# Lecture 15 - Sampling and Diffusion Models

赵尉辰

南开大学 统计与数据科学学院

# 目录

## 采样问题

**采样(Sampling)**

设 $\mu$ 是一个概率分布，采样问题是指：如何获得随机样本$X$，使得$X$的分布为$\mu$。

- 计算：Monte Carlo 方法；
- 优化：模拟退火；
- 生成式AI；
- …

# 生成

**生成任务**

在机器学习中，生成任务 是指模型的目标为生成新的数据实例，这些实例与已有数据(训练数据)具有相似的特征或模式，常见的生成任务包括文本生成、图像生成、视频生成等。

- 从给定概率密度中采样：

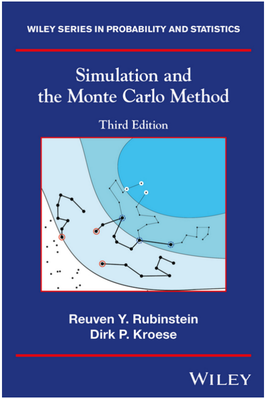
  给定函数$H : \mathcal{X}^d \to \mathbb{R}$，如何获得服从概率分布$\mu(x) \propto e^{-H(x)}$的随机样本？

- 从给定数据中采样：

  给定数据样本$X_1, X_2, \ldots, X_n \sim p_{data}$，如何生成服从数据分布$p_{data}$的数据样本？

# 目录

# 随机模拟




- 随机数的模拟
- 基本随机变量的模拟
- 随机过程/随机向量的模拟

# Markov Chain Monte Carlo (MCMC)

**定义 1 (Markov链)**

一个*Markov链*是一个随机过程 $\{X_n\}_{n=0}^{\infty}$，满足：未来状态只依赖于当前状态，而与过去状态无关。即，对于任意的 $n$ 和状态 $i, j$，有：

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = P(X_{n+1} = j \mid X_n = i)$$

进一步，如果转移概率不随时间变化，即：

$$P(X_{n+1} = j \mid X_n = i) = P(X_1 = j \mid X_0 = i) \quad \forall n$$

那么我们称*Markov*链是时齐的。

# Markov Chain Monte Carlo (MCMC)

**定义 2 (平稳分布)**

设 $\{X_n\}_{n=0}^{\infty}$ 是一个*Markov*链，其状态空间为 $S$。如果存在一个概率分布 $\pi$ 满足以下条件：

$$\pi(j) = \sum_{i \in S} \pi(i) P(i,j) \quad \forall j \in S$$

其中 $P(i,j)$ 是从状态 $i$ 转移到状态 $j$ 的概率，则称 $\pi$ 为该*Markov*链的平稳分布*(Stationary Distribution)*/不变测度*(Invariant Measure)*。

MCMC的思想即是：构造一个Markov链，使得它的平稳分布是我们采样的目标分布$\pi$。那么从任意状态分布(容易获得样本的分布)出发，经过充分的状态转移，就能获得目标分布$\pi$的样本。

## Metropolis-Hastings 算法

1. 选择初始状态 $x_0$。
2. 对于每一步 $n = 1, 2, \ldots, N$:
   - 从提议分布 $q(x'|x_{n-1})$ 中生成候选状态 $x'$。
   - 计算接受概率：
   $$\alpha = \min\left(1, \frac{\pi(x')q(x_{n-1}|x')}{\pi(x_{n-1})q(x'|x_{n-1})}\right)$$
   - 以概率 $\alpha$ 接受候选状态 $x'$，否则保持状态 $x_{n-1}$。

设 $X_{n-1}$ 是当前状态，$X_n$ 是下一个状态，Metropolis-Hastings 算法转移概率可以表示为：

$$P(X_n = x'|X_{n-1} = x) = q(x'|x) \cdot \min\left(1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right)$$

容易check Metropolis-Hastings 算法的平稳分布是目标分布$\pi$

$$\pi(x) = \sum_y \pi(y)q(x|y) \cdot \min\left(1, \frac{\pi(x)q(y|x)}{\pi(y)q(x|y)}\right)$$

# 目录

# Langevin Dynamics

**定义 3 (Langevin Dynamics)**

给定势能函数*(potential)* $V(x)$，*Langevin Dynamics*是如下形式的*SDE*

$$\mathrm{d}X_t = -\nabla V(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t. \tag{1}$$

其解一般称之为*Langevin Diffusion.*

Langevin 扩散的生成元为

$$\mathcal{L}_{LD} f = -\nabla V \cdot \nabla f + \Delta f$$

生成元的伴随为

$$\mathcal{L}_{LD}^{*} g = \nabla \cdot (g \nabla V) + \Delta g$$

# Langevin Dynamics

Kolmogorov backward方程:

$$\frac{\partial}{\partial t}P_t f(x) = \mathcal{L}_{LD} P_t f(x) = -\nabla V(x) \cdot \nabla P_t f(x) + \Delta P_t f(x)$$

Fokker-Planck方程:

$$\partial_t \mu(x,t) = \mathcal{L}_{LD}^{*} \mu(x,t) = \nabla \cdot (\mu(x,t)\nabla V(x)) + \Delta \mu(x,t)$$

**命题 1**

*Langevin* 扩散 $\mathrm{d}X_t = -\nabla V(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$ 的不变测度为

$$d\pi(x) \propto e^{-V(x)} dx$$

# Langevin Algorithm

**定义 4 (Langevin Algorithm)**

对*Langevin*扩散*Euler–Maruyama*离散化：

$$X_{(k+1)h} := X_{kh} - h\nabla V(X_{kh}) + \sqrt{2}(B_{(k+1)h} - B_{kh}).$$

我们得到了一种*Langevin*扩散的实现方式，称为*(Unadjusted) Langevin Algorithm, ULA*，或者 *Langevin Monte Carlo, LMC*. 其中$h$是迭代步长，$k$是迭代轮数。

由于时间离散化，Langevin Monte Carlo与Langevin Dynamics不再一致，Langevin Monte Carlo的平稳分布也不再是目标分布。一般可以通过Metropolis调整保证采样分布的准确性。

# Metropolis-adjusted Langevin Algorithm (MALA)

## MALA

- Proposal step: same as in ULA

$$Z_{k+1} = X_k - h\nabla V(X_k) + \sqrt{2h}\xi_k$$

- Accept-reject step: go to

$$X_{k+1} = \begin{cases} Z_{k+1} & \text{with probability} \min\left\{1, \frac{\pi(Z_{k+1})\mathcal{P}_{Z_{k+1}}(X_k)}{\pi(X_k)\mathcal{P}_{X_k}(Z_{k+1})}\right\} \\ X_k & \text{with the remaining probability.} \end{cases}$$

注意到给定$X_k$，提议分布是一个均值为 $X_k - h\nabla V(X_k)$，方差为 $2h\mathbb{I}_n$ 的高斯分布，即提议分布显式表达为

$$\mathcal{P}_z(x) = \frac{1}{(2\pi \cdot 2h)^{\frac{n}{2}}} \exp\left(-\frac{\|x - (z - h\nabla V(z))\|_2^2}{4h}\right).$$

接受概率也具有显式表达：

$$\min\left\{1, \exp\left(-V(z) - \frac{1}{4h}\|x - (z - h\nabla V(z))\|_2^2 + V(x) + \frac{1}{4h}\|z - (x - h\nabla V(x))\|_2^2\right)\right\}$$

# 目录

# 耦合方法

**定义 5 (Wasserstein distance)**

概率测度$\mu$和$\nu$之间的*2-Wasserstein Distance*定义为：

$$W_2(\mu,\nu) := \inf_{\gamma\in\mathcal{C}(\mu,\nu)} \left(\int \|x-y\|^2 \gamma(\mathrm{d}x,\mathrm{d}y)\right)^{\frac{1}{2}}. \tag{2}$$

其中$\mathcal{C}(\mu,\nu)$是$\mu$和$\nu$的耦合*(Couplings)*构成的空间，$\|\cdot\|$是欧式范数。

**定理 1**

设$\{X_t\}$为初值为$X_0\sim\mu_0$，平稳分布为$\mu\propto e^{-V}$的*Langevin*扩散，假设$\mu$是$\alpha$-强*log-concave*的，那么

$$W_2^2(\mu_t,\mu)\le \exp(-2\alpha t) W_2^2(\mu_0,\mu).$$

# 耦合方法

**定理 2**

对于 $k \in \mathbb{N}$, 记$\mu_{kh}$为*LMC*的的第*k*轮迭代的分布， $h > 0$为迭代步长。设目标分布为 $\mu \propto e^{-V}$，满足 $\alpha I_d \le \nabla^2 V \le \beta I_d$. 如果 $h \lesssim \frac{1}{\beta\kappa}$，那么对于所有 $N \in \mathbb{N}$,

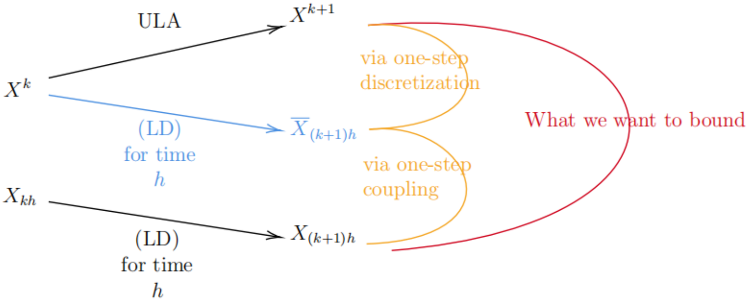
$$W_2(\mu_{Nh}, \mu) \le \exp\left(-\frac{\alpha N h}{2}\right) W_2(\mu_0, \mu) + O\left(\frac{\beta d^{1/2} h^{1/2}}{\alpha}\right).$$

如果 $h = O(\frac{\varepsilon^2}{\beta\kappa d})$, 那么对于任意 $\varepsilon \in [0, \sqrt{d}]$, 在

$$N = O\Big(\frac{\kappa^2 d}{\varepsilon^2} \log \frac{\sqrt{\alpha} W_2(\mu_0, \mu)}{\varepsilon}\Big)$$

轮迭代之后，有 $\sqrt{\alpha} W_2(\mu_{Nh}, \mu) \le \varepsilon$.

# Proof sketch[1]




- 计算ULA和LD之间的一步时间离散化误差；
- 通过LD在$W_2$距离下的指数压缩性分析多步迭代误差。

[1] https://chewisinho.github.io/main.pdf Sec.4.1

# 泛函不等式方法[2]

耦合方法的分析要求目标分布的log-concavity，泛函不等式方法可以减弱这一假设。

**定义 6 (Log-Sobolev inequality)**

称$\nu$满足$\alpha$ *Log-Sobolev inequality*，如果对于$\mathbb{E}_\nu[g^2] < \infty$的光滑函数$g : \mathbb{R}^n \to \mathbb{R}$，

$$\mathrm{Ent}_\nu(g) \triangleq \mathbb{E}_\nu[g^2 \log g^2] - \mathbb{E}_\nu[g^2] \log \mathbb{E}_\nu[g^2] \leq \frac{2}{\alpha}\mathbb{E}_\nu[\|\nabla g\|^2]. \tag{3}$$

**定义 7 (Poincaré inequality)**

称$\nu$满足$\alpha$ *Poincaré inequality*，如果对于光滑函数$g : \mathbb{R}^n \to \mathbb{R}$, 有

$$\mathrm{Var}_\nu(g) \triangleq \mathbb{E}_\nu[g^2] - \mathbb{E}_\nu[g]^2 \leq \frac{1}{\alpha}\mathbb{E}_\nu[\|\nabla g\|^2]. \tag{4}$$

[2] Bakry D, Gentil I, Ledoux M. Analysis and geometry of Markov diffusion operators[M]. Cham: Springer, 2014.

# KL散度

**定义 8 (KL散度)**

$\mu$对于$\nu$的*KL*散度定义为

$$\mathrm{KL}(\mu\|\nu) = H_\nu(\mu) = \int_{\mathbb{R}^n} \mu(x) \log \frac{\mu(x)}{\nu(x)} dx. \tag{5}$$

**命题 2 (Pinsker's inequality)**

$$\mathrm{d_{TV}}(\mu,\nu)^2 \le \tfrac{1}{2} H_\nu(\mu).$$

**命题 3 (Talagrand inequality)**

若$\nu$满足$\alpha$ *Log-Sobolev inequality*

$$\tfrac{\alpha}{2} W_2(\mu,\nu)^2 \le H_\nu(\mu).$$

Pinsker's inequality和Talagrand inequality说明KL散度是一个相对更强的距离度量，我们bound KL散度自然能够给出TV和$W_2$距离的界。

# KL散度+LSI下Langevin Dynamics的指数收敛性

**定理 3**

若$\nu \propto e^{-V}$满足$\alpha$ *LSI*，那么*Langevin Dynamics*

$$dX_t = -\nabla V(X_t)dt + \sqrt{2}dW_t$$

的分布$\mu_t$满足：

$$H_\nu(\mu_t) \le e^{-2\alpha t} H_\nu(\mu_0).$$

进一步地，$W_2(\mu_t, \nu) \le \sqrt{\frac{2}{\alpha} H_\nu(\mu_0)} e^{-\alpha t}$.

## Proof sketch[3]

**定义 9 (Fisher information)**

$\mu$对于$\nu$的*Fisher information*定义为

$$J_\nu(\mu) = \int_{\mathbb{R}^n} \mu(x) \left\| \nabla \log \frac{\mu(x)}{\nu(x)} \right\|^2 dx. \tag{6}$$

令$g^2 = \frac{\mu}{\nu}$，Log-Sobolev inequality可以得到KL散度和Fisher information的如下关系：

$$H_\nu(\mu) \le \frac{1}{2\alpha} J_\nu(\mu).$$

[3] Vempala S, Wibisono A. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices[J]. Advances in neural information processing systems, 2019, 32.

## Proof sketch

**引理 1**

分布$\mu_t$满足：

$$\frac{d}{dt}H_\nu(\mu_t) = -J_\nu(\mu_t). \tag{7}$$

利用*Langevin Dynamics*的*Fokker-Planck*方程：$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla V(x)) + \Delta \mu_t$ 计算可得。

由Log-Sobolev inequality，

$$H_\nu(\mu) \leq \frac{1}{2\alpha} J_\nu(\mu).$$

结合(7)式，有

$$\frac{d}{dt}H_\nu(\mu_t) \leq -2\alpha H_\nu(\mu_t)$$

两边积分有

$$H_\nu(\mu_t) \leq e^{-2\alpha t} H_\nu(\mu_0).$$

# KL散度+LSI下LMC的指数收敛性[4]

**定理 4**

若$\nu := e^{-V}$满足$\alpha$ *LSI* 并且是 *L-smooth*的($-LI \preceq \nabla^2 V(x) \preceq LI$ for all $x \in \mathbb{R}^n$)，那么对于任意$x_0 \sim \mu_0$满足$H_\nu(\mu_0) < \infty$，步长$0 < \eta \leq \frac{\alpha}{4L^2}$的*ULA*

$$x_{k+1} = x_k - \eta \nabla V(x_k) + \sqrt{2\eta} z_k$$

的分布$x_k \sim \mu_k$满足：

$$H_\nu(\mu_k) \leq e^{-\alpha \eta k} H_\nu(\mu_0) + \frac{8\eta n L^2}{\alpha}.$$

因此，对任意精度$\delta > 0$，为了$H_\nu(\mu_k) < \delta$，*LMC*需要满足步长$\eta \leq \frac{\alpha}{4L^2} \min\{1, \frac{\delta}{4n}\}$，并且经过$k \geq \frac{1}{\alpha\eta} \log \frac{2H_\nu(\mu_0)}{\delta}$次迭代。

[4] Vempala S, Wibisono A. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices[J]. Advances in neural information processing systems, 2019, 32.

## Proof sketch

- 给出一步LMC迭代的界

**引理 2**

若$\nu := e^{-V}$满足$\alpha$ *Log-Sobolev inequality* 并且 *L-smooth*，步长$0 < \eta \leq \frac{\alpha}{4L^2}$，那么*LMC*满足：

$$H_\nu(\mu_{k+1}) \leq e^{-\alpha\eta} H_\nu(\mu_k) + 6\eta^2 nL^2.$$

$$\frac{d}{dt} H_\nu(\mu_t) \leq -\frac{3}{4} J_\nu(\mu_t) + \frac{4t^2L^4}{\alpha} H_\nu(\mu_0) + 2t^2nL^3 + 2tnL^2.$$

由Log-Sobolev inequality

$$\frac{d}{dt} H_\nu(\mu_t) \leq -\frac{3\alpha}{2} H_\nu(\mu_t) + \frac{4t^2L^4}{\alpha} H_\nu(\mu_0) + 2t^2nL^3 + 2tnL^2.$$

$t = 0$到$t = \eta$积分，整理可得引理2。

- 给出多步迭代的界

# Rényi散度+PI下Langevin Dynamics的指数收敛性

**定义 10 (Rényi散度)**

对于$q > 0$, $q \neq 1$，概率分布$\mu$对于$\nu$的$q$阶*Rényi*散度定义为：

$$R_{q,\nu}(\mu) := \frac{1}{q-1} \log F_{q,\nu}(\mu), \tag{8}$$

其中

$$F_{q,\nu}(\mu) := \mathbb{E}_\nu \left[ \left( \frac{\mu}{\nu} \right)^q \right] = \int_{\mathbb{R}^n} \nu(x) \frac{\mu(x)^q}{\nu(x)^q} dx = \int_{\mathbb{R}^n} \frac{\mu(x)^q}{\nu(x)^{q-1}} dx.$$

Rényi散度来源于Rényi熵：$H_q(\mu) := \frac{1}{q-1} \log \int \mu(x)^q dx.$

**定理 5**

若$\nu := e^{-f}$满足$\alpha$ *Poincaré inequality*，$q \geq 2$，那么*Langevin Dynamics*的分布$\mu_t$满足：

$$R_{q,\nu}(\mu_t) \leq \begin{cases} R_{q,\nu}(\mu_0) - \frac{2\alpha t}{q} & if R_{q,\nu}(\mu_0) \geq 1 \text{ and as long as } R_{q,\nu}(\mu_t) \geq 1, \\ e^{-\frac{2\alpha t}{q}} R_{q,\nu}(\mu_0) & if R_{q,\nu}(\mu_0) \leq 1. \end{cases}$$

# Rényi散度+PI下LMC的指数收敛性

**定理 6**

若$\nu_\eta$满足$\beta$ *Poincaré inequality*，$q \geq 1$，$\nu := e^{-V}$是 *L-smooth*的，且$1 \leq R_{2q,\nu_\eta}(\mu_0) < \infty$，令$0 < \eta \leq \min\left\{\frac{1}{3L}, \frac{1}{9\beta}\right\}$，$q > 1$，那么对于$k \geq k_0 := \frac{2q}{\beta\eta}(R_{2q,\nu_\eta}(\mu_0) - 1)$，*LMC*满足：

$$R_{q,\nu}(\mu_k) \leq \left(\frac{q-\frac{1}{2}}{q-1}\right) e^{-\frac{\beta\eta(k-k_0)}{2q}} + R_{2q-1,\nu}(\nu_\eta).$$

对任意精度$\delta > 0$，为了$R_{q,\nu}(\mu_k) \leq \delta$，*LMC*需要满足步长$\eta = \Theta\left(\min\left\{\frac{1}{L}, \gamma_{2q-1}\left(\frac{\delta}{2}\right)\right\}\right)$，其中$\gamma_q(\delta) = \sup\{\eta > 0\colon R_{q,\nu}(\nu_\eta) \leq \delta\}$，并且经过$k = \Theta\left(\frac{1}{\beta\eta}\left(R_{2q,\nu_\eta}(\mu_0) + \log\frac{1}{\delta}\right)\right)$次迭代。

# 目录

# 生成——从数据分布中采样

数据分布是未知的，我们仅有一些样本，扩散模型的基本思想是：

(1) 学习数据分布；(2) 根据学习到的数据分布生成实例

# SMLD

Recall that Langevin dynamics

$$\mathrm{d}X_t = -\nabla V(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

具有不变测度$\pi \propto e^{-V}$，若我们需要采样$p_{data}$，可以通过

$$\mathrm{d}X_t = \nabla \log p_{data}(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t.$$

其中$\nabla \log p$称为概率分布$p$的Score function.

如果$p_{data}$已知，那么可以显式计算Score，然而生成任务中，我们需要从数据中学习Score $\nabla \log p_{data}$，通过神经网络近似

$$\min_\theta \mathbb{E}[\|\nabla \log p_{data}(X) - s_\theta(X)\|_2^2]$$

其中$s_\theta$为参数为$\theta$的神经网络。

# Score Matching

- Score matching[5]

$$
\begin{aligned}
&\mathbb{E}_{X\sim p_{data}} \|\nabla \log p_{data}(X) - s_\theta(X)\|_2^2 \\
&= \underbrace{\mathbb{E}\,\|\nabla \log p_{data}(X)\|_2^2}_{\text{does not depend on } \theta} - 2\mathbb{E}\,\langle s_\theta(X), \nabla \log p_{data}(X)\rangle + \mathbb{E}\,\|s_\theta(X)\|_2^2 .
\end{aligned}
$$

计算第二项

$$
\begin{aligned}
-\mathbb{E}\,\langle s_\theta(X), \nabla \log p_{data}(X)\rangle &= -\int \langle s_\theta(x), \nabla \log p_{data}(x)\rangle p_{data}(x)dx \\
&= \int \nabla \cdot s_\theta(x) p_{data}(x)dx = \mathbb{E}\nabla \cdot s_\theta(X),
\end{aligned}
$$

那么我们的优化问题即为：

$$
\min_\theta \mathbb{E}_{X\sim p_{data}} \left[\|s_\theta(X)\|_2^2 + 2\nabla \cdot s_\theta(X)\right] .
$$

[5] Hyvärinen A, Dayan P. Estimation of non-normalized statistical models by score matching[J]. Journal of Machine Learning Research, 2005, 6(4).

# Denoising score matching

实际训练神经网络优化经验风险函数：

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left[ \|s_\theta(x_i)\|_2^2 + 2\nabla \cdot s_\theta(x_i) \right].$$

然而高维情形计算散度项$\nabla \cdot s_\theta(x_i)$比较困难，考虑通过Denoising score matching避免散度的计算。

- Denoising score matching[6]

考虑扰动$\tilde{x} = x + \sigma z$, 其中$z \sim N(0, I)$，Denoising score matching的目标为

$$\min_{\theta} \mathbb{E}_{q_\sigma(\tilde{x}|x) p_{data}(x)} [\|s_{\boldsymbol{\theta}}(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x)\|_2^2].$$

可以证明$s_{\theta^*}(\tilde{x}) = \nabla_{\tilde{x}} \log q_\sigma(\tilde{x})$几乎处处成立[7]，其中$q_\sigma(\tilde{x}) \triangleq \int q_\sigma(\tilde{x} \mid x) p_{\mathrm{data}}(x) \mathrm{d}x$.

[6] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In Advances in Neural Information Processing Systems, pp. 11895–11907, 2019.

[7] Vincent P. A connection between score matching and denoising autoencoders[J]. Neural computation, 2011, 23(7): 1661-1674.

## Denoising score matching

虽然只有在$\sigma$比较小的时候，有

$$s_{\theta^*}(\tilde{x}) = \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}) \approx \nabla_x \log p_{\text{data}}(x)$$

但是$q_\sigma(\tilde{x} \mid x) \sim \mathcal{N}(x, \sigma^2 I)$是条件高斯的在计算上十分高效

$$\begin{aligned}
\nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x) &= \nabla_{\tilde{x}} \log \frac{1}{(\sqrt{2\pi\sigma^2})^d} \exp\left\{-\frac{\|\tilde{x}-x\|^2}{2\sigma^2}\right\} \\
&= \nabla_{\tilde{x}} \left\{-\frac{\|\tilde{x}-x\|^2}{2\sigma^2} - \log(\sqrt{2\pi\sigma^2})^d\right\} \\
&= -\frac{\tilde{x}-x}{\sigma^2}.
\end{aligned}$$

Denoising score matching的优化问题即为：

$$\min_\theta \mathbb{E}_{x\sim p_{data}, \tilde{x}\sim\mathcal{N}(x,\sigma^2 I)} \left\| s_\theta(\tilde{x}, \sigma) + \frac{\tilde{x}-x}{\sigma^2} \right\|_2^2. \tag{9}$$

# Noise Conditional Score Networks

(9)中参数化的神经网络模型$s_\theta(x,\sigma)$称为 Noise Conditional Score Networks。由于只有在$\sigma$比较小的时候，有

$$s_{\theta^*}(x,\sigma) \approx \nabla_x \log p_{\mathrm{data}}(x)$$

考虑设计一个time schedule，使得$\sigma_t \to 0$.

**Algorithm 1** Annealed Langevin dynamics.

```
Require: {σ_i}_{i=1}^L, ε, T.
  Initialize x̃_0
  for i ← 1 to L do
      α_i ← ε · σ_i^2 / σ_L^2          ▷ α_i is the step size.
      for t ← 1 to T do
          Draw z_t ~ N(0, I)
          x̃_t ← x̃_{t-1} + (α_i/2) s_θ(x̃_{t-1}, σ_i) + sqrt(α_i) z_t
      end for
      x̃_0 ← x̃_T
  end for
  return x̃_T
```

**Require:** $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.

Initialize $\tilde{\mathbf{x}}_0$

**for** $i \leftarrow 1$ to $L$ **do**

  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$  ▷ $\alpha_i$ is the step size.

  **for** $t \leftarrow 1$ to $T$ **do**

    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$

    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$

  **end for**

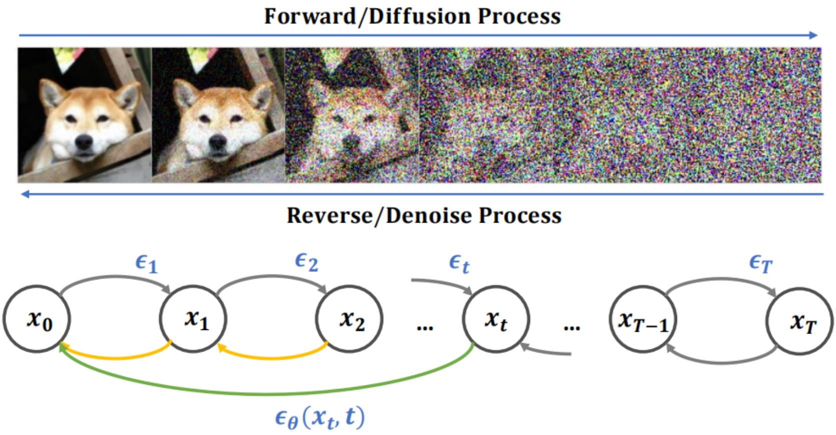
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$

**end for**

**return** $\tilde{\mathbf{x}}_T$

# 目录

# Denoising Diffusion Probabilistic Models, DDPM[8]




[8] Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33, 2020

# Forward Diffusion Process

- 一步加噪过程

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{(1-\alpha_t)}\boldsymbol{\epsilon}_{t-1}, \quad \text{where } \boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(0, \mathbf{I}).$$

- $t$步加噪过程

命题 4

条件分布$q(\boldsymbol{x}_t|\boldsymbol{x}_0)$为

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\mathbf{I}),$$

其中 $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. 即 $\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}_0$.

能够计算$q(\boldsymbol{x}_t|\boldsymbol{x}_0)$的好处在于给定$\boldsymbol{x}_0$，给一个$t$可以直接得到$\boldsymbol{x}_t$.

# Proof

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \underbrace{\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}}_{\mathbf{w}_1}.
\end{aligned}
$$

由于$\boldsymbol{\epsilon}_{t-2}$和$\boldsymbol{\epsilon}_{t-1}$都是标准高斯的，$\mathbf{w}_1$是均值为0的高斯，我们下面计算协方差

$$
\begin{aligned}
\mathbb{E}[\mathbf{w}_1\mathbf{w}_1^T] &= [(\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}})^2 + (\sqrt{1-\alpha_t})^2]\mathbf{I} \\
&= [\alpha_t(1-\alpha_{t-1}) + 1 - \alpha_t]\mathbf{I} = [1-\alpha_t\alpha_{t-1}]\mathbf{I}.
\end{aligned}
$$

延用记号$\boldsymbol{\epsilon}_t$

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} \\
&= \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\mathbf{x}_{t-3} + \sqrt{1-\alpha_t\alpha_{t-1}\alpha_{t-2}}\boldsymbol{\epsilon}_{t-3} \\
&= \cdots = \sqrt{\prod_{i=1}^{t}\alpha_i}\mathbf{x}_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\boldsymbol{\epsilon}_0.
\end{aligned}
$$

# Reverse Denoising Process

我们希望用一个神经网络实现降噪过程，即

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx q(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

由Markov性，

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_t)}{q(\mathbf{x}_{t-1})} \xRightarrow{\text{condition on } \mathbf{x}_0} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}$$

在优化神经网络的过程中转化为[9][10]

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$$

[9] Luo C. Understanding diffusion models: A unified perspective[J]. arXiv preprint arXiv:2208.11970, 2022.
[10] Chan S H. Tutorial on Diffusion Models for Imaging and Vision[J]. arXiv preprint arXiv:2403.18103, 2024.

# Reverse Denoising Process

### 命题 5

条件分布$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$为一个高斯分布 $\mathcal{N}(\mathbf{x}_{t-1};\ \boldsymbol{\mu}_q(\mathbf{x}_t,\mathbf{x}_0),\boldsymbol{\Sigma}_q(t))$，其中

$$\mu_q(\mathbf{x}_t,\mathbf{x}_0)=\frac{(1-\overline{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\overline{\alpha}_t}\mathbf{x}_t+\frac{(1-\alpha_t)\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\mathbf{x}_0$$

$$\boldsymbol{\Sigma}_q(t)=\frac{(1-\alpha_t)(1-\sqrt{\overline{\alpha}_{t-1}})}{1-\overline{\alpha}_t}\mathbf{I}$$

$$\begin{aligned}
q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)&=\frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\\
&=\frac{\mathcal{N}(\boldsymbol{x}_t;\sqrt{\alpha_t}\boldsymbol{x}_{t-1},(1-\alpha_t)\mathbf{I})\mathcal{N}(\boldsymbol{x}_{t-1};\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0,(1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\boldsymbol{x}_t;\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0,(1-\bar{\alpha}_t)\mathbf{I})}\\
&\propto\exp\left\{-\left[\frac{(\boldsymbol{x}_t-\sqrt{\alpha_t}\boldsymbol{x}_{t-1})^2}{2(1-\alpha_t)}+\frac{(\boldsymbol{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_{t-1})}-\frac{(\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_t)}\right]\right\}
\end{aligned}$$

# Reverse Denoising Process

注意到，给定加噪schedule，$\boldsymbol{\Sigma}_q(t)$是已知的，所以我们只需要参数化均值部分，即

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q(t))$$

两个高斯分布之间的KL散度可以容易计算：

$$D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) = \frac{1}{2\sigma_q^2(t)} \left[ \left\| \boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q \right\|_2^2 \right]$$

注意到

$$\begin{aligned}
\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{(1-\overline{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\overline{\alpha}_t}\boldsymbol{x}_t + \frac{(1-\alpha_t)\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\boldsymbol{x}_0 \\
&= \frac{(1-\overline{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\overline{\alpha}_t}\boldsymbol{x}_t + \frac{(1-\alpha_t)\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}} \\
&= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0
\end{aligned}$$

# Denoising Diffusion Probabilistic Models

考虑

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

我们要学习的目标其实是一个 Denoiser $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$。

**Algorithm 1** Training

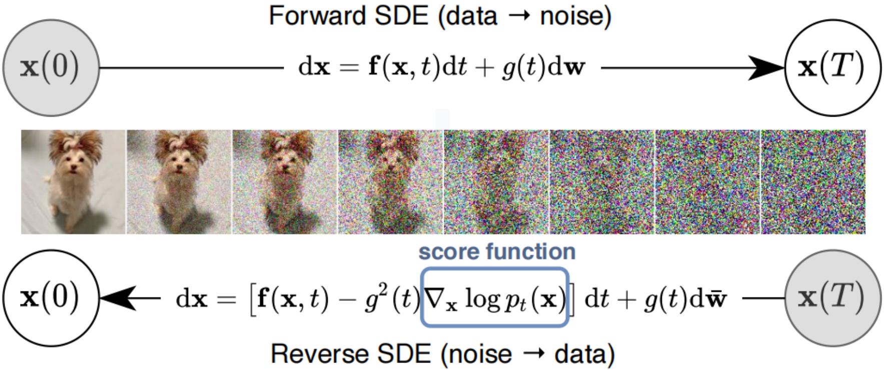
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

**Algorithm 2** Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

# 目录

# Score-based Generative Models, SGM[11]




[11] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In Proc. ICLR, 2021.

## Reverse SDE

**定理 7**

对于如下*SDE*:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{G}(\mathbf{x}, t)d\mathbf{w}, \tag{10}$$

它的 *Reverse SDE* 为

$$d\mathbf{x} = \{\mathbf{f}(\mathbf{x}, t) - \nabla \cdot [\mathbf{G}(\mathbf{x}, t)\mathbf{G}(\mathbf{x}, t)^{\mathsf{T}}] - \mathbf{G}(\mathbf{x}, t)\mathbf{G}(\mathbf{x}, t)^{\mathsf{T}} \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\}dt + \mathbf{G}(\mathbf{x}, t)d\bar{\mathbf{w}}$$

# Proof Sketch

SDE (10)的Fokker-Planck方程为

$$\begin{aligned}\frac{\partial p_t(\mathbf{x})}{\partial t} &= -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}[f_i(\mathbf{x},t)p_t(\mathbf{x})] + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\partial^2}{\partial x_i \partial x_j}\left[\sum_{k=1}^{d}G_{ik}(\mathbf{x},t)G_{jk}(\mathbf{x},t)p_t(\mathbf{x})\right] \\ &= -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}[f_i(\mathbf{x},t)p_t(\mathbf{x})] + \frac{1}{2}\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left[\sum_{j=1}^{d}\frac{\partial}{\partial x_j}\left[\sum_{k=1}^{d}G_{ik}(\mathbf{x},t)G_{jk}(\mathbf{x},t)p_t(\mathbf{x})\right]\right].\end{aligned}$$

注意到

$$\begin{aligned}&\sum_{j=1}^{d}\frac{\partial}{\partial x_j}\left[\sum_{k=1}^{d}G_{ik}(\mathbf{x},t)G_{jk}(\mathbf{x},t)p_t(\mathbf{x})\right] \\ &= \sum_{j=1}^{d}\frac{\partial}{\partial x_j}\left[\sum_{k=1}^{d}G_{ik}(\mathbf{x},t)G_{jk}(\mathbf{x},t)\right]p_t(\mathbf{x}) + \sum_{j=1}^{d}\sum_{k=1}^{d}G_{ik}(\mathbf{x},t)G_{jk}(\mathbf{x},t)p_t(\mathbf{x})\frac{\partial}{\partial x_j}\log p_t(\mathbf{x}) \\ &= p_t(\mathbf{x})\nabla\cdot[\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}] + p_t(\mathbf{x})\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\end{aligned}$$

# Proof Sketch

回代Fokker-Planck方程

$$
\begin{aligned}
\frac{\partial p_t(\mathbf{x})}{\partial t} &= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[f_i(\mathbf{x},t)p_t(\mathbf{x})] \\
&+ \frac{1}{2}\sum_{i=1}^{d} \frac{\partial}{\partial x_i}\Big[p_t(\mathbf{x})\nabla\cdot[\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}] + p_t(\mathbf{x})\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\Big] \\
&= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}\Big\{f_i(\mathbf{x},t)p_t(\mathbf{x}) \\
&- \frac{1}{2}\Big[\nabla\cdot[\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}] + \mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\Big]p_t(\mathbf{x})\Big\} \\
&\triangleq -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[\tilde{f}_i(\mathbf{x},t)p_t(\mathbf{x})],
\end{aligned}
$$

做时间逆转

$$
\frac{\partial p_t(\mathbf{x})}{\partial t} = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[-\tilde{f}_i(\mathbf{x},t)p_t(\mathbf{x})] \tag{11}
$$

# Proof Sketch

整理(11)，得

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[\bar{f}_i(\mathbf{x},t)p_t(\mathbf{x})] + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\left[\sum_{k=1}^{d} G_{ik}(\mathbf{x},t)G_{jk}(\mathbf{x},t)p_t(\mathbf{x})\right]$$

其中

$$\bar{\mathbf{f}}(\mathbf{x},t) = \mathbf{f}(\mathbf{x},t) - \nabla \cdot [\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}] - \mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$$

所以Reverse SDE 为

$$\mathrm{d}\mathbf{x} = \{\mathbf{f}(\mathbf{x},t) - \nabla \cdot [\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}] - \mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^{\mathsf{T}}\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\}\mathrm{d}t + \mathbf{G}(\mathbf{x},t)\mathrm{d}\bar{\mathbf{w}}$$

# Forward Process of DDPM & OU Process

考虑离散时间$i = 1, 2, \ldots, N$，DDPM的前向加噪过程

$$\mathbf{x}_i = \sqrt{1-\beta_i}\mathbf{x}_{i-1} + \sqrt{\beta_i}\mathbf{z}_{i-1}, \quad \mathbf{z}_{i-1} \sim \mathcal{N}(0, \mathbf{I}).$$

定义时间步长$\Delta t = \frac{1}{N}$，$t \in \{0, 1, \cdots, \frac{N-1}{N}\}$。加噪schedule为

$$\beta_i = \beta\left(\frac{i}{N}\right) \cdot \frac{1}{N} = \beta(t+\Delta t)\Delta t, \quad N \to \infty, \beta\left(\frac{i}{N}\right) \to \beta(t)$$

于是

$$\begin{aligned}
\mathbf{x}(t+\Delta t) &= \sqrt{1-\beta(t+\Delta t)\Delta t}\mathbf{x}(t) + \sqrt{\beta(t+\Delta t)\Delta t}\mathbf{z}(t) \\
&\approx \mathbf{x}(t) - \frac{1}{2}\beta(t+\Delta t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t+\Delta t)\Delta t}\mathbf{z}(t) \\
&\approx \mathbf{x}(t) - \frac{1}{2}\beta(t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t)\Delta t}\mathbf{z}(t),
\end{aligned}$$

当$\Delta t \to 0$，

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}.$$

# Denoiser和Score的联系

**引理 3 (Tweedie Formula)**

对于一个高斯随机变量$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$，有

$$\mathbb{E}\left[\boldsymbol{\mu}_z | \boldsymbol{z}\right] = \boldsymbol{z} + \boldsymbol{\Sigma}_z \nabla_z \log p(\boldsymbol{z})$$

在DDPM中，我们证明过

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

应用Tweedie Formula

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t} | \boldsymbol{x}_t\right] = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)$$

带入到$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}$中计算，有

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla \log p(\boldsymbol{x}_t)$$

# Denoiser和Score的联系

可以通过学习到的Score计算

$$\mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

又由

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}} = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}}$$

可以得到Denoiser和Score的联系

$$\nabla \log p(\boldsymbol{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_0$$

# 目录

# Reverse OU Process

- Forward process

$$\mathrm{d}\mathbf{X}_t = -\beta_t \mathbf{X}_t \mathrm{d}t + \sqrt{2\beta_t}\mathrm{d}\mathbf{B}_t$$

- Backward process (BP)

$$\mathrm{d}\mathbf{Y}_t = \beta_{T-t}\{\mathbf{Y}_t + 2\nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + \sqrt{2\beta_{T-t}}\mathrm{d}\mathbf{B}_t$$

# Diffusion Model 收敛性分析

- Girsanov 定理
  - Chen S, et al. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. ICLR. 2023.
  - Chen H, et al. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. ICML. 2023.
  - Benton J, et al. Nearly $d$-Linear Convergence Bounds for Diffusion Models via Stochastic Localization. ICLR. 2024
- Log-Sobolev inequality
  - Convergence for score-based generative modeling with polynomial complexity. NeuralPS. 2022.
  - Convergence of score-based generative modeling for general data distributions. 2023.
- 其他
  - A Note on the Convergence of Denoising Diffusion Probabilistic Models. TMLR. 2024.

## Reverse Diffusion Monte Carlo[12]

设采样目标为$x \propto e^{-f_*(x)}$，考虑 Reverse Diffusion Process

$$\mathrm{d}\mathbf{X}_t = \beta_{T-t}\{\mathbf{X}_t + 2\nabla \log p_{T-t}(\mathbf{X}_t)\}\mathrm{d}t + \sqrt{2\beta_{T-t}}\mathrm{d}\mathbf{B}_t$$

**引理 4**

*The score function can be rewritten as*

$$\nabla_{\boldsymbol{x}} \log p_{T-t}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{x}_0 \sim q_{T-t}(\cdot|\boldsymbol{x})} \frac{e^{-(T-t)}\boldsymbol{x}_0 - \boldsymbol{x}}{\left(1 - e^{-2(T-t)}\right)},$$

*where*

$$q_{T-t}(\boldsymbol{x}_0|\boldsymbol{x}) \propto \exp\left(-f_*(\boldsymbol{x}_0) - \frac{\left\|\boldsymbol{x} - e^{-(T-t)}\boldsymbol{x}_0\right\|^2}{2\left(1 - e^{-2(T-t)}\right)}\right).$$

[12] Huang X, Dong H, Yifan H A O, et al. Reverse diffusion monte carlo[C]//The Twelfth International Conference on Learning Representations. 2024.

# Reverse Diffusion Monte Carlo

**Algorithm 1** RDMC: reverse diffusion Monte Carlo

1: **Input:** Initial particle $\tilde{\boldsymbol{x}}_0$ sampled from $\tilde{p}_0$, Terminal time $T$, Step size $\eta, \eta'$, Sample size $n$.
2: **for** $k = 0$ to $\lfloor T/\eta \rfloor - 1$ **do**
3: Set $\boldsymbol{v}_k = \boldsymbol{0}$;
4: Create $n$ Monte Carlo samples to estimate

$$\boldsymbol{v}_k \approx \mathbb{E}_{\boldsymbol{x} \sim q_{T-t}} \left[ -\frac{\tilde{\boldsymbol{x}}_{k\eta} - e^{-(T-k\eta)}\boldsymbol{x}}{\left(1 - e^{-2(T-k\eta)}\right)} \right], \text{ where } q_{T-t}(\boldsymbol{x}|\tilde{\boldsymbol{x}}_{k\eta}) \propto \exp\left( -f_*(\boldsymbol{x}) - \frac{\left\| \tilde{\boldsymbol{x}}_{k\eta} - e^{-(T-k\eta)}\boldsymbol{x} \right\|^2}{2\left(1 - e^{-2(T-k\eta)}\right)} \right).$$

5: $\tilde{\boldsymbol{x}}_{(k+1)\eta} = e^{\eta}\tilde{\boldsymbol{x}}_{k\eta} + (e^{\eta} - 1)\,\boldsymbol{v}_k + \xi$ where $\xi$ is sampled from $\mathcal{N}\left(0, \left(e^{2\eta} - 1\right)\boldsymbol{I}_d\right)$.
6: **end for**
7: **Return:** $\tilde{\boldsymbol{x}}_{\lfloor T/\eta \rfloor \eta}$.