AIGC

扩散模型

作者：Sisyphes
组织：家里蹲
时间：March 9, 2024
版本：0.1

ElegantLaTeX Program

目录

第一章 零

1.1 符号说明

- 采样步数 s
- 噪声 z
- 参数 θ
- 原始样本 x_0
- 扰动核：$p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) := \mathcal{N}\left(\tilde{\mathbf{x}}; \mathbf{x}, \sigma^2 \mathbf{I}\right)$
- 扰动数据分布：$p_\sigma(\tilde{\mathbf{x}}) := \int p_{\text{data}}(\mathbf{x}) p_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \mathbf{dx}$
- 扰动数据的分布分数：$\nabla_{\mathbf{x}} \log p_{\alpha_i}(\mathbf{x})$
- 噪声条件网络优化目标：$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \sigma_i^2 \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{p_{\sigma_i}(\tilde{\mathbf{x}}|\mathbf{x})} \left[\left\| \mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, \sigma_i) - \nabla_{\tilde{\mathbf{x}}} \log p_{\sigma_i}(\tilde{\mathbf{x}} \mid \mathbf{x}) \right\|_2^2 \right]$
- 原始数据分布: p_x，先验分布 p_T
- 连续采样步数：$t \in [0, T]$
- 连续扩散过程的变量：$x(t)_{t=0}^{T}$
- 离散扩散过程变量：
- Itô SDE：$\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)\mathbf{d}t + g(t)\mathbf{dw}$
- 前向 SDE：$\mathbf{dx} = \mathbf{f}(\mathbf{x}, t)\mathbf{d}t + g(t)\mathbf{dw}$
- $x(t)$ 的 drift coefficient：$\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$
- $x(t)$ 的 diffusion coefficient：$g(\cdot) : \mathbb{R} \to \mathbb{R}$
- 反向 SDE：$\mathbf{d}x = \left[f(x, t) - g^2(t) \nabla_x \log p_t(x) \right] \mathbf{d}t + g(t)\mathbf{d}\bar{w}$，其中 $\bar{w}$ 是标准维纳过程，时间从 T 到 0，dt 也是负向。
- standard Wiener proces(Brownian motion)：$\boldsymbol{w}$
- $\mathbf{x}(s)$ 到 $\mathbf{x}(t)$ 的转移核：$p_{st}(\mathbf{x}(t) \mid \mathbf{x}(s)), 0 \leqslant s < t \leqslant T$
- probability flow ODE
- ELBO

1.2 一些废话

不重复生产，因此部分使用了两个博客的原文部分内容，和一些研究者的见解，在统一符号上做了一些调整。

1.2.1 无中生有

。

1.2.2 翻译

将问题翻译为计算机数学语言，将计算机数学语言翻译为数学语言，将代数翻译为几何，将模糊翻译为半模糊，将清晰翻译为模糊，将模糊翻译为模糊，将模糊翻译为清晰，将公式翻译为比喻，将公式翻译为代码，将代码翻译为业务代码，将论文翻译为项目，将项目翻译为产品部分...

第二章 总体介绍

- 模板官网：elegantlatex.org
- 模板 GitHub 地址：ElegantLaTeX
- 项目地址：Sisyphes
- 作图工具：geogebra

2.1 基本介绍

2.2 学术进展

1. 什么是扩散模型
2. 什么是分数生成模型
3. NCSN Yang, Ermon, 2019
4. DDPM; Ho et al. 2020
5. stochastic gradient Langevin dynamics Welling , Teh 2011
6. denoising/slice score matching 2011
7. Deep Unsupervised Learning using Nonequilibrium Thermodynamics 2015
8. Improved Denoising Diffusion Probabilistic Models 2021
9. High-Resolution Image Synthesis with Latent Diffusion Models 2022
10. Classifier-Free Diffusion Guidance 2021
11. Reparameterization Trick
12. 漫谈重参数：从正态分布到 Gumbel Softmax
13. SD 综述和应用
14. AIT-SD
15. TRT-SD
16. Oneflow-SD
17. PEFT

第三章 数学基础

3.1 高斯分布、熵，KL 散度、重参数

Evidence lower bound 定义：设 X, Z 是随机变量，其联合分布为 p_θ. 比如 $p_\theta(X)$ 是关于 X 的边缘分布，$p_\theta(Z|X)$ 是给定 X 关于 Z 的条件分布。于是有，对于样本 $x \sim p_\theta$, 对任意分布 q_ϕ，其 ELBO 定义为：

$$L(\phi, \theta; x) := \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z \mid x)}\right] \tag{3.1}$$

等价形式为：

$$\begin{aligned} L(\phi, \theta; x) &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln p_\theta(x,z)\right] + H\left[q_\phi(z \mid x)\right] \\ &= \ln p_\theta(x) - D_{KL}\left(q_\phi(z \mid x) \| p_\theta(z \mid x)\right). \end{aligned} \tag{3.2}$$

其中 H 表示熵，$\ln p_\theta$ 为 x 的证据，而 KL 散度恒大于 0，因此 $\ln p_\theta \geq \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$

deeplearningbook 2016 或者 deeplearningbook 2016 autoencoders 从中了解到，VLB 和 ELBO 在定义上存在一个正负符号只差。dlbook notation 2016

https://en.wikipedia.org/wiki/Entropy_(information_theory)Entropy information theory

$$\mathbf{H}(X) := -\sum_{x\in\mathcal{X}} p(x)\log p(x) \tag{3.3}$$

shannonsentropy。

两个一元高斯分布的 KL 散度公式：

$$KL\left(\mathcal{N}\left(\mu_1, \sigma_1^2\right) || \mathcal{N}\left(\mu_2, \sigma_2^2\right)\right) = \log\frac{\sigma_2}{\sigma_1} - \frac{1}{2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} \tag{3.4}$$

多元的参考:kl-divergence-between-2-gaussian-distributions

重参数技巧：reparameterization-trick

3.2 贝叶斯定理

关于先验，后验，极大似然，似然函数等

3.3 Fisher score 、Stein score

Fisher score 费希尔信息

Stein score A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. 1972 A Kernel Test of Goodness of Fit 2016 A Kernelized Stein Discrepancy for Goodness-of-fit Tests 2016 定义 2.1 给出了概率分布 p 的 Stein score function。该定义从 Stein 1972 年写的一篇文章中抽取出来，相关内容可参考Stein's method。

3.4 Markov chain Monte Carlo

马尔可夫链蒙特卡罗算法（MCMC） The Monte-Carlo Method Markov chain Monte Carlo

3.5 证据下界或变分下界

对于给定前向过程，$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{1-\beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$ $q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)=\prod_{t=1}^T q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$，逆向过程用 p_θ 表示，$p_\theta\left(\mathbf{x}_{0: T}\right)=p\left(\mathbf{x}_T\right) \prod_{t=1}^T p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right)$，数据的负对数似然为 $-log_{p_\theta}(x_0)$，x_0 属于训练数据空间中的元素，一种常见的变分下界推导方式是给负对数似然增加一个恒大于 0 的变量，关于在给定数据的情况下，过程数据的条件联合分布函数从正向和反向两个角度得到，他们的 KL 散度值。加上这个项，整个就能训练了，因此可叫训练项目。

$$
\begin{aligned}
-\log p_\theta\left(\mathbf{x}_0\right) & \leq-\log p_\theta\left(\mathbf{x}_0\right)+D_{\mathbf{KL}}\left(q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)\right) \\
& =-\log p_\theta\left(\mathbf{x}_0\right)+\mathbb{E}_{\mathbf{x}_{1: T} \sim q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}\left[\log \frac{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0: T}\right) / p_\theta\left(\mathbf{x}_0\right)}\right] \\
& =-\log p_\theta\left(\mathbf{x}_0\right)+\mathbb{E}_q\left[\log \frac{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0: T}\right)}+\log p_\theta\left(\mathbf{x}_0\right)\right] \\
& =\mathbb{E}_q\left[\log \frac{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0: T}\right)}\right] \\
\text{Let } L_{\mathbf{VLB}} & =\mathbb{E}_{q\left(\mathbf{x}_{0: T}\right)}\left[\log \frac{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0: T}\right)}\right] \geq-\mathbb{E}_{q\left(\mathbf{x}_0\right)} \log p_\theta\left(\mathbf{x}_0\right)
\end{aligned}
\tag{3.5}
$$

从交叉熵的角度，根据 Jensen's 不等式，也能得出变分下界：

$$
\begin{aligned}
L_{\mathbf{CE}} & =-\mathbb{E}_{q\left(\mathbf{x}_0\right)} \log p_\theta\left(\mathbf{x}_0\right) \\
& =-\mathbb{E}_{q\left(\mathbf{x}_0\right)} \log \left(\int p_\theta\left(\mathbf{x}_{0: T}\right) d \mathbf{x}_{1: T}\right) \\
& =-\mathbb{E}_{q\left(\mathbf{x}_0\right)} \log \left(\int q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right) \frac{p_\theta\left(\mathbf{x}_{0: T}\right)}{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)} d \mathbf{x}_{1: T}\right) \\
& =-\mathbb{E}_{q\left(\mathbf{x}_0\right)} \log \left(\mathbb{E}_{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)} \frac{p_\theta\left(\mathbf{x}_{0: T}\right)}{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}\right) \\
& \leq-\mathbb{E}_{q\left(\mathbf{x}_{0: T}\right)} \log \frac{p_\theta\left(\mathbf{x}_{0: T}\right)}{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)} \\
& =\mathbb{E}_{q\left(\mathbf{x}_{0: T}\right)}\left[\log \frac{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0: T}\right)}\right]=L_{\mathbf{VLB}}
\end{aligned}
\tag{3.6}
$$

通过一步一步的推导，多步变分下界可以写为多个 KL 散度和熵的和。这一过程，原因在于扩散扩散的一条马尔科夫链。整个过程是比较简单的，核心在等式第五行的一个贝叶斯转换，转换的原因在于给定扩散过程，逆向转移在给定 x_0 的基础上，能被具体的计算出来。

$$
\begin{aligned}
&\text{Attention}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right)=\text{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^{\top}}{\sqrt{d}}\right) \cdot \mathbf{V} \\
&\text{where } \mathbf{Q}=\mathbf{W}_Q^{(i)} \cdot \varphi_i\left(\mathbf{z}_i\right), \mathbf{K}=\mathbf{W}_K^{(i)} \cdot \tau_\theta(y), \mathbf{V}=\mathbf{W}_V^{(i)} \cdot \tau_\theta(y) \\
&\text{and } \mathbf{W}_Q^{(i)} \in \mathbb{R}^{d \times d_e^i}, \mathbf{W}_K^{(i)}, \mathbf{W}_V^{(i)} \in \mathbb{R}^{d \times d_r}, \varphi_i\left(\mathbf{z}_i\right) \in \mathbb{R}^{N \times d_c^i}, \tau_\theta(y) \in \mathbb{R}^{M \times d_r}
\end{aligned}
\tag{3.7}
$$

$$
\begin{aligned}
\mathbf{KL}\left(p_0(\mathbf{x}) \| p_\theta(\mathbf{x})\right) \leq \frac{T}{2} \mathbb{E}_{t \in \mathcal{U}(0, T)} \mathbb{E}_{p_t(\mathbf{x})}\left[\lambda(t)\left\|\nabla_{\mathbf{x}} \log p_t(\mathbf{x})-\mathbf{s}_\theta(\mathbf{x}, t)\right\|_2^2\right] \\
+\mathrm{KL}\left(p_T \| \pi\right)
\end{aligned}
\tag{3.8}
$$

3.6 SDE 和 ODE

简单介绍下。

概率流常微分的迹，随机微分方程的边缘概率密度，以及这两者的关系。从同样的数据分布采样，迹有同样的边缘分布？概率流常微分的半边结构。

第四章 扩散模型

4.1 整体认识

- Fisher divergence $\mathbb{E}_{p(\mathbf{x})}\left[\|\nabla_{\mathbf{x}}\log p(\mathbf{x})-\mathbf{s}_\theta(\mathbf{x})\|_2^2\right]=\int p(\mathbf{x})\,\|\nabla_{\mathbf{x}}\log p(\mathbf{x})-\mathbf{s}_\theta(\mathbf{x})\|_2^2\,\mathbf{dx}$
- score matching
- Langevin dynamics
- Correlation functions and computer simulations 1981
- Correlation functions and computer simulations 1994
- A Kernel Test of Goodness of Fit 2016
- A kernelized Stein discrepancy for goodness-of-fit tests 2016
- $\mathbf{x}_{i+1}\leftarrow\mathbf{x}_i+\epsilon\nabla_{\mathbf{x}}\log p(\mathbf{x})+\sqrt{2\epsilon}\mathbf{z}_i,\quad i=0,1,\cdots,K$, 从任意分布 $\mathbf{x}_0\sim\pi(\mathbf{x})$ 出发，其中 z_i 服从标准正太分布，$\epsilon\to 0, K\to\infty$, 分数模型的 MCMC 采样 (郎之万动力)，收敛到 $p(\mathbf{x})$。分数模型 $\mathbf{s}_\theta(\mathbf{x})\approx\nabla_{\mathbf{x}}\log p(\mathbf{x})$ 是通过 score-matching(分数匹配算法) 得到，不需要难处理的常数化因子，核心是优化 Fisher divergence，score-matching 规避了对 Stein score function 的具体计算。通过 score matching 去训练一个 score-based model，然后使用 Langevin dynamics 采样得到服从目标分布的数据。但是这种理论上的逻辑，在实际操作中，会有分数模型在低密度区精度很差 (从积分公式上看到，率密度权重因子)，另外分数模型是对高维张量的求导，因此在实际处理中需要做计算量的简化。解决方案是用噪声扰动数据点，并在噪声数据点上训练基于分数的模型。当噪声强度足够大的时候，噪声可以填充低数据密度区域，以提高估计分数的准确性。论文参考。$p_{\sigma_i}(\mathbf{x})=\int p(\mathbf{y})\mathcal{N}\left(\mathbf{x};\mathbf{y},\sigma_i^2 I\right)\mathbf{dy}$，从 $p(\mathbf{x})$ 中采样，然后使用重参数得到 $p_{\sigma_i}(\mathbf{x})$ 中的样本，其中 σ_i 是噪声方差，且递增。
- Generative Modeling by Estimating Gradients of the Data Distribution 2019
- Sliced score matching: A scalable approach to density and score estimation 2020
- Improved Techniques for Training Score-Based Generative Models 2020
- Score-Based Generative Modeling through Stochastic Differential Equations 2021 训练目标变为:$\mathbf{s}_\theta(\mathbf{x},i)\approx\nabla_{\mathbf{x}}\log p_{\sigma_i}(\mathbf{x})$ for all $i=1,2,\cdots,L$, $\sum_{i=1}^{L}\lambda(i)\mathbb{E}_{p_{\sigma_i}(\mathbf{x})}\left[\left\|\nabla_{\mathbf{x}}\log p_{\sigma_i}(\mathbf{x})-\mathbf{s}_\theta(\mathbf{x},i)\right\|_2^2\right]$，噪声 $\mathcal{N}\left(0,\sigma_i^2 I\right), i=1,2,\cdots,L$ 上面式子使用 score-matching 来训练。log-likelihood(对数似然) 的计算，这里先验是 $p_T(\mathbf{x})$，利用 n.c, s.c 中的定理 1 和等式 4，可以利用 ODE 的数值求解，在先验概率密度 p_T 已知的情况下，计算未知数据 $\mathbf{x}$ 的概率密度 p_0，得到数据的对数似然。这里和贝叶斯中的似然函数，有区别。在上面的设定中，似然函数是 $p(\mathbf{x}|\epsilon)$，其中 ϵ 是标准正太分布的噪声，而对数似然是根据数据得到的带参数 θ 的关于数据的概率密度函数，在某些情况下，你可以把这里的 ϵ 看做参数 θ，但这里还是有区别的。
- Maximum Likelihood Training of Score-Based Diffusion Models 2021
- Reverse-time diffusion equation models 1982
- A connection between score matching and denoising autoencoders 2011
- Neural Ordinary Differential Equations 2018
- Scalable Reversible Generative Models with Free-form Continuous Dynamics 2019
- variational dequantization to obtain likelihoods on discrete images
- 求关于不好计算的东西的的其他变量的导数，将不好计算的量消去，变为从导数的角度来刻画该问题。
- 生成模型的逆向问题，或者贝叶斯推理问题：设随机变量 $\mathbf{x},\mathbf{y}$，设计好前向 $\mathbf{x}\to\mathbf{y}$，于是其转移概率 $p(\mathbf{y}|\mathbf{x})$ 已知，逆向问题是计算 $p(\mathbf{x}|\mathbf{y})$。根据贝叶斯法则：$p(\mathbf{x}\mid\mathbf{y})=p(\mathbf{x})p(\mathbf{y}\mid\mathbf{x})/\int p(\mathbf{x})p(\mathbf{y}\mid\mathbf{x})\mathbf{dx}$。设 $\mathbf{x}$ 为先验，则后验若按照贝叶斯法则，但积分项难于计算，若两边对 $\mathbf{x}$ 求导，即 $\nabla_{\mathbf{x}}\log p(\mathbf{x}\mid\mathbf{y})=\nabla_{\mathbf{x}}\log p(\mathbf{x})+\nabla_{\mathbf{x}}\log p(\mathbf{y}\mid\mathbf{x})$，容易看到利用 score matching 和 Langevin-type 采样能直接得到后验分数函数值。
- 公式翻译，理论和应用间的近似和区别，模糊地方的经验法则，代码上的并行考虑，部署上的性能考虑，应

用上的扩展性，资源的考虑。

- 分数模型是从分数匹配和郎之万动力采样中得到的，而扩散模型是有证据下界 ELBO 和解码模型采样得到的。而 ELBO 训练的扩散模型和分数模型是等价的。这个是在 2020 年提出 DDPM 之后，2021 年宋飏等证明的，并提出了 SDE 的统一架构。这种不同角度对同一任务的描述，类似量子力学中的波函数角度和矩阵力学角度最终被证明是等价的类似，来个更简单点的，单位圆盘上的有理数点的无穷性和本原勾股数的无穷性是等价的。
- 基于分数模型的成功关键：(1) 使用多个噪声尺度的扰动数据，并为每个噪声尺度训练基于分数的模型；(2) 使用 U-Net 架构 (我们使用 RefineNet，因为它是现代版本的基于分数的模型；(3) 将朗之万 MCMC 应用于每个噪声尺度，并将它们链接在一起。
- 基于分数的生成模型，只能用在连续数据上，若离散数据，可先利用 VAE 将其映射到连续的隐空间中，然后训练分数模型，最后解码。对于分数模型采样速度慢的问题，可以利用 ODE 的快速求解，来克服。
- 基于分数的生成模型或扩散模型：数据生成，密度估计，逆问题求解。
- ELBO 或 VLB(两者是同种东西的不同说法)，分别叫证据下界，变分下界。变分下界来自 VAE 模型的说法，而且扩散模型本身，也可以看成多步 VAE 模型。但实际上，这里面存在一些正负符号的区别，有误用嫌疑。
- LDM 2022
- Elucidating the Design Space of Diffusion-Based Generative Models 对当前的几种 diffusion model 做了一个统一的框架。解决了 diffusion ODE 的生成效果一直都不如 diffusion SDE，效果反而更好。但，实际上是个大杂烩 (这点我和 Baof 观点相同)。
- Analytic-DPM “DDPM 的均值实际上是对 diffusion SDE 的 maximum likelihood SDE solver，并且最优方差有解析形式，且可以被 score function 唯一确定。最优方差也可以用 score model 来近似。”
- dpm-solver，DPM-SOLVER++ “证明了 DDIM 是 diffusion ODE 的一阶 ODE solver，并且提出了二阶、三阶 solver，可以做到 10 步采样质量很不错，20 步几乎收敛。不需要任何额外训练，任给一个 pretrained model 都可以直接用。DDIM 对应了 diffusion ODE 的 1 阶 ODE solver，它的加速效果好是因为它考虑了 ODE 的半线性结构，而 DPM-Solver 给出了对应的更高阶的 solver，可以让 10 步左右的采样达到与 DDPM 的 1000 步的采样相当。”
- Maximum Likelihood Training for Score-Based Diffusion ODEs by High-Order Denoising Score Matching “从理论角度彻底分析出了 Diffusion SDE 与 Diffusion ODE 的最大似然训练的联系和区别，并且提出了新的二阶、三阶 denoising score matching 算法来解决了 diffusion ODE 的最大似然训练问题。结论就是，之前训练 diffusion model 的“去噪”方法其实是一阶 score matching，它只适合训练 diffusion SDE 的最大似然估计，但并不适合训练 diffusion ODE 的最大似然估计。想要对 diffusion ODE 做最大似然估计，需要采用上面提出的方法。”
- “深度生成模型除了可以生成数据以外，还有一类核心任务是估计数据的概率密度（可以用模型计算的数据似然（likelihood）来刻画）。GAN 被诟病的一点就是无法计算似然，因为它是隐式生成模型（implicit generative model），这导致 GAN 无法被用来数据压缩等领域。而 VAE 只能计算数据似然的一个下界，也不太令人满意。”
- “DDPM 与上文提到的连续版本的 reverse SDE 可以称为 diffusion SDE（也叫 Score-based SDE），它们都是基于 SDE（离散版本为条件高斯分布）定义的生成模型。然而，这类模型与 VAE 一样，无法计算精确的数据似然（likelihood），而只能计算 ELBO。Diffusion model 的另一个优势是，只要训练了 score model（上文提到的三种等价形式都可以转换为 score model），那么就可以导出一个 Neural ODE，它是一类 continuous normalizing flow，可以精确计算数据的似然。”
- ”然后说说我认为 diffusion model 的美妙之处。它把困难的数据生成，拆分成了若干个相对简单的任务，每个任务即对应一次去噪。这种拆分有坏处（即推断慢，但我们的工作一定程度上解决了这个问题），也有好处（更加可控的生成）。”
- 一个东西，有一些框架，发展到一定时候，会出现新的框架。
- PPT

- 问题，NCSN 是如何训练的？和 DDPM 训练有啥区别？
- DDIM 是使用的 NCSN 或变分去噪的训练方式，但论文公式推导是使用的 V_LB 思路
- Analysis-DPM 是在 DDIM 的基础上分析了其其实可以计算出中间的最优均值和方差
- DPM-solver 是将 SDE 对应推出来的 ODE 的常微分方程，做了具体的计算，加速了推理采样速度，但是中间公式的推导却使用的是 DDPM 原始的表示，而非分数模型，虽然做了两者的等价变换。
- DPM-solver++ 是在 DPM-solver 的基础上，对 guide 做了分析，但是在公式推导过程中，做了 DDPM 模型的 x_0 替换，使得最终表示出来的微分方程，在高阶部分的展开不同。不过这个替换的核心原因，似乎是来自 Photorealistic text-to-image diffusion models with deep language understanding 2022b 这篇文章的启发，该文章实验出，对输入变量的尺度变换，能有效抑制数据的边界范围溢出的情况，而且采样生成的数据效果确实更好了。从直观上，论文也做了两点分析，主要是 s 因子，带来的尺度上的变化。
- 我比较好奇的是，使用的是 DDPM 的建模思路，但具体公式推导上采用的是 SDE 的模型表示，或者反之，做一些替换，前者建模是从贝叶斯，极大似然，的思路，做变分界的推导，后者是从分数模型对应的 SDE 的逆向推导，虽然两者得到的模型在效果上等价的，刻画了原始数据的分布，但是，彼此互换，就能解决不同的问题，有点迷。
- 分数模型 SGM 的训练是预测噪声除以方差，同 DDPM 差不多，不过 SGM 设置添加的噪声服从高斯分布，但没假设其满足马尔科夫性。通过 sliced score matching 训练 NCSN，即去噪分数匹配。推理使用 nnealed Langevin dynamics。
- 分数模型的建模思路和 DDPM 的建模思路不同，前者通过分数匹配优化，郎之万动力采样，后者通过贝叶斯，极大似然的变分下界优化，” 逆向” 采样。但二者的连续版本被 SDE 统一。而 SDE 有对应的 ODE，其解能保持 SDE 的所有边缘分布，从而对 OD E 的求解，能极大提高推理的速度。
- 在连续化 DDPM 和 SGM 的时候，就去掉了 DDPM 的马尔科夫性的假设，虽然 DDIM 也是去掉了 DDPM 的前向的马尔科夫性假设，但同时假设了与 DDPM 设定前向为马尔科夫性之后，推导出的逆向中条件后验分布为高斯分布同等形式的高斯分布。DDIM 是通过什么，带来的推理加速 (逆向采样几乎一样，多了一个 σ_t^2，在附录 C.1 中说明了是通过 DDPM 的逆采样的子序列采样得到)？但是奇怪的是，DDIM 的训练，竟然是变分去噪或者分数模型。其实差不多，这点需要代码上确认下区别。在实际中，DDIM 的效果并不怎么样，后续的 SDE 均比它好很多。
- It is well-known that maximizing the log-likelihood of a probabilistic model is equivalent to m inimizing the KL divergence from the data distribution to the model distribution.
- 在特殊的权重因子上，极小化 SDE 的分布与数据分布的 KL 散度的上界，其上界是和分数匹配相关的损失，从而达到优化分数模型来提高数据的似然函数值的精度。并且，优化组合分数匹配损失，并不能直接得到数据的对数似然函数，通过概率流模型 continuous normalizing flows (CNFs) 优化 SGM 能得到 SGM 的极大似然值，并且提高其似然函数值的准确性。ScoreFlow。
- SDE 的模型是不能计算似然值的，但是 ODE 是可以精确就算的。优化一阶分数损失并不能得到 ODE 的似然，负对数似然被 ODE 的一阶，二阶，三阶分数误差所控制。因此一种高阶的去噪分数匹配算法能达到更精确的似然估计。具体来说和 SDE 类似的计算了模型和数据的散度，并在 ODE 这块，得到了类似的优化界。SDE 被 J_{SM} 控制，ODE 被 J_{SM}, J_{Fisher} 两者控制。这里有个具体的方格例子，可以测试下。ScoreODEs。问题，似然函数的具体计算，意味着啥？这里也是隐含计算的，并且只针对连续值，实际的值需要做量化处理，量化处理靠谱吗？经过仔细分析了吗？可以分析还是类似于以前分类网络的感知损失这种参考评估？
- sliced score matching：分数模型的估计在参数估计不一致，估计方差较大，实现繁琐，三方面的改进。提出了投影分数函数到随机向量然后做比较的方法。从而能处理更高维的数据，更复杂的分布。核心思路是拟合高维函数在不同方向上的投影。文章也做了理论分析，在一些正则条件下，能做到一致的和渐近正态的参数估计，是一个良定义的统计估计标准。(能量) 密度估计和分数估计。整个文章可以理解为：对 2005 年文章 Hyvärinen (2005)) 中，梯度迹的估计。但是仍然要前向 n 次。理论上是这样，但是实际上 sliced score matching 的损失还是非常不稳定的，若增加了噪声，就稳定多了。
- 极大似然对 unnormalized models 的优化，存在难以处理的归一化分母 Z_θ，对其做关于 x 的微分，则消除了该

问题，于是问题转化为拟合被称为分数函数的问题，但同时也带来了计算复杂的问题，黑塞矩阵 (Hessian)，论文思路启发自 Sliced Wasserstein distance (Rabin et al., 2012),

- Denoising Score Matching(2011) 彻底克服了需要计算 Hessian 的问题。但是论文只是考虑了一次加噪声，后续论文 NCSN 是做了多次加噪。相关对分数匹配在高维上的优化，还有 Approximate Backpropagation，Curvature Propagation 等不同类型文章。Denoising Score Matching 是估计扰动后的数据的分数。sliced score matching 是对未扰动的数据的分数估计，但是计算量大了几倍。
- NCSN, 郎之万动力采样：从分数模型中采样得到符合原始数据分布的数据。数据本身是嵌入在高维流形中的低维流形，而分数函数是在原始空间高维流形中计算梯度的，当数据被限制在低维流形中时，该微分是无定义的，其次分数匹配的优化目标能优化出一个一致性分数估计的前提是，数据分布的支撑集得是整个空间，当数据集中在低维流形时，该估计是不一致的。低密度区域的数据，不仅导致分数的估计偏差，而且使得郎之万动力采样结果不准确。因此添加噪声，不仅能保证数据集的支撑是整个空间，而且更大的噪声，能减少低密度区域的出现。使得分数估计定义良好，且估计准确，采样准确。通过添加多尺度的噪声，使得噪声扰动分布逼近到真实数据分布。这将通过，训练一个拟合所有尺度噪声的模型。采样从大尺度噪声逐渐过渡到小尺度噪声。添加也是从大噪声到小噪声，直到数据变为高斯分布。

4.1.1 问题

- 马尔科夫过程对任意分布都能变换的定理
- 分数模型的训练
- SMLD,DDPM 的 SDE 化细节
- Langevin MCMC
- 前向 SDE 的逆向由什么定理得到
- 条件逆过程
- progressive distillation
- classifier-free guidance(CFG)

4.2 离散版 DDPM 等

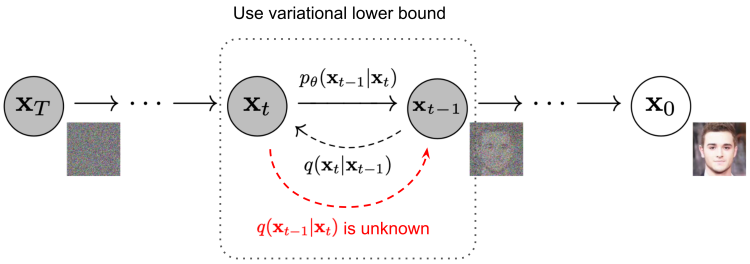

图 4.1: DDPM

真实数据分布 $\mathbf{x}_0 \sim q(\mathbf{x})$，这里的 x_0 可能是 $512 * 512 * 3$ 大小的图像集合，也可能是 $\mathbb{R}^n$ 中的向量，比如动画 BVH 解析出的数据表示。

前向扩散过程定义为 T 步的马尔科夫过程，转移概率为 $q(\mathbf{x}_t|\mathbf{x}_{t-1}), t \in \mathbb{N}$。对于给定数据 $\mathbf{x}_0$，向其添加同维度的高斯样本 (微小扰动)，然后递推。设置好每一步添加样本的均值和方差，使得输入样本 x_0 变为标准正太分布的样本，如图 4.1 所示。比如序列 $\{\beta_t \in (0,1)\}_{t=1}^T, \beta_1 < \beta_2 < \cdots < \beta_T\}$，将转移概率 (条件概率) 定义为 $q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{1-\beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$ 容易看到，当 $T \rightarrow \infty$ 时，$q(\mathbf{x}_t|\mathbf{x}_{t-1})$ 趋近于标准状态分布。

令

$$\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i \tag{4.1}$$

对于 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$，由重参数化得 $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$，又由高斯分布的叠加特性，可得 $\mathbf{x_0} \to \mathbf{x_t}$ 的转化：

$$\begin{aligned}\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} && ;\epsilon_{t-1}, \epsilon_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \\ &= \cdots \\ &= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\end{aligned} \tag{4.2}$$

其中合并的高斯分布 $\bar{\epsilon}_{t-2} = \sqrt{\alpha_t(1-\alpha_{t-1})}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1}$。于是得到

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{4.3}$$

根据变换关系，有 $\bar{\alpha_1} > \cdots > \bar{\alpha_T}$。

前向过程：人为设计一个马尔科夫扩散过程，将任意分布 (有限维度)，在有限步数之内，逼近一个标准正太分布。

关于 DL 的建模，可以简单理解为将目标转化特定的语言体系，然后在该体系中构建一个闭环，而且需要梳理一下闭环中的链接关系，并在优化中把这些关系体现出来。

这里的任务是生成训练数据对应空间的样本，由贝叶斯定理，只要知道 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), t = 1, \cdots, T$ 整个链条闭环就有了。首先数据的分布 $q(\mathbf{x}_0)$ 是未知的，且数据分布是我们的目标，但我们会准备很多数据，所以可以假想 $q(\mathbf{x}_0)$ 已知，又前向是已知的，于是，整个链条中，只剩下反向 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$，可用参数模型 $\mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right)$ 来表示。

前向和反向的闭环，前向：

$$q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T) = q(\mathbf{x}_{0:T}) := q(\mathbf{x}_0)\prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \tag{4.4}$$

反向：

$$p_\theta(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T) = p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \tag{4.5}$$

这两者相等，因为它们都是表示初始数据和中间数据的联合分布。要优化这两分布，就要找一个刻画两分布的距离的函数。其中 KL 散度，交叉熵，Stein's score 函数均可。对于 KL 散度，有

$$\begin{aligned}&\mathbf{KL}(q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T) \| p_\theta(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)) \\ \overset{(i)}{=}\ & -\mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)}\left[\log p_\theta(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)\right] + \mathbf{const} \\ \overset{(ii)}{=}\ & \underbrace{\mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)}\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^{T}\log\frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})}\right]}_{:=-L_{\mathbf{VLB}}(\mathbf{x}_0)} + \mathbf{const} \\ \overset{(iii)}{\geq}\ & \mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] + \text{const},\end{aligned} \tag{4.6}$$

这里需要说明一下，很多地方说变分下界 VLB 和证据下界是一样的，包括 Wikipedia，但实际上 Wikipedia 关于 ELBO 的定义和一些扩散论文对 VLB 的定义能看出来，两者差一个正负符号 (这里面应该存在一些误用)。

从式 (4.6) 的推导中看到

$$\log\frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \tag{4.7}$$

是关键。这里前向 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ 均值、方差已知，而逆向我们假设也满足高斯分布，利用网络 $\mathbf{s}_\theta(\mathbf{x}_t, t)$ 来表示它。当前情况可以做优化，但是实际没有选择这个，而是考虑到前向的马尔可夫性，有 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, x_0)$，然后利用贝叶斯法则，将三元组 $(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_{t-1})$ 做个逆向变形，得到 $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$ 的具体计算表达式。然后相关

的式子都有表达式，于是可以对 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, x_0) = q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ 做等量替换。细节如下：

$$\begin{aligned} q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) &= q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0\right) \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)} \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\left(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1}\right)^2}{\beta_t} + \frac{\left(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\right)^2}{1-\bar{\alpha}_{t-1}} - \frac{\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0\right)^2}{1-\bar{\alpha}_t}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\frac{\mathbf{x}_t^2 - 2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1} + \alpha_t\mathbf{x}_{t-1}^2}{\beta_t} + \frac{\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1} + \bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}} - \frac{\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0\right)^2}{1-\bar{\alpha}_t}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)\mathbf{x}_{t-1} + C\left(\mathbf{x}_t, \mathbf{x}_0\right)\right)\right) \end{aligned} \tag{4.8}$$

其中 $C\left(\mathbf{x}_t, \mathbf{x}_0\right)$ 是和 $\mathbf{x}_{t-1}$ 无关的量，$\propto$ 表示前面的比例项去掉了，从上面看到，对于逆向的某一步，增加初始数据 $\mathbf{x}_0$ 的条件下，仍然是一个高斯分布，且能计算。也即

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right) \tag{4.9}$$

其中 $\tilde{\beta}_t = 1/\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right) = 1/\left(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}\right) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right) &= \left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right) / \left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right) \\ &= \left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right) \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t \\ &= \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 \end{aligned} \tag{4.10}$$

将前向过程 $\mathbf{x}_0$ 和 $\mathbf{x}_t$ 的关系代入，得到

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t\right) \\ &= \frac{1}{\sqrt{\alpha_t}}\left(\mathrm{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) \end{aligned} \tag{4.11}$$

而这个时候，只需要将优化目标 (4.6)*ii* 式中的 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ 用式 (4.8) 中的 $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0)$ 替换一下即得到最终’ 可操作的’ 优化目标。

具体如下：

$$
\begin{aligned}
L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0:T}\right)}\right] \\
&= \mathbb{E}_q\left[\log \frac{\prod_{t=1}^T q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_T\right) \prod_{t=1}^T p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right)+\sum_{t=1}^T \log \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right)+\sum_{t=2}^T \log \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}+\log \frac{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right)+\sum_{t=2}^T \log \left(\frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)} \cdot \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0\right)}\right)+\log \frac{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right)+\sum_{t=2}^T \log \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}+\sum_{t=2}^T \log \frac{q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0\right)}+\log \frac{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right)+\sum_{t=2}^T \log \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}+\log \frac{q\left(\mathbf{x}_T \mid \mathbf{x}_0\right)}{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}+\log \frac{q\left(\mathbf{x}_1 \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}\right] \\
&= \mathbb{E}_q\left[\log \frac{q\left(\mathbf{x}_T \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_T\right)}+\sum_{t=2}^T \log \frac{q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}-\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)\right] \\
&= \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_T\right)\right)}_{L_T}+\sum_{t=2}^T \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \underbrace{-\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}_{L_0}]
\end{aligned}
\tag{4.12}
$$

看起来比较长，其实非常简单，核心就是第 4 个等式到第 5 个等式的变换，变换是根据建模的基本假设来做的，容易想到。上面的 KL 散度是关于两个高斯分布的散度，且取了对数，容易得到，损失函数是关于高斯分布均值的平方差的式子 (令方差为常数)。从 DL 的角度来说，只需让模型预测每一步的均值即可，也即

$$
\mathbf{x}_{t-1}=\mathcal{N}\left(\mathbf{x}_{t-1} ; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{4.13}
$$

其中 $\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)$ 表示神经网络。经过简化 (将不影响优化的系数等删去等)，最终得到

$$
\begin{aligned}
L_t^{\text{simple}} &= \mathbb{E}_{t \sim[1, T], \mathbf{x}_0, \boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right] \\
&= \mathbb{E}_{t \sim[1, T], \mathbf{x}_0, \boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}_t, t\right)\right\|^2\right]
\end{aligned}
\tag{4.14}
$$

从中可以看到，只需要预测所有前向步骤所添加的噪声。最终得到训练和推理算法如下：

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
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** $t=T, \dots, 1$ **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** $\mathbf{x}_0$

图 4.2: DDPM algo

4.3 NCSN

数据样本 $\{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^N$ 满足独立同分布，数据的分布用 $p_{data}(\mathbf{x})$ 表示，概率密度 $p(\mathbf{x})$ 的分数定义为 $\nabla_{\mathbf{x}} \log p(\mathbf{x})$，分数网络 score network $s_\theta : \mathbb{R}^D \to \mathbb{R}^D$，其中 θ 表示网络参数。用分数网络来刻画数据分布，因为若概率密度的分数已知，则可以利用郎之万动力采样，其采样出的数据符合数据原始分布。因此基于分数的生成模型的核心，就是分数匹配和郎之万动力采样。

能量模型定义为：

$$p_m(\mathbf{x};\boldsymbol{\theta}) = \frac{\tilde{p}_m(\mathbf{x};\boldsymbol{\theta})}{Z_{\boldsymbol{\theta}}}, \quad Z_{\boldsymbol{\theta}} = \int \tilde{p}_m(\mathbf{x};\boldsymbol{\theta})\mathrm{d}\mathbf{x} \tag{4.15}$$

其分数则为：

$$\mathbf{s}_m(\mathbf{x};\boldsymbol{\theta}) \triangleq \nabla_{\mathbf{x}} \log p_m(\mathbf{x};\boldsymbol{\theta}) = \nabla_{\mathbf{x}} \log \tilde{p}_m(\mathbf{x};\boldsymbol{\theta}), \quad \mathbf{s}_d(\mathbf{x};\boldsymbol{\theta}) \triangleq \nabla_{\mathbf{x}} \log p_d(\mathbf{x};\boldsymbol{\theta}) \tag{4.16}$$

消去了 $Z_{\boldsymbol{\theta}}$，其中 p_d 是 p_{data} 的简写。

上述分数匹配于 2005(Hyvärinen, 2005) 年被提出，主要用来解决基于能量的概率密度估计 (极大似然建模) 分母归一化因子难处理的问题，原始优化目标为：

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|_2^2\right] \tag{4.17}$$

该论文得出其等价形式：

$$J(\boldsymbol{\theta}) \triangleq \mathbb{E}_{p_d}\left[\operatorname{tr}\left(\nabla_{\mathbf{x}}\mathbf{s}_m(\mathbf{x};\boldsymbol{\theta})\right) + \frac{1}{2}\|\mathbf{s}_m(\mathbf{x};\boldsymbol{\theta})\|_2^2\right], \tag{4.18}$$

其中 tr 表示矩阵的迹，上述公式需要计算 log-density 的 Hessian 矩阵，计算量非常大。

在数学上，1981 年就有了相关定理。参考数学基础部分。

Sliced Score Matching 一文对 (Hyvärinen, 2005) 做了改进，得出

$$\mathbb{E}_{p_{\mathbf{v}}}\mathbb{E}_{p_{\text{data}}}\left[\mathbf{v}^\top \nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\mathbf{v} + \frac{1}{2}\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\|_2^2\right] \tag{4.19}$$

其中 $p_{\mathbf{v}}$ 是一个简单分布的随机向量，比如多元标准高斯分布。该结论表明，将原始高维矩阵关于随机向量做投影，去逼近式 (33) 的迹，能减少计算量的同时，保持无偏估计。

另外 2010 年的一篇文章 A Connection Between Score Matching and Denoising Autoencoders 提出的 Denoising score matching 能做到 0 梯度计算 (计算量更少了) 达到等价的优化效果，但是是对扰动后的数据的密度估计。具体来说，定义扰动后的数据分布为：

$$q_\sigma(\tilde{\mathbf{x}}) \triangleq \int q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) p_{\text{data}}(\mathbf{x})\mathrm{d}\mathbf{x} \tag{4.20}$$

于是优化目标

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\|_2^2\right] \tag{4.21}$$

的最优解 $\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x})$ 当扰动很小的时候，有：

$$\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}) = \nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) \tag{4.22}$$

朗之万动力学在已知分数函数 $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ 的情况下，可以采样出概率密度为 $p(\mathbf{x})$ 的样本。具体来说：

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2}\nabla_{\mathbf{x}} \log p(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon}\mathbf{z}_t \tag{4.23}$$

其中 ϵ 表示固定的步长大小，$\tilde{\mathbf{x}}_0 \sim \pi(\mathbf{x})$ 是一个先验分布，$z_t \sim \mathcal{N}(0, I)$。当 $\epsilon \to 0, T \to \infty$，在一些正则条件下，$\tilde{\mathbf{x}}_T \sim p(\mathbf{x})$。具体分析见：Bayesian Learning via Stochastic Gradient Langevin Dynamics 2011。

NCSN 是在以上基础上提出的，作者分析了基于分数的生成模型的两个问题。基于一些基础实验，1：Sliced score matching 在原始数据上的迭代损失曲线没添加噪声后的迭代平稳，而且前者是极度不平稳，上下波动很大，对此作者给出了基于流形假设的分析；2. 通过分数匹配的方法，去做一个混合高斯分布的分数估计，发现在低密度区域匹配的很不准确，同时混合模型的权重因子失效严重，在混合区域，使用郎之万动力采样，是极度缓慢才能收敛到实际情况，因为分数函数关于不同混合密度的支撑集上的函数值，是和混合因子无关的，导致优化

很难处理混合模型各自的连接区域的密度情况，以至于采样不准确。

基于这些基本观察，分析，作者提出了带尺度的扰动分数模型：Noise Conditional Score Networks。一来解决了高低密度不均匀难以优化，导致采样不准的问题，二来解决了数据在高维空间的低维嵌入的普遍现象导致的分数模型无法正确优化的问题。因为将数据不断扰动，能使其充满整个空间，同时，低密度区域得到了有效填充，另外不同尺度，从极小扰动到不断增幅扰动因子，使得一个分数模型在全空间中去拟合不同尺度的扰动，得到非常充分的训练，因此其逆向采样，能更好的近似到原始分布。

称 $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, \sigma)$ 为 Noise Conditional Score Network (NCSN) 是指：设扰动因子 $\{\sigma_i\}_{i=1}^{L}$ 满足 $\frac{\sigma_1}{\sigma_2} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$，定义扰动尺度为 σ 的数据分布

$$q_\sigma(\tilde{\mathbf{x}}) \triangleq \int p_{\text{data}}(\approx) \mathcal{N}(\tilde{\mathbf{x}} \mid \approx, \sigma^2 I) \tag{4.24}$$

训练一个

$$\forall \sigma \in \{\sigma_i\}_{i=1}^{L} : \mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, \sigma) \approx \nabla_{\mathbf{x}} \log q_\sigma(\tilde{\mathbf{x}}) \tag{4.25}$$

的网络，其中 $\mathbf{x} \in \mathbb{R}^D, \mathbf{s}_{\boldsymbol{\theta}} \in \mathbb{R}^D$。

因 $q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}\left(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 I\right)$，有 $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = -(\tilde{\mathbf{x}} - \mathbf{x})/\sigma^2$，于是对于给定的尺度 σ，其去噪分数匹配损失为：

$$\ell(\boldsymbol{\theta}; \sigma) \triangleq \frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathcal{N}(\mathbf{x}, \sigma^2 I)} \left[\left\| \mathrm{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, \sigma) + \frac{\tilde{\mathbf{x}} - \mathrm{x}}{\sigma^2} \right\|_2^2 \right] \tag{4.26}$$

对于所有尺度 $\forall \sigma \in \{\sigma_i\}_{i=1}^{L}$ 得到最终的优化目标：

$$\mathcal{L}\left(\boldsymbol{\theta}; \{\sigma_i\}_{i=1}^{L}\right) \triangleq \frac{1}{L} \sum_{i=1}^{L} \lambda(\sigma_i) \ell(\boldsymbol{\theta}; \sigma_i) \tag{4.27}$$

其中 $\lambda(\sigma_i) > 0$，为依赖于 σ_i 的系数函数。

该目标函数的最优解 $\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}, \sigma)$ 满足 $\forall i \in \{1, 2, \cdots, L\}, \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}, \sigma_i) = \nabla_{\mathbf{x}} \log q_{\sigma_i}(\mathbf{x})$。$\lambda(\cdot)$ 选择使得所有尺度因子的数量级一致，经验得知，一个训练好的分数模型，其范数正比于尺度的倒数，又 $\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma} \sim \mathcal{N}(0, I)$，因此，选择 $\lambda(\sigma) = \sigma^2$，使得 $\|\sigma \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, \sigma)\|_2 \propto 1$。

最后论文提出了 annealed Langevin dynamics 采样算法 (对比式 (38))：将原始步长 ϵ 改进为 $\epsilon \cdot \sigma_i^2 / \sigma_L^2$。

Algorithm 1 Annealed Langevin dynamics.

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

1: Initialize $\tilde{\mathbf{x}}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad \alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$ ▷ α_i is the step size.
4: $\quad$ **for** $t \leftarrow 1$ to T **do**
5: $\quad\quad$ Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
6: $\quad\quad \tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2} \mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$
7: $\quad$ **end for**
8: $\quad \tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
9: **end for**
return $\tilde{\mathbf{x}}_T$

图 4.3: NCSN annealed Langevin dynamics

4.3.1 score matching

4.3.2 极大似然的计算

Yang Song 继其 SGM 模型，给出了 SDE 解的边缘分布同数据的分布的 KL 散度与 SM 模型优化对象的存在一个不等关系，使得优化 SM 模型，能得到 SDE 对应的数据边缘分布，也就是原始数据的似然函数值。

具体来讲：

$$D_{\mathrm{KL}}\left(p \| p_{\theta}^{\mathrm{SDE}}\right) \leq \mathcal{J}_{\mathrm{SM}}\left(\boldsymbol{\theta} ; g(\cdot)^{2}\right)+D_{\mathrm{KL}}\left(p_{T} \| \pi\right) \tag{4.28}$$

论文的网络是使用概率流网络来训练的?。

在此基础上 Cheng Lu 等人，给出了扩散对应的 ODE 的似然函数的上界。

$$\mathcal{J}_{\mathrm{ODE}}(\theta) \leq \sqrt{\mathcal{J}_{\mathrm{SM}}(\theta)} \cdot \sqrt{\mathcal{J}_{\text{Fisher }}(\theta)} \tag{4.29}$$

论文证明了一阶分数模型并不能得到扩散对应的 ODE 的似然，该似然被一阶，二阶，三阶分数匹配误差所控制，以此提出了一个高阶的分数去噪模型，能得到更好的原始数据的似然逼近，并同时能保证生成高质量数据。

4.4 连续版 Score SDE

SDE 中的噪声尺度因子，和 NCSN 的写法就大小比较在指标上是相反的，原因在于使用 DDPM 的方式，重新定义了 NCSN 中的优化目标，得到

$$\boldsymbol{\theta}^{*}=\arg \min _{\boldsymbol{\theta}} \sum_{i=1}^{N}\left(1-\alpha_{i}\right) \mathbb{E}_{p_{\text{data }}(\mathbf{x})} \mathbb{E}_{p_{\alpha_{i}}(\tilde{\mathbf{x}} \mid \mathbf{x})}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, i)-\nabla_{\tilde{\mathbf{x}}} \log p_{\alpha_{i}}(\tilde{\mathbf{x}} \mid \mathbf{x})\right\|_{2}^{2}\right] . \tag{4.30}$$

和逆向马尔科夫链 (采样)

$$\mathbf{x}_{i-1}=\frac{1}{\sqrt{1-\beta_{i}}}\left(\mathbf{x}_{i}+\beta_{i} \mathbf{s}_{\boldsymbol{\theta}^{*}}\left(\mathbf{x}_{i}, i\right)\right)+\sqrt{\beta_{i}} \mathbf{z}_{i}, \quad i=N, N-1, \cdots, 1 \tag{4.31}$$

论文中将其称为 ancestral sampling。

问题是 (43) 式能通过 ELBO 推导出来？通过 ELBO 推导出来，优化的是均值，而转移概率为高斯，对数求导后，差不多也是均值。仔细检查下，应该就是通过 ELBO 得到 (43) 所示的分数表达形式。

两者的分数表达形式和采样形式差别不大，将其连续化，并做形式上的推广，得到 *Itô* SDE 方程：

$$\mathrm{d} \mathbf{x}=\mathbf{f}(\mathbf{x}, t) \mathrm{d} t+g(t) \mathrm{d} \mathbf{w} \tag{4.32}$$

而 Anderson (1982) 的论文表示，其逆向也是扩散过程，reverse-time SDE 为：

$$\mathrm{d} \mathbf{x}=\left[\mathbf{f}(\mathbf{x}, t)-g(t)^{2} \nabla_{\mathbf{x}} \log p_{t}(\mathbf{x})\right] \mathrm{d} t+g(t) \mathrm{d} \overline{\mathbf{w}} \tag{4.33}$$

其对应的优化目标可写为：

$$\boldsymbol{\theta}^{*}=\arg \min _{\boldsymbol{\theta}} \mathbb{E}_{t}\left\{\lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t) \mid \mathbf{x}(0)}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)-\nabla_{\mathbf{x}(t)} \log p_{0 t}(\mathbf{x}(t) \mid \mathbf{x}(0))\right\|_{2}^{2}\right]\right\} . \tag{4.34}$$

其中 $\lambda:[0, T] \rightarrow \mathbb{R}_{>0}, t \in[0, T], \mathbf{x}(0) \sim p_{0}(\mathbf{x}), \mathbf{x}_{t} \sim p_{0 t}(\mathbf{x}(t) \mid \mathbf{x}(0))$ 其解 $\mathbf{s}_{\boldsymbol{\theta}^{*}}(\mathbf{x}, t)=\nabla_{\mathbf{x}} \log p_{t}(\mathbf{x})$ 对几乎所有 $\mathbf{x}, t$ 成立。另外需要注意的是，当转移核是优化目标 (47) 的解的时候，若 $\mathbf{f}(\cdot, t)$ 是仿射的，则转移核是高斯分布。均值和方程能具体计算出来，参考 (Arno Solin 2019)Applied stochastic differential equations。

这里需要注意的是，论文 2.1 节指出，NCSN 尺度上的逆向描述被简写为 SMLD(DENOISING SCORE MATCHING WITH LANGEVIN DYNAMICS)，另外在 3.4 节，对其做连续化 SDE 推导的时候，扩散过程也是满足马尔可夫性的。这点和 NCSN 中从大到小尺度添加噪声训练分数匹配模型 NCSN 是不同的，其扰动核没有马尔可夫性假设，但是蕴含了马尔科夫性。具体来说，$p_{\sigma_{i}}(\mathbf{x} \mid \mathbf{x}_{\mathbf{0}})$ 对应的 $\mathbf{x}_{i}$ 满足如下的马尔科夫链：

$$\mathbf{x}_{i}=\mathbf{x}_{i-1}+\sqrt{\sigma_{i}^{2}-\sigma_{i-1}^{2}} \mathbf{z}_{i-1}, \quad i=1, \cdots, N \tag{4.35}$$

这里方差是 $\sigma_{i}^{2}-\sigma_{i-1}^{2}$ 的原因在于 SMLD 并不是 DDPM 那样递归的定义，而是在 $\mathbf{x}_{0}$ 上，逐级的增加扰动噪声

$\mathbf{x}_{\sigma_i} = \mathbf{x}_0 + \sigma_i \mathbf{I}$。上式连续化得到

$$\mathrm{d}\mathbf{x} = \sqrt{\frac{\mathrm{d}\left[\sigma^2(t)\right]}{\mathrm{d}t}}\,\mathrm{d}\mathbf{w}. \tag{4.36}$$

表达了 $\{\mathbf{x}(t)\}_{t=0}^{1}$ 的前向过程的 SDE 形式。

DDPM 的扰动核 $\left\{p_{\alpha_i}(\mathbf{x} \mid \mathbf{x}_0)\right\}_{i=1}^{N}$ 对应的离散马尔科夫链：

$$\mathbf{x}_i = \sqrt{1-\beta_i}\mathbf{x}_{i-1} + \sqrt{\beta_i}\mathbf{z}_{i-1}, \quad i = 1, \cdots, N \tag{4.37}$$

当 $N \to \infty$ 时，收敛到如下 SDE：

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w} \tag{4.38}$$

论文称 SMLD 的 SDE 为 Variance Exploding (VE) SDE，称 DDPM 的 SDE 为 Variance Preserving (VP) SDE，并给出了 sub-VP SDE：

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)\left(1 - e^{-2\int_0^t \beta(s)\mathrm{d}s}\right)}\mathrm{d}\mathbf{w} \tag{4.39}$$

其在似然函数的计算上更加准确。

当得到分数模型后，就可以利用 reverse-time SDE 采样得到生成样本。数值求解器也可以提供来自 SDEs 的近似轨迹，比如 Euler-Maruyama，Runge-Kutta methods 等方法，可以用来求解 reverse-time SDE，获得生成样本。另外 DDPM 的祖先采样法，不能用在 VE SDE，作者这里提出了 reverse diffusion samplers，具体分析细节可看附录 E 部分。因为这里有训练好的分数模型，有利于求解，而分数模型有对应的郎之万 MCMC 采样，因此作者综合考虑这些特性，给出了 Predictor-Corrector(PC) samplers 采样器，其结合了分数模型的采样特点，也考虑了 SDE 的数值求解，同时也考虑了 DDPM 等的采样方法。细节见附录部分。

Algorithm 2 PC sampling (VE SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2\mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}'_i \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, \sigma_{i+1})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 3 PC sampling (VP SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}'_i \leftarrow (2 - \sqrt{1-\beta_{i+1}})\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, i+1)$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\beta_{i+1}}\mathbf{z}$ Predictor
  for $j = 1$ **to** M **do** Corrector
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 4 Corrector algorithm (VE SDE).

Require: $\{\sigma_i\}_{i=1}^{N}, r, N, M.$
$\mathbf{x}_N^0 \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2\mathbf{I})$
for $i \leftarrow N$ to 1 **do**
  for $j \leftarrow 1$ to M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{g} \leftarrow \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i^{j-1}, \sigma_i)$
    $\epsilon \leftarrow 2(r\,\|\mathbf{z}\|_2 / \|\mathbf{g}\|_2)^2$
    $\mathbf{x}_i^j \leftarrow \mathbf{x}_i^{j-1} + \epsilon\,\mathbf{g} + \sqrt{2\epsilon}\,\mathbf{z}$
  $\mathbf{x}_{i-1}^0 \leftarrow \mathbf{x}_i^M$
return $\mathbf{x}_0^0$

Algorithm 5 Corrector algorithm (VP SDE).

Require: $\{\beta_i\}_{i=1}^{N}, \{\alpha_i\}_{i=1}^{N}, r, N, M.$
$\mathbf{x}_N^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i \leftarrow N$ to 1 **do**
  for $j \leftarrow 1$ to M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{g} \leftarrow \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i^{j-1}, i)$
    $\epsilon \leftarrow 2\alpha_i(r\,\|\mathbf{z}\|_2 / \|\mathbf{g}\|_2)^2$
    $\mathbf{x}_i^j \leftarrow \mathbf{x}_i^{j-1} + \epsilon\,\mathbf{g} + \sqrt{2\epsilon}\,\mathbf{z}$
  $\mathbf{x}_{i-1}^0 \leftarrow \mathbf{x}_i^M$
return $\mathbf{x}_0^0$

图 4.4: SDE Predictor Corrector

另外 SDE 有对应的常微分方程，

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t \tag{4.40}$$

该常微分方程的解，保持 SDE 所有边缘分布，且构成是确定的。作者将其称为 probability flow ODE。当分析模型得到恰当的训练后，它就是一个 neural network(Neural Ordinary Differential Equations)，且能用来精确的计算数据 (在量化的意义下) 的似然值。这块内容在“极大似然的计算”章节中细讲。

4.5 一些理论结果

4.6 采样加速

4.6.1 DDIM

将 DDPM 的前向马尔科夫链推广为非马尔科夫过程。

$$q_\sigma\left(x_{1:T} \mid x_0\right) := q_\sigma\left(x_T \mid x_0\right) \prod_{t=2}^{T} q_\sigma\left(x_{t-1} \mid x_t, x_0\right)$$

where $q_\sigma\left(\boldsymbol{x}_T \mid \boldsymbol{x}_0\right) = \mathcal{N}\left(\sqrt{\alpha_T}\boldsymbol{x}_0, \left(1-\alpha_T\right)\boldsymbol{I}\right)$ and for all $t > 1$,

$$q_\sigma\left(x_{t-1} \mid x_t, x_0\right) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \boldsymbol{I}\right).$$

The mean function is chosen to order to ensure that $q_\sigma\left(x_t \mid x_0\right) = \mathcal{N}\left(\sqrt{\alpha_t}x_0, \left(1-\alpha_t\right)\boldsymbol{I}\right)$ for all t (see Lemma 1 of Appendix B), so that it defines a joint inference distribution that matches the ”marginals” as desired. The forward process [3] can be derived from Bayes' rule:

$$q_\sigma\left(x_t \mid x_{t-1}, x_0\right) = \frac{q_\sigma\left(x_{t-1} \mid x_t, x_0\right) q_\sigma\left(x_t \mid x_0\right)}{q_\sigma\left(x_{t-1} \mid x_0\right)}$$

因为这篇文章是在 SGM 之后，所以也做了 ODE 的分析。推导过程和 ELBO 一致，不赘述。

后续文章 ANALYTIC-DPM 证明了 DDPM，DDIM 论文中轨迹中的最优均值和最优方差，均可解析解到，但是，解是用分数函数来表达的，其实就相当于没有，因为分数模型的训练就是 DDPM 模型的训练。ANALYTIC-DPM 一文的前向是 DDIM 的模式，但是在推导过程中，作者把均值用 DDIM 模型表达的式子替换为理论上等价的分数模型，因分数模型有具体的表达式，因此整个计算能走下去。

4.6.2 ODE 求解:dpm++ 等

DPM-Solver 是用 SDE 的思路，写出 DDPM 的 reverse time SDE 和 diffusion ODE，并利用变量替换法，给出了 ODE 的具体解，具体解的形式在 ODE 的数值求解中有充分的研究，因此能避免成百次的迭代，达到加速的效果。

DPM-Solver++ 是在 DPM-Solver 的基础上，考虑了引导因子 (guide scale)s，并在多阶展开逼近上，给出了较 DPM-Solver 更具体的多阶求解器，同时证明了 DDIM 是其一阶求解器。另外考虑引导因子，是因为扩散模型训练中发现增加分类引导 (calssify-free guide)，能带来更好的生成效果。

因为算法本身和训练无关，只用在采样生成样本中，因此法只需要不多的代码，就能实现，而且在实际中成为比较好的加速版的采样求解器。

4.6.3 蒸馏

4.7 一致性模型

PF ODE 将数据光滑的映射到噪声，一致性模型将 ODE 的迹上的任意点 $x_{t'}, x_T$ 映射到初始点 x_0。这些所有映射被称为一致性模型，因为其在一个迹上的输出都被映射到迹中初始点了。A notable property of our model is self-consistency: points on the same trajectory map to the same initial point.

这里的一致性主要为了加速采样的。两种训练方式：需要一个扩散模型 + 对应的 ODE，在 PF ODE 的迹上生成成对的相邻点。最小化这些成对点关于模型的输出的差 (L1/L2) 值。二种方式是单独训练一致性模型。蒸馏训练：

$$\mathcal{L}_{CD}^{N}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \boldsymbol{\phi}\right) := \mathbb{E}\left[\lambda\left(t_n\right) d\left(\boldsymbol{f}_{\boldsymbol{\theta}}\left(\mathbf{x}_{t_{n+1}}, t_{n+1}\right), \boldsymbol{f}_{\boldsymbol{\theta}^{-}}\left(\hat{\mathbf{x}}_{t_n}^{\phi}, t_n\right)\right)\right]$$

蒸馏方式可以将扩散模型蒸馏到一次采样。

独立训练基础：

Lemma 1. Let $\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \mathbf{x}_t \sim \mathcal{N}\left(\mathbf{x}; t^2 \boldsymbol{I}\right)$, and $p_t\left(\mathbf{x}_t\right) = p_{\text{data}}(\mathbf{x}) \otimes \mathcal{N}\left(\mathbf{0}, t^2 \boldsymbol{I}\right)$. We have

$$\nabla \log p_t(\mathbf{x}) = -\mathbb{E}\left[\frac{\mathbf{x}_t - \mathbf{x}}{t^2} \mid \mathbf{x}_t\right]. \tag{4.41}$$

一个随便的网络作为一致性函数的逼近表达：$\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t) F_{\boldsymbol{\theta}}(\mathbf{x}, t)$

4.7.1 LCM

自编码 $(\mathcal{E}, \mathcal{D})$：$z = \mathcal{E}(x), \bar{x} = \mathcal{D}(z)$ 将图像压缩到隐变量空间，在隐变量空间中，逆向扩散过程的 PF-ODE 为：

$$\frac{\mathbf{d}\boldsymbol{z}_t}{\mathbf{d}t} = f(t)\boldsymbol{z}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right), \quad \boldsymbol{z}_T \sim \mathcal{N}\left(\mathbf{0}, \tilde{\sigma}^2 \boldsymbol{I}\right) \tag{4.42}$$

其中 $\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right)$ 表示噪声预测模型，$\boldsymbol{c}$ 是条件因子，比如文本。

定义一致性函数：$f_{\boldsymbol{\theta}} : (\boldsymbol{z}_t, \boldsymbol{c}, t) \mapsto z_0$

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}, t) = c_{\text{skip}}(t)\boldsymbol{z} + c_{\mathbf{out}}(t)\left(\frac{\boldsymbol{z} - \sigma_t \hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{z}, \boldsymbol{c}, t)}{\alpha_t}\right) \tag{4.43}$$

其中 $c_{skip}(0) = 1, c_{skip}(0) = 0, \bar{\boldsymbol{\epsilon}}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right)$ 为一预训练扩散模型作为初始化的网络。

这里常微分的边缘解，用了 DPM++ 这种数值解方式。容易给出一致性蒸馏损失：

$$\mathcal{L}_{C\mathcal{D}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{-}; \Psi\right) = \mathbb{E}_{\boldsymbol{z}, \boldsymbol{c}, n}\left[d\left(\boldsymbol{f}_{\boldsymbol{\theta}}\left(\boldsymbol{z}_{t_{n+1}}, \boldsymbol{c}, t_{n+1}\right), \boldsymbol{f}_{\boldsymbol{\theta}^{-}}\left(\hat{\boldsymbol{z}}_{t_n}^{\Psi}, \boldsymbol{c}, t_n\right)\right)\right] \tag{4.44}$$

其中 $\hat{\boldsymbol{z}}_{t_n}^{\Psi}$ 表示使用 ODE 解 Ψ 从 $t_{n+1} \rightarrow t_n$：

$$\hat{\boldsymbol{z}}_{t_n}^{\Psi} - \boldsymbol{z}_{t_{n+1}} = \int_{t_{n+1}}^{t_n}\left(f(t)\boldsymbol{z}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right)\right)\mathbf{d}t \approx \Psi\left(\boldsymbol{z}_{t_{n+1}}, t_{n+1}, t_n, \boldsymbol{c}\right) \tag{4.45}$$

增加 CFG 对文字图像对齐，以及生成质量提高都能带来好处，在原始扩散模型中，CFG 使用如下：

$$\tilde{\boldsymbol{\epsilon}}_\theta\left(\boldsymbol{z}_t, \omega, \boldsymbol{c}, t\right) := (1+\omega)\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \boldsymbol{c}, t\right) - \omega\boldsymbol{\epsilon}_\theta\left(\boldsymbol{z}_t, \varnothing, t\right) \tag{4.46}$$

其中 ω 称为 guidance scale。将此式作为替换项即可。同时为了能使用 DDIM, DPM++ 来解 PF-ODE, 对替换后的

式子需做如下简化：

$$\begin{aligned}\hat{z}_{t_n}^{\Psi,\omega} - z_{t_{n+1}} &= \int_{t_{n+1}}^{t_n} \left(f(t) z_t + \frac{g^2(t)}{2\sigma_t} \tilde{\epsilon}_\theta \left(z_t, \omega, \boldsymbol{c}, t \right) \right) \mathrm{d}t \\ &= (1+\omega) \int_{t_{n+1}}^{t_n} \left(f(t) z_t + \frac{g^2(t)}{2\sigma_t} \epsilon_\theta \left(z_t, \boldsymbol{c}, t \right) \right) \mathrm{d}t - \omega \int_{t_{n+1}}^{t_n} \left(f(t) z_t + \frac{g^2(t)}{2\sigma_t} \epsilon_\theta \left(z_t, \varnothing, t \right) \right) \mathrm{d}t \\ &\approx (1+\omega) \Psi \left(z_{t_{n+1}}, t_{n+1}, t_n, \boldsymbol{c} \right) - \omega \Psi \left(z_{t_{n+1}}, t_{n+1}, t_n, \varnothing \right).\end{aligned} \tag{4.47}$$

Algorithm 1 Latent Consistency Distillation (LCD)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, ODE solver $\Psi(\cdot,\cdot,\cdot,\cdot)$, distance
EMA rate μ, noise schedule $\alpha(t), \sigma(t)$, guidance scale $[w_{\min}, w_{\max}]$, skipping interval k, and
Encoding training data into latent space: $\mathcal{D}_z = \{(\boldsymbol{z}, \boldsymbol{c}) | \boldsymbol{z} = E(\boldsymbol{x}), (\boldsymbol{x}, \boldsymbol{c}) \in \mathcal{D}\}$
$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$
repeat
  Sample $(\boldsymbol{z}, \boldsymbol{c}) \sim \mathcal{D}_z$, $n \sim \mathcal{U}[1, N-k]$ and $\omega \sim [\omega_{\min}, \omega_{\max}]$
  Sample $\boldsymbol{z}_{t_{n+k}} \sim \mathcal{N}(\alpha(t_{n+k})\boldsymbol{z}; \sigma^2(t_{n+k})\mathbf{I})$
  $\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega} \leftarrow \boldsymbol{z}_{t_{n+k}} + (1+\omega)\Psi(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, t_n, \boldsymbol{c}) - \omega\Psi(\boldsymbol{z}_{t_{n+k}}, t_{n+k}, t_n, \varnothing)$
  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \Psi) \leftarrow d(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}_{t_{n+k}}, \omega, \boldsymbol{c}, t_{n+k}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\boldsymbol{z}}_{t_n}^{\Psi,\omega}, \omega, \boldsymbol{c}, t_n))$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$
  $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu\boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$
until convergence

图 4.5: Latent Consistency Distillation

4.7.2 角色一致性

4.8 应用

- 1 千万文本对数据，模型参数 22M ： DeepSpeed ZeRO-2 + 8 V100 + 1M steps + bs=8 per GPU
- 5 千万图文对 +1 千万高质量人数据：48 NVIDIA H800 GPUs (80GB) + 2 per GPU (InstantID)
- 3 百万图文对数据，模型参数 1431.6M(367M)：10 epochs using 8 Tesla A100 GPUs+progressive
- 百万数据 + 万 ID+caption：8 A100 GPUs + two weeks + bs=48
- 256-coreTPU-v3 pod, training DiT-L takes 12 hours.
- 通常是 128*60=48*180：沉没成本比较大
- 组合思路难度小，混杂的库太多，数据集制作繁杂，比较占资源 (意义不大)
- 优点：1d 扩散，simplify unet, only one bvh(400+frames) data, any topology, 4GGPU memory, 1 hour training cost, 5 infer applications，long frames generator

4.8.1 多模态

排列组合 + 卡 + 数据，太多了。

4.8.1.1 VAE、CLIP、U-Net、U-ViT 等模块

4.8.1.2 条件控制

4.8.2 变大

4.8.2.1 SDXL

4.8.3 Finetune

4.8.3.1 LORA

4.8.4 3D 重建

4.8.5 动画动作生成

4.8.5.1 SinMDM

4.9 框架实现

4.9.1 Diffusers、SCEPTER

4.9.2 ComfyUI、WebUI、Fooocus

第五章 部署

5.1 量化

5.2 TensorRTt

5.3 TritonServer

5.4 OneDiff

第六章 商业

6.1 Sora

第七章 ElegantBook 写作示例

内容提要

- 积分定义 7.1
- Fubini 定理 7.1
- 最优性原理 7.1
- 柯西列性质 7.1.1
- 韦达定理

7.1 Lebesgue 积分

在前面各章做了必要的准备后，本章开始介绍新的积分。在 Lebesgue 测度理论的基础上建立了 Lebesgue 积分，其被积函数和积分域更一般，可以对有界函数和无界函数统一处理。正是由于 Lebesgue 积分的这些特点，使得 Lebesgue 积分比 Riemann 积分具有在更一般条件下的极限定理和累次积分交换积分顺序的定理，这使得 Lebesgue 积分不仅在理论上更完善，而且在计算上更灵活有效。

Lebesgue 积分有几种不同的定义方式。我们将采用逐步定义非负简单函数，非负可测函数和一般可测函数积分的方式。

由于现代数学的许多分支如概率论、泛函分析、调和分析等常常用到一般空间上的测度与积分理论，在本章最后一节将介绍一般的测度空间上的积分。

7.1.1 积分的定义

我们将通过三个步骤定义可测函数的积分。首先定义非负简单函数的积分。以下设 E 是 $\mathcal{R}^n$ 中的可测集。

定义 7.1 (可积性)

设 $f(x)=\sum_{i=1}^{k} a_i\chi_{A_i}(x)$ 是 E 上的非负简单函数，中文其中 $\{A_1,A_2,\dots,A_k\}$ 是 E 上的一个可测分割，$a_1,a_2,\dots,a_k$ 是非负实数。定义 f 在 E 上的积分为 $\int_a^b f(x)$

$$\int_E f\,dx=\sum_{i=1}^{k} a_i m(A_i)\pi\alpha\beta\sigma\gamma\nu\xi\epsilon\varepsilon.\oint_a^b\oint_a^b\prod_{i=1}^{n} \tag{7.1}$$

一般情况下 $0\le\int_E f\,dx\le\infty$。若 $\int_E f\,dx<\infty$，则称 f 在 E 上可积。

一个自然的问题是，Lebesgue 积分与我们所熟悉的 Riemann 积分有什么联系和区别？在 4.4 在我们将详细讨论 Riemann 积分与 Lebesgue 积分的关系。这里只看一个简单的例子。设 $D(x)$ 是区间 $[0,1]$ 上的 Dirichlet 函数。即 $D(x)=\chi_{Q_0}(x)$，其中 Q_0 表示 $[0,1]$ 中的有理数的全体。根据非负简单函数积分的定义，$D(x)$ 在 $[0,1]$ 上的 Lebesgue 积分为

$$\int_0^1 D(x)dx=\int_0^1 \chi_{Q_0}(x)dx=m(Q_0)=0 \tag{7.2}$$

即 $D(x)$ 在 $[0,1]$ 上是 Lebesgue 可积的并且积分值为零。但 $D(x)$ 在 $[0,1]$ 上不是 Riemann 可积的。

有界变差函数是与单调函数有密切联系的一类函数。有界变差函数可以表示为两个单调递增函数之差。与单调函数一样，有界变差函数几乎处处可导。与单调函数不同，有界变差函数类对线性运算是封闭的，它们构成一线空间。练习题 7.1 是一个性质的证明。

练习 7.1 设 $f\notin\in L(\mathcal{R}^1)$，$g$ 是 $\mathcal{R}^1$ 上的有界可测函数。证明函数

$$I(t)=\int_{\mathcal{R}^1} f(x+t)g(x)dx\quad t\in\mathcal{R}^1 \tag{7.3}$$

是 $\mathcal{R}^1$ 上的连续函数。

解 即 $D(x)$ 在 $[0,1]$ 上是 Lebesgue 可积的并且积分值为零。但 $D(x)$ 在 $[0,1]$ 上不是 Riemann 可积的。

证明 即 $D(x)$ 在 $[0,1]$ 上是 Lebesgue 可积的并且积分值为零。但 $D(x)$ 在 $[0,1]$ 上不是 Riemann 可积的。

定理 7.1 (Fubini 定理)

（1）若 $f(x,y)$ 是 $\mathcal{R}^p \times \mathcal{R}^q$ 上的非负可测函数，则对几乎处处的 $x \in \mathcal{R}^p$，$f(x,y)$ 作为 y 的函数是 $\mathcal{R}^q$ 上的非负可测函数，$g(x) = \int_{\mathcal{R}^q} f(x,y)dy$ 是 $\mathcal{R}^p$ 上的非负可测函数。并且

$$\int_{\mathcal{R}^p \times \mathcal{R}^q} f(x,y)dxdy = \int_{\mathcal{R}^p} \left(\int_{\mathcal{R}^q} f(x,y)dy \right) dx. \tag{7.4}$$

（2）若 $f(x,y)$ 是 $\mathcal{R}^p \times \mathcal{R}^q$ 上的可积函数，则对几乎处处的 $x \in \mathcal{R}^p$，$f(x,y)$ 作为 y 的函数是 $\mathcal{R}^q$ 上的可积函数，并且 $g(x) = \int_{\mathcal{R}^q} f(x,y)dy$ 是 $\mathcal{R}^p$ 上的可积函数。而且 (7.4) 成立。

笔记 在本模板中，引理（lemma），推论（corollary）的样式和定理 7.1 的样式一致，包括颜色，仅仅只有计数器的设置不一样。

我们说一个实变或者复变量的实值或者复值函数是在区间上平方可积的，如果其绝对值的平方在该区间上的积分是有限的。所有在勒贝格积分意义下平方可积的可测函数构成一个希尔伯特空间，也就是所谓的 L^2 空间，几乎处处相等的函数归为同一等价类。形式上，L^2 是平方可积函数的空间和几乎处处为 0 的函数空间的商空间。

命题 7.1 (最优性原理)

如果 u^* 在 $[s,T]$ 上为最优解，则 u^* 在 $[s,T]$ 任意子区间都是最优解，假设区间为 $[t_0,t_1]$ 的最优解为 u^*，则 $u(t_0) = u^*(t_0)$，即初始条件必须还是在 u^* 上。

我们知道最小二乘法可以用来处理一组数据，可以从一组测定的数据中寻求变量之间的依赖关系，这种函数关系称为经验公式。本课题将介绍最小二乘法的精确定义及如何寻求点与点之间近似成线性关系时的经验公式。假定实验测得变量之间的 n 个数据，则在平面上，可以得到 n 个点，这种图形称为“散点图”，从图中可以粗略看出这些点大致散落在某直线近旁，我们认为其近似为一线性函数，下面介绍求解步骤。

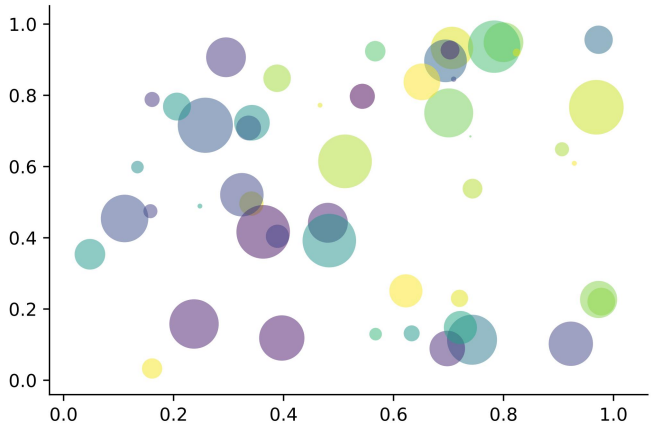

图 7.1: 散点图示例 $\hat{y} = a + bx$

以最简单的一元线性模型来解释最小二乘法。什么是一元线性模型呢？监督学习中，如果预测的变量是离散的，我们称其为分类（如决策树，支持向量机等），如果预测的变量是连续的，我们称其为回归。回归分析中，如果只包括一个自变量和一个因变量，且二者的关系可用一条直线近似表示，这种回归分析称为一元线性回归分析。如果回归分析中包括两个或两个以上的自变量，且因变量和自变量之间是线性关系，则称为多元线性回归分析。对于二维空间线性是一条直线；对于三维空间线性是一个平面，对于多维空间线性是一个超平面。

性质 柯西列的性质

1. $\{x_k\}$ 是柯西列，则其子列 $\{x_k^i\}$ 也是柯西列。
2. $x_k \in \mathcal{R}^n$，$\rho(x,y)$ 是欧几里得空间，则柯西列收敛，$(\mathcal{R}^n, \rho)$ 空间是完备的。

结论 回归分析（regression analysis）是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。运用十分广泛，回归分析按照涉及的变量的多少，分为一元回归和多元回归分析；按照因变量的多少，可分为简单回归分析和多重回归分析；按照自变量和因变量之间的关系类型，可分为线性回归分析和非线性回归分析。

第七章 练习

1. 设 A 为数域 K 上的 n 级矩阵。证明：如果 K^n 中任意非零列向量都是 A 的特征向量，则 A 一定是数量矩阵。
2. 证明：不为零矩阵的幂零矩阵不能对角化。
3. 设 $A = (a_{ij})$ 是数域 K 上的一个 n 级上三角矩阵，证明：如果 $a_{11} = a_{22} = \cdots = a_{nn}$，并且至少有一个 $a_{kl} \neq 0 (k < l)$，则 A 一定不能对角化。

参考文献

[1] Charles T Carlstrom and Timothy S Fuerst. "Agency Costs, Net Worth, and Business Fluctuations: A Computable General Equilibrium Analysis". In: *The American Economic Review* (1997), pp. 893–910. ISSN: 0002-8282.

[2] 方军雄. "所有制、制度环境与信贷资金配置". In: 经济研究 12 (2007), pp. 82–92. ISSN: 0577-9154.

[3] Qiang Li, Liwen Chen, and Yong Zeng. "The Mechanism and Effectiveness of Credit Scoring of P2P Lending Platform: Evidence from Renrendai.com". In: *China Finance Review International* 8.3 (2018), pp. 256–274.

[4] 刘凤良, 章潇萌, and 于泽. "高投资、结构失衡与价格指数二元分化". In: 金融研究 02 (2017), pp. 54–69. ISSN: 1002-7246.

[5] 吕捷 and 王高望. "CPI 与 PPI "背离" 的结构性解释". In: 经济研究 50.04 (2015), pp. 136–149. ISSN: 0577-9154.

[6] Vincenzo Quadrini. "Financial Frictions in Macroeconomic Fluctuations". In: *FRB Richmond Economic Quarterly* 97.3 (2011), pp. 209–254.

[7] Pascal Vincent. "A Connection Between Score Matching and Denoising Autoencoders". In: *Neural Computation* 23.7 (2011), pp. 1661–1674. DOI: 10.1162/NECO_a_00142.

[8] Ling Yang et al. *Diffusion Models: A Comprehensive Survey of Methods and Applications*. 2023. arXiv: 2209.00796 [cs.LG].

附录 A 基本数学工具

本附录包括了计量经济学中用到的一些基本数学，我们扼要论述了求和算子的各种性质，研究了线性和某些非线性方程的性质，并复习了比例和百分数。我们还介绍了一些在应用计量经济学中常见的特殊函数，包括二次函数和自然对数，前 4 节只要求基本的代数技巧，第 5 节则对微分学进行了简要回顾；虽然要理解本书的大部分内容，微积分并非必需，但在一些章末附录和第 3 篇某些高深专题中，我们还是用到了微积分。

A.1 求和算子与描述统计量

求和算子是用以表达多个数求和运算的一个缩略符号，它在统计学和计量经济学分析中扮演着重要作用。如果 $\{x_i : i = 1, 2, \ldots, n\}$ 表示 n 个数的一个序列，那么我们就把这 n 个数的和写为：

$$\sum_{i=1}^{n} x_i \equiv x_1 + x_2 + \cdots + x_n \tag{A.1}$$