \log p_t(x)$$

For Brownian motion where $dX = dW$, the corresponding ODE with an equivalent marginal distribution is:

$$dX_t^{ODE} = \frac{1}{2}\frac{X_t^{ODE}}{t}dt$$



Reversed ODE Process

Essentially, we also have a reversed ODE process:

$$dX_s^{\text{rev}} = \frac{1}{2} X_s^{\text{rev}} ds + \frac{1}{2} \nabla \log p_{T-s}(X_s^{\text{rev}}) ds$$