

2025 年 12 月 1 日至 12 月 7 日周报

何瑞杰

中山大学, 大湾区大学

目录

1. 项目进展	2
1.1. 使用神经网络学习生命游戏的演化动力学	2
2. 文献阅读	3
2.1. Diffusion Schrödinger Bridge with Application to Score-Based Generative Modeling [1]	3
3. 学习进度	11
3.1. 实分析	11
4. 下周计划	13
5. 附录	14
参考文献	15

速 览

本周大部分时间花在了阅读 Schrödinger 桥论文[1], 它给出了一个以 Schrödinger 桥问题观察生成模型的视角。一般的薛定谔桥问题(或静态薛定谔桥问题)没有闭式解, 作者转而通过修改 IPF 算法使之适配分数匹配生成模型的框架, 并采用迭代的方式学习薛定谔桥。

在生命游戏项目方面, 本周实现了一个极小的朴素规则提取函数, 在 B3678/S34678 数据集上可以成功提取规则。

最后本周学习了 Stein 实分析中开篇的少量内容。

1. 项目进展

1.1. 使用神经网络学习生命游戏的演化动力学

本周对训练网络在全部数据集上的统计结果套用简单的要求从而获取提取的规则序列。具体的说，提取统计中转换频数（如统计得到的 5 个存活邻居下从死亡转换为存活的案例数）超过全体转换频数乘以预测错误率的百分之六十作为“入围”标准，然后再在这个入围标准下分别筛选当前时刻下死亡或存活，并在下一时刻预测存活的频数超过预测为死亡频数的 10 倍的规则作为最终得到的规则。对应的代码片段如下

```
# class RuleSimulatorStats
def infer_rule_str(self, counters, acc) -> Tuple[List, List]:
    list_str = lambda x: list(map(lambda k: str(int(k)), x))

    dd, dl = sum(counters[0].values()), sum(counters[1].values())
    ld, ll = sum(counters[2].values()), sum(counters[3].values())

    d_th = (1-acc)*(dd+dl)
    l_th = (1-acc)*(ld+ll)

    d_all = counters[0] + counters[1]
    l_all = counters[2] + counters[3]

    filtered_b = sorted(list(filter(lambda x: x[1]>0.6*d_th, d_all.items())),
                        key=lambda x: x[0])
    filtered_s = sorted(list(filter(lambda x: x[1]>0.6*l_th, l_all.items())),
                        key=lambda x: x[0])

    born = []
    survive = []

    for i, _ in filtered_b:
        if counters[1][i] > 10 * counters[0][i]:
            born.append(i)
    for i, _ in filtered_s:
        if counters[3][i] > 10 * counters[2][i]:
            survive.append(i)
    return list_str(born), list_str(survive)
```

这样朴素的规则提取函数在 B3678/S34678 可以成功推出真实的规则。

2. 文献阅读

2.1. Diffusion Schrödinger Bridge with Application to Score-Based Generative Modeling [1]

[NIPS 2021](#) | [Valentin De Bortoli et al.](#)

本文以 Schrödinger 桥 (SB) 的视角构建了一个新的生成模型。本质上是建立从人为规定的先验分布 p_{prior} 到未知数据分布 p_{data} 的最优传输。由于一般形式的 SB 难以求解，本文作者于是退而求其次，考虑迭代求解的 IPF 算法，而最终又将它转化为分数匹配模型可以使用的范畴。除此之外，本文的大部分篇幅和附录给出了有关分数匹配模型的拟合能力以及 SB 和离散情况 IPF 算法的收敛性证明，为模型提供理论保证。除此之外，本文指出该方法相比于扩散模型需要更少的步数。

2.1.1. 记号

记号	表达式	意义
\mathcal{C}	$\mathcal{C}([0, T], \mathbb{R}^d)$	从 $[0, T]$ 到 \mathbb{R}^d 的连续函数全体
$\mathcal{B}(\mathcal{C})$	—	\mathcal{C} 上的所有 Borel 集
$\mathcal{P}(\mathbf{E})$	—	可测空间 \mathbf{E} 上的概率测度的全体
\mathcal{P}_ℓ	$\mathcal{P}((\mathbb{R}^d)^\ell)$	—
$H(p)$	$-\int_{\mathbb{R}^d} p(x) \log p(x) \, dx$	p 的熵
$D_{\text{KL}}(p q)$	$\int_{\mathbb{R}^d} p(x) \log \frac{p(x)}{q(x)} \, dx$	p 和 q 的 Kullback-Leiber (KL) 散度
$J(\mu, \nu)$	$D_{\text{KL}}(\mu \nu) + D_{\text{KL}}(\nu, \mu)$	概率测度 μ 和 ν 的 Jeffrey 散度
$\ \cdot\ _{\text{TV}}$		全变差范数
\mathbb{P}^R		路径测度 \mathbb{P} 的时间逆

2.1.2. 去噪扩散模型、分数匹配和逆时 SDE 回顾

2.1.2.1. 离散情形

考虑从数据分布 p_{data} 出发的 Markov 链，它的前向转移概率是 $p_{k+1|k}$, $k \in \{0, 1, \dots, N-1\}$ 。于是 $\{x_k\}_{k=1}^N$ 的联合分布可以写为

$$p(x_{0:N}) = p_0(x_0) \prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k) \quad (1)$$

同样，我们也可以将其表示为反向转移概率是乘积：

$$p(x_{0:N}) = p_N(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}) \quad (2)$$

我们希望 p_N 逼近先验分布 p_{prior} ，从而可以从先验分布出发，沿着式 2 演化得到未知数据分布 p_{data} ，这就是 DDPM 中提出的[祖先采样 \(Ancestral Sampling\)](#)。一般地，我们假定该 Markov 链的正向转移概率为 $p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1}f(x_k), 2\gamma_{k+1}\mathbf{I})$ ，要得到逆向转移概率 $p_{k|k+1}$ ，根据 Bayes 公式和一些其他技巧，可以得到

$$\begin{aligned}
& p_{k|k+1}(x_k|x_{k+1}) \\
&= \frac{p_{k+1|k}(x_{k+1}|x_k)p_k(x_k)}{p_{k+1}(x_{k+1})} = p_{k+1|k}(x_{k+1}|x_k) \cdot \exp[\log p_k(x_k) - \log p_{k+1}(x_{k+1})] \\
&= p_{k+1|k}(x_{k+1}|x_k) \cdot \exp[\log p_{k+1}(x_k) - \log p_{k+1}(x_{k+1})] \quad p_k \approx p_{k+1} \\
&= p_{k+1|k}(x_{k+1}|x_k) \cdot \exp\langle \nabla \log p_{k+1}(x_{k+1}), x_k - x_{k+1} \rangle \quad \text{在 } x_{k+1} \text{ 处 Taylor 展开到一阶} \\
&= C \exp \left\{ -\frac{\|x_{k+1} - x_k - \gamma_{k+1}f(x_k)\|^2 + 2\langle 2\gamma_{k+1} \nabla \log p_{k+1}(x_{k+1}), x_k - x_{k+1} \rangle}{2 \cdot 2\gamma_{k+1}} \right\} \quad (3) \\
&= C \cdot C' \cdot \exp \left\{ -\frac{\|x_{k+1} - x_k - \gamma_{k+1}f(x_k) + 2\gamma_{k+1} \nabla \log p_{k+1}(x_{k+1})\|^2}{2 \cdot 2\gamma_{k+1}} \right\} \\
&\approx C'' \exp \left\{ -\frac{\|x_{k+1} - x_k - \gamma_{k+1}f(x_{k+1}) + 2\gamma_{k+1} \nabla \log p_{k+1}(x_{k+1})\|^2}{2 \cdot 2\gamma_{k+1}} \right\} \quad x_k \approx x_{k+1} \\
&= \mathcal{N}(x_k; x_{k+1} - \gamma_{k+1}f(x_{k+1}) + \nabla \log p_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I})
\end{aligned}$$

由于 $\nabla \log p_{k+1}(x_k + 1)$ 事先未知, 可以用分数匹配的方法训练一个网络 $s_\theta(x, k + 1)$ 拟合之, 具体原因如下: 假设条件概率 $p_{k+1|0}(x_{k+1}|x_0)$ 已经给定, 那么 $p_{k+1|k}(x_{k+1}|x_k)$ 就可以写为

$$p_{k+1}(x_{k+1}) = \int p_0(x_0)p_{k+1|0}(x_{k+1}|x_0) \, dx_0 \quad (4)$$

可以得到等式左边的对数梯度:

$$\begin{aligned}
\nabla_{x_{k+1}} \log p_{k+1}(x_{k+1}) &= \nabla_{x_{k+1}} \log \int p_0(x_0)p_{k+1|0}(x_{k+1}|x_0) \, dx_0 \\
&= \frac{\nabla_{x_{k+1}} \int p_0(x_0)p_{k+1|0}(x_{k+1}|x_0) \, dx_0}{p_{k+1}(x_{k+1})} \\
&= \frac{\int p_0(x_0)\nabla_{x_{k+1}} p_{k+1|0}(x_{k+1}|x_0) \, dx_0}{p_{k+1}(x_{k+1})} \quad (5) \\
&= \int \frac{p_0(x_0)p_{k+1|0}(x_{k+1}|x_0)}{p_{k+1}(x_{k+1})} \cdot \nabla_{x_{k+1}} \log p_{k+1|0}(x_{k+1}|x_0) \, dx_0 \\
&= \int p_{0|k+1}(x_0|x_{k+1}) \nabla_{x_{k+1}} \log p_{k+1|0}(x_{k+1}|x_0) \, dx_0 \, dx_0 \\
&= \mathbb{E}_{p_{0|k+1}} [\nabla_{x_{k+1}} \log p_{k+1|0}(x_{k+1}|x_0)]
\end{aligned}$$

因此分数网络的学习目标为 $\mathbb{E}_{p_{0|k+1}} [\nabla_{x_{k+1}} \log p_{k+1|0}(x_{k+1}|x_0)]$ 这样就自然地得到了下面的分数匹配的优化目标:

$$\text{minimize}_{\theta} \sum_{k=1}^N \mathbb{E}_{p_{0,k}} [\|s_{\theta}(x_k, k) - \nabla_{x_k} \log p_{k|0}(x_k|x_0)\|^2] \quad (6)$$

其中 $p_{0,k} = p_0 p_{k|0}$ 。如果无法计算 $p_{k|0}$, 就将上述目标中的 $k|0$ 一项改为 $k|k-1$, 和 SMLD 论文中的形式相同。最后, 要从先验分布 p_{prior} 生成样本, 只需从中采样 $x_N \sim p_{\text{prior}}$, 然后进行 Langevin 采样。

2.1.2.2. 连续情形

考虑将上一节中的离散情形求极限, 发现离散情形为遵循下面 SDE 的随机过程 $(\mathbf{X}_t)_{t \in [0, T]}$ Euler-Maruyama 离散化形式:

$$d\mathbf{X}_t = f(\mathbf{X}_t) dt + \sqrt{2} d\mathbf{W}, \quad \mathbf{X}_0 \sim p_0 = p_{\text{data}}. \quad (7)$$

其中 $\mathbf{W}(\cdot)$ 是和 \mathbf{X} 维数相同的标准 Brown 运动, 映射 $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ 为使得强解存在而恰当选取的函数。一般令其为线性函数, 即 $f(\mathbf{x}) = -\alpha\mathbf{x}$ 。在此种情况下, 其逆向过程 $(\mathbf{Y}_t)_{t \in [0, T]} = (\mathbf{X}_{T-t})_{t \in [0, T]}$ 满足下面的逆向 SDE:

$$d\mathbf{Y}_t = [-f(\mathbf{Y}_t) + 2\nabla \log p_{T-t}(\mathbf{Y}_t)] dt + \sqrt{2} d\mathbf{W}_t \quad (8)$$

读者可以看见, 其 Euler-Maruyama 离散化的情形正好对应式 3。值得注意的是, DDPM 和 SMLD 论文中对应的正向 SDE 都为 Ornstein-Uhlenbeck 过程。下面的定理给出了分数匹配模型应用于逆时 OU 过程预测 X_0 数据分布 $\mathcal{L}(X_0)$ 的误差界:

定理 2.1: 假设存在某个 $M \geq 0$, 是的对任意 $t \in [0, T]$ 和 $x \in \mathbb{R}^d$, 都有

$$\|s_\theta(x, t) - \nabla \log p_t(x)\| \leq M, \quad (9)$$

其中 $s_{\theta^*} \in \mathcal{C}([0, T] \times \mathbb{R}^d, \mathbb{R}^d)$ 。并假设数据分布 $p_{\text{data}} \in \mathcal{C}^3(\mathbb{R}^d, \mathbb{R}_{>0})$ 有界, 且满足一些条件¹时, 有依照上述方法生成得到的分布 $\mathcal{L}(X_0)$ 和数据分布 p_{data} 之间的全变差范数距离被控制:

$$\|\mathcal{L}(X_0) - p_{\text{data}}\|_{\text{TV}} \leq C(\alpha, M, T) \quad (10)$$

其中 $C(\alpha, M, T)$ 是一个依赖于 α , M 和 T 的常数。

换言之, 该定理给出了这样的叙述: 如果我们有一个不甚准确的分数模型 s_θ , 那么我们依靠它生成得到的分布 $\mathcal{L}(X_0)$ 相比于 p_{data} 可以近到什么程度。该定理还揭示了一个权衡的问题。当 α 比较大, 即 SDE 中的偏移项较大时, 需要更小的数值模拟步长, 以及更小的混合总时间 T ; α 较小, 即混合速度较慢时, 需要更大的混合步长和更长的混合总时间以趋近先验分布 p_{prior} 。

2.1.3. Schrödinger 桥

在生成的语境中, 先验分布 p_{prior} 和数据分布 p_{data} 之间的 Schrödinger 桥指的是满足下面条件的路径测度 $\pi^* \in \mathcal{P}_{N+1}$:

$$\pi^* = \arg \min \{D_{\text{KL}}(\pi|p) : \pi \in \mathcal{P}_{N+1}, \pi_0 = p_{\text{data}}, \pi_N = p_{\text{prior}}\} \quad (11)$$

自然地, 如果我们得到了 π^* , 自然就可以通过它构造出前文中提到的逆向 Markov 链的转移概率 $\pi_{k|k+1}^*$ 。将视角缩小到 Schrödinger 桥的两头, 能得到所谓静态 Schrödinger 桥问题。给定路径测度 $\pi \in \mathcal{P}_{N+1}$, $\pi_{|0, N}$ 指的是 $X_{1:N-1}$ 的条件分布。对于路径测度 $\pi, p \in \mathcal{P}_{N+1}$, 有下面的恒等关系

¹见论文原文

$$\begin{aligned}
D_{\text{KL}}(\pi|p) &= \int \pi(x) \log \frac{\pi(x)}{p(x)} \, d x \\
&= \int \pi(x) \log \frac{\pi_{0,N}(x_0, x_N)}{p_{0,N}(x_0, x_N)} \, d x + \int \pi(x) \log \frac{\pi_{|0,N}(x_{1:N})}{p_{|0,N}(x_{1:N})} \, d x \\
&= \int \pi_{|0,N}(x_{1:N}) \int \pi_{0,N}(x_0, x_N) \log \frac{\pi_{0,N}(x_0, x_N)}{p_{0,N}(x_0, x_N)} \, d x_{0,N} \, d x_{1:N} \\
&\quad + \int \pi_{0,N}(x_0, x_N) \int \pi_{|0,N}(x_{1:N}) \log \frac{\pi_{|0,N}(x_{1:N})}{p_{|0,N}(x_{1:N})} \, d x_{1:N} \, d x_{0,N} \\
&= \underbrace{\mathbb{E}_{\pi_{|0,N}}[D_{\text{KL}}(\pi_{0,N}|p_{0,N})]}_{\text{常数}} + \underbrace{\mathbb{E}_{\pi_{0,N}}[D_{\text{KL}}(\pi_{|0,N}|p_{|0,N})]}_{\text{随机变量}} \\
&= D_{\text{KL}}(\pi_{0,N}|p_{0,N}) + \mathbb{E}_{\pi_{0,N}}[D_{\text{KL}}(\pi_{|0,N}|p_{|0,N})]
\end{aligned} \tag{12}$$

注意, $D_{\text{KL}}(\pi|p)$ 是 Schrödinger 桥定义中的优化目标。根据上面的结果可以拆成两部分, 第一部分 $D_{\text{KL}}(\pi_{0,N}|p_{0,N})$ 是边缘分布的匹配项, 第二部分 $\mathbb{E}_{\pi_{0,N}}[D_{\text{KL}}(\pi_{|0,N}|p_{|0,N})]$ 是中间路径的匹配项。匹配两端的边缘分布的版本就是 **静态 Schrödinger 桥**, 下式给出了该问题的解:

$$\pi^{s,*} = \arg \min \{ D_{\text{KL}}(\pi^s|p_{0,N}) : \pi \in \mathcal{P}_2, \pi_0^s = p_{\text{data}}, \pi_N^s = p_{\text{prior}} \} \tag{13}$$

得到该解后, 完整的 Schrödinger 桥问题的解可以写成

$$\pi^* = \pi^{s,*} \otimes p_{|0,N} \tag{14}$$

其中 $p_{|0,N}$ 可以是任给的一个分布, 例如可以是高斯转移核 $p_{k+1|k} = \mathcal{N}(x_k, \sigma_{k+1}^2)$, 这样就得到了 Schrödinger 桥的解: 若参考路径测度对应的是标准 Brown 运动, 则将上述构造代入上述对 $D_{\text{KL}}(\pi|p)$ 的分解中, 第二项路径分布匹配项为 0, 因此 π^* 确实是 Schrödinger 桥。

如果恰当的重写静态 Schrödinger 桥定义中的优化目标, 我们可以看到它与最优传输的联系:

$$\begin{aligned}
D_{\text{KL}}(\pi^s|p_{0,N}) &= \int \pi^s(x_{0,N}) \log \frac{\pi^s(x_{0,N})}{p_{0,N}(x_{0,N})} \, d x_{0,N} \\
&= \int \pi^s(x_{0,N}) \log \pi^s(x_{0,N}) \, d x_{0,N} - \int \pi^s(x_{0,N}) \log p_{N|0}(x_N|x_0) \, d x_{0,N} \\
&\quad - \int \pi^s(x_{0,N}) \log p_0(x_0) \, d x_{0,N} \\
&= H(\pi^s) - \mathbb{E}_{\pi^s}[\log p_{N|0}(x_N|x_0)] + \text{constant}
\end{aligned} \tag{15}$$

因此静态 Schrödinger 桥也可以写成下面带有熵正则化的最优传输问题:

$$\pi^{s,*} = \arg \min \{ -\mathbb{E}_{\pi^s}[\log p_{N|0}(x_N|x_0)] + H(\pi^s) : \pi \in \mathcal{P}_2, \pi_0^s = p_{\text{data}}, \pi_N^s = p_{\text{prior}} \} \tag{16}$$

若取 $p_{k+1|k}$ 为高斯核, 均值为 0, 方差为 σ_k , 则 $p_{N|0}(x_N|x_0) = \mathcal{N}(x_N; x_0, \sigma^2)$, 其中 $\sigma^2 = \sum_{k=1}^N \sigma_k^2$ 。因此式 16 也可以写为

$$\pi^{s,*} = \arg \min \{ \mathbb{E}_{\pi^s}[\|x_0 - x_N\|^2] - 2\sigma^2 H(\pi^s) : \pi \in \mathcal{P}_2, \pi_0^s = p_{\text{data}}, \pi_N^s = p_{\text{prior}} \} \tag{17}$$

2.1.4. IPF (Iterative Proportional Fitting) 算法

遗憾的是, 大多数非平凡情况下的 SB 都没有闭式解。但我们可以使用 IPF 算法迭代求解。取 $n \in \mathbb{N}$, 和初始路径测度 $\pi^0 = p$, 然后循环执行下面的迭代操作:

$$\begin{aligned}\pi^{2n+1} &= \arg \min \{ D_{\text{KL}}(\pi | \pi^{2n}) : \pi \in \mathcal{P}_{N+1}, \pi_N = p_{\text{prior}} \} \\ \pi^{2n+2} &= \arg \min \{ D_{\text{KL}}(\pi | \pi^{2n+1}) : \pi \in \mathcal{P}_{N+1}, \pi_0 = p_{\text{data}} \}\end{aligned}\quad (18)$$

这相当于将 π^0 重复地在 KL 散度的意义下分别向限制了一个端点边缘分布的空间中投影。不过这样的形式依然难以计算。作者在引入改进前, 首先提到了下面的命题

命题 2.1: 假设 $D_{\text{KL}}(p_{\text{data}} \otimes p_{\text{prior}} | p_{0,N}) < \infty$, 则对任意 $n \in \mathbb{N}$, π^{2n} 和 π^{2n+1} 分别在 Lebesgue 测度 p^n 和 q^n 下总有正密度, 且对任意 $x_{0:N} \in \mathcal{X}$, 我们有 $p^0(x_{0:N}) = p(x_{0:N})$ 以及

$$\begin{aligned}q^n(x_{0:N}) &= p_{\text{prior}}(x_N) \prod_{k=0}^{N-1} p_{k|k+1}^n(x_k | x_{k+1}) \\ p^{n+1}(x_{0:N}) &= p_{\text{data}}(x_0) \prod_{k=0}^{N-1} q_{k+1|k}^n(x_{k+1} | x_k)\end{aligned}\quad (19)$$

在实际操作中, 我们使用 Bayes 公式相互转化 $p_{k|k+1}^n$ 和 $q_{k+1|k}^n$ 。我们最初使用加噪过程得到的联合分布 p 作为初始值 p^0 。在第 $2n$ 步时, 有 $\pi^{2n} = p^n$ 。从 p^n 得到的逆向过程 $p_{k|k+1}^n$ 结合上数据分布 p_{data} 就得到了逆向过程 $\pi^{2n+1} = q^n$ 。最后 q^n 对应的正向过程 $q_{k+1|k}^n$ 又定义了新的 $\pi^{2n+2} = p^{n+1}$ 。根据前文对逆向过程分布的推导, 我们可以得到前向转移分布 $p_{k+1|k}^n(x_{k+1} | x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f_k^n(x_k), 2\gamma_{k+1} \mathbf{I})$ 对应的逆向转移分布 $p_{k|k+1}^n(x_k | x_{k+1}) \approx \mathcal{N}(x_k; x_{k+1} + \gamma_{k+1} b_{k+1}^n(x_{k+1}), 2\gamma_{k+1} \mathbf{I})$ 其中 $b_{k+1}^n(x_{k+1}) = -f_k^b(x_{k+1}) + 2\nabla \log p_{k+1}^n(x_{k+1})$ 。我们还可以用同样的方法估计 $p_{k+1|k}^{n+1}(x_{k+1} | x_k)$ 为

$$\begin{aligned}p_{k+1|k}^{n+1}(x_{k+1} | x_k) &\approx \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f_k^{n+1}(x_k), 2\gamma_{k+1} \mathbf{I}) \\ &= \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} [-b_{k+1}^n(x_k) + 2\nabla \log q_k^n(x_k)], 2\gamma_{k+1} \mathbf{I})\end{aligned}\quad (20)$$

于是可以得到 $f_k^{n+1}(x_k) = f_k^n(x_k) - 2\nabla \log p_{k+1}^n(x_k) + 2\nabla \log q_k^n(x_k)$, 我们可以通过训练分数网络达到估计 f_k^{n+1} 和 b_k^{n+1} 的目的。下面的命题给出了对这二者迭代估计的可行性。

命题 2.2: 假设任意 $n \in \mathbb{N}$, $k \in \{0, \dots, N-1\}$ 都有

$$\begin{aligned} q_{k|k+1}^n(x_k|x_{k+1}) &= \mathcal{N}(x_k; B_{k+1}^n(x_{k+1}), 2\gamma_{k+1}\mathbf{I}) \\ p_{k+1|k}^n(x_{k+1}|x_k) &= \mathcal{N}(x_{k+1}; F_k^n(x_k), 2\gamma_{k+1}\mathbf{I}) \end{aligned} \quad (21)$$

其中 $B_{k+1}^n(x) = x + \gamma_{k+1}b_{k+1}^n(x)$, $F_k^n(x) = x + \gamma_{k+1}f_k^n(x)$; 则对任意满足前述条件的 n, k , 都有

$$\begin{aligned} B_{k+1}^n &= \arg \min_{B \in L^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p_{k,k+1}^n} \left[\|B(X_{k+1}) - [X_{k+1} + F_k^n(X_k) - F_k^n(X_{k+1})]\|^2 \right] \\ F_k^{n+1} &= \arg \min_{B \in L^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p_{k,k+1}^n} \left[\|F(X_k) - [X_{k+1} + B_{k+1}^n(X_{k+1}) - B_{k+1}^n(X_k)]\|^2 \right] \end{aligned} \quad (22)$$

我们可以分别用两个神经网络来学习 B_k^n 和 F_k^n , 然后迭代执行上面的两个优化步骤至收敛, 就得到了切实可行的 Schrödinger 桥生成模型。

Algorithm 1 Diffusion Schrödinger Bridge

```

1: for  $n \in \{0, \dots, L\}$  do
2:   while not converged do
3:     Sample  $\{X_{k,j}^j\}_{j=0}^{N,M}$ , where  $X_0^j \sim p_{\text{data}}$ , and
        $X_{k+1}^j = F_{\alpha^n}(k, X_k^j) + \sqrt{2\gamma_{k+1}}Z_{k+1}^j$ 
4:     Compute  $\hat{\ell}_n^b(\beta^n)$  approximating (12)
5:      $\beta^n \leftarrow \text{Gradient Step}(\hat{\ell}_n^b(\beta^n))$ 
6:   end while
7:   while not converged do
8:     Sample  $\{X_{k,j}^j\}_{j=0}^{N,M}$ , where  $X_N^j \sim p_{\text{prior}}$ , and
        $X_{k-1}^j = B_{\beta^n}(k, X_k^j) + \sqrt{2\gamma_k}\bar{Z}_k^j$ 
9:     Compute  $\hat{\ell}_{n+1}^f(\alpha^{n+1})$  approximating (13)
10:     $\alpha^{n+1} \leftarrow \text{Gradient Step}(\hat{\ell}_{n+1}^f(\alpha^{n+1}))$ 
11:   end while
12: end for
13: Output:  $(\alpha^{L+1}, \beta^L)$ 

```

文章在引入可行的 IPF 迭代算法后接着讨论了这个离散情形迭代算法的收敛性。文中指出在一定的前提下, 迭代得到的路径测度序列 $(\pi^n)_{n \in \mathbb{N}}$ 是良定的、相邻之间的 KL 散度递减, 其边缘分布与 p_{data} 和 p_{prior} 之间的 KL 散度之和为 $o(1/n)$ 。接着作者指出在同一假设条件下 SB 问题解的存在性和 IPF 算法的收敛性。文章最后考虑了连续情形的 IPF 算法并指出离散 IPF 为连续 IPF 之离散化这一关系。

2.1.5. 实验

2.1.5.1. Gauss 分布之间的 Schrödinger 桥

取 $p_{\text{prior}} = \mathcal{N}(-\alpha, \mathbf{I})$, $p_{\text{data}} = \mathcal{N}(\alpha, \mathbf{I})$, 其中 $\alpha = 0.1 \times \mathbf{1}$ 。这个问题的存在解析形式的静态 SB。取数据维数为 $d = 5, 50$, 并在其上训练 DSB, 结果如下。小网络 (small) 可以在低维数据的情况下求得正确结果, 但在高维情况下需要增加网络大小和训练轮数。

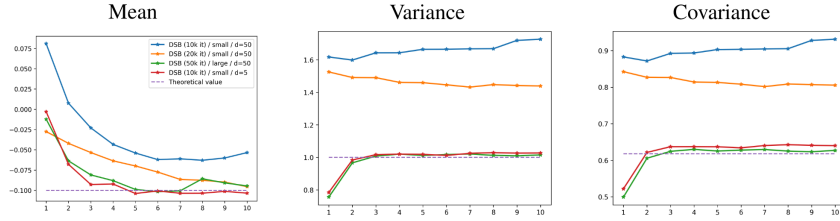


图 2 DSB 模型在 Gaussian 分布实验上收敛

2.1.5.2. 二维分布

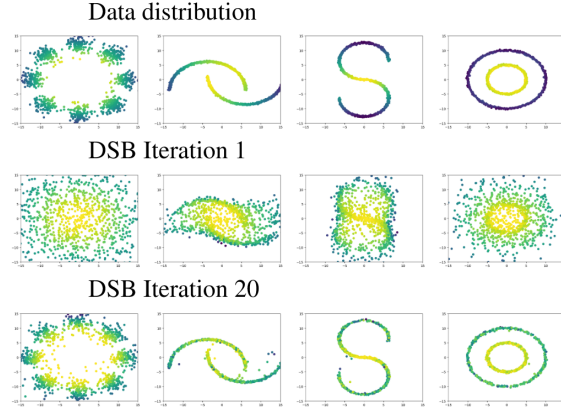


图 3 DSB 模型二维数据集上的训练结果

2.1.5.3. 图像生成

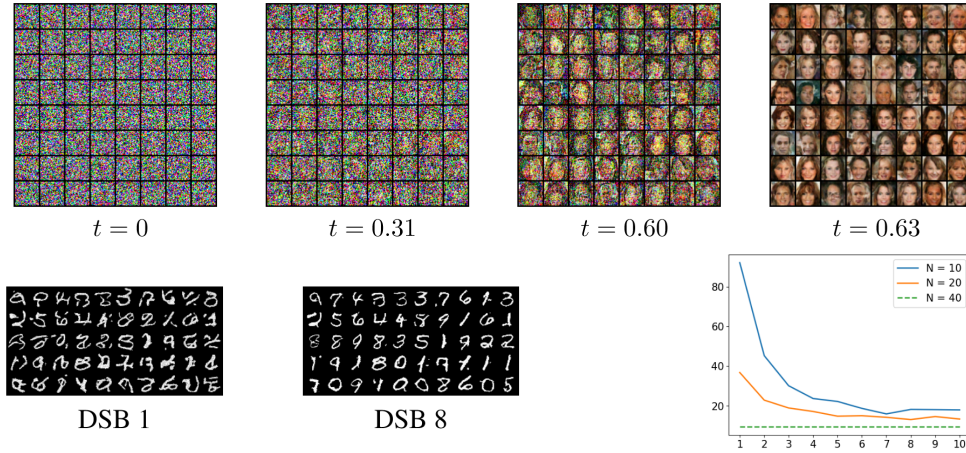


图 4 DSB 模型二维数据集上的训练结果

2.1.5.4. 数据插值

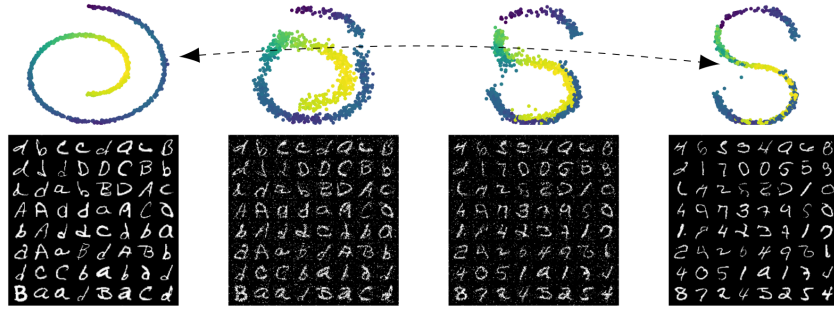


图 5 DSB 模型二维数据集上的训练结果

2.1.6. 讨论

本文首次将 Schrödinger 桥引入生成模型中，相比其他生成模型，SDB 可以以更少的步数达到较好的效果。此外，DSB 的解也是一个扩散过程，这意味着可以套用 Song 等人[2]文章中的 ODE 解法来实现更快地采样生成。尽管如此，如果将 DSB 的迭代轮数 N 降得太低，则可能因 p_N 与 p_{prior} 不再接近导致生成效果变差。本文在理论推导上进行了很多探索，保证了分数模型用于生成的可行性，以及使用 IPF 迭代算法的收敛性。此外，DSB 还可以用于多边缘分布约束的 Schrödinger 桥问题、计算 Wasserstein barycenter、求解熵正则化的 Gromov-Wasserstein 问题的最小值解以及求解连续状态空间中的域适应问题。

3. 学习进度

3.1. 实分析

3.1.1. 动机

第一个问题源于 Fourier 变换。

第二个问题是极限和积分的可交换性。

第三个问题是可求长曲线的问题。

第四个问题是

3.1.2. 方体

为了求 \mathbb{R}^p 中某些集合的“大小”或者“体积”，我们需要后者分解为可以轻易求得体积的“基本构件”的“几乎无交”并。而我们采用方体为基本构件。

定义 3.1 (\mathbb{R}^p 中的 (闭) 方体): \mathbb{R}^p 中的方体是一个这样的集合 R :

$$R = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_p, b_p]. \quad (23)$$

它的意思是

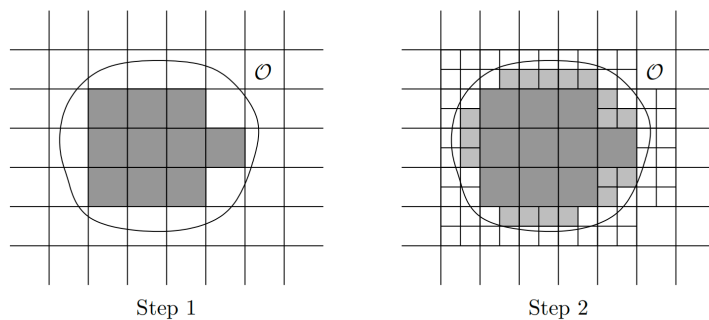
$$R = \{x \in \mathbb{R}^p : a_1 \leq x_1 \leq b_1, \dots, a_p \leq x_p \leq b_p\}. \quad (24)$$

它的体积为 $|R| = \prod_{i=1}^p (b_i - a_i)$.

相应地; 开方体只需将定义中的闭区间变成开区间即可, 且其体积与相同端点的闭方体相同。如果两个方体的内部不交, 我们称它们几乎无交。对于 \mathbb{R}^2 中的开集, 我们能得到十分有趣的结论: $\mathbb{R}^d (d \geq 1)$ 中的任意开集都可以写成可数个几乎无交的闭方体的并:

定理 3.1: \mathbb{R}^p 中的任意开集 \mathcal{O} 都可以写成可数个几乎无交个闭方体的并。

证明它的方法并不复杂。首先画一个边长为 1 的网格, 就得到了若干边长为 1 的方体。然后做下面的操作。(1) 如果某方体被集合 \mathcal{O} 包含, 那么我们接受该方体; (2) 如果某方体与集合 \mathcal{O} 不交, 我们拒绝之; (3) 如果该方体和集合 \mathcal{O} 的边界交集非空, 我们暂且接受。接下来将每个暂且接受的方体, 划分为大小相同、边长相等的四个小方体, 然后继续一直做上面的接受-拒绝测试, 然后对于暂且接受的方体一直划分下去。由于所有边长的方体可以和 $\mathbb{Z}^p \times \mathbb{Z}$ 形成一一对应, 因此全体方体的集合是可数的, 接收得到的方体的集合也是可数的。

图 6 将开集 \mathcal{O} 分解为可数个几乎无交的方体

3.1.3. Cantor 集

定义 3.2 (Cantor 集): 定义这样的一系列集合 $\{C_n\}_{n=1}^{\infty}$, 其中 $C_0 = [0, 1]$. $C_1 = \left[0, \frac{1}{3}\right] \cup \left[\frac{2}{3}, 1\right]$, 相当于将 C_0 中的闭区间每个切成三份, 弃去中间的一份。然后一直这样做下去, 得到 C_2, C_3 等等。Cantor 集 \mathcal{C} 定义为这些集合的交:

$$\mathcal{C} = \bigcap_{i=1}^{\infty} C_i \quad (25)$$

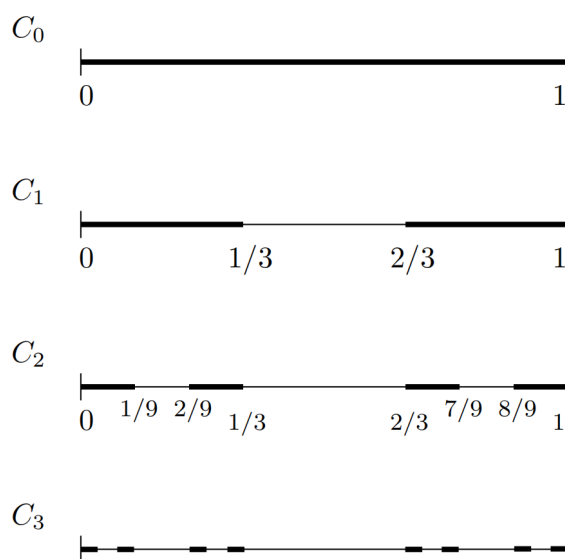


图 7 Cantor 集的构造

它有一些有趣的性质, 例如它的“长度”为零, 但它却是不可数集。

4. 下周计划

论文阅读

1. 生成模型
 - 薛定谔桥
 - DDIM

项目进度

1. 使用神经网络学习生命游戏的演化动力学
 - 考虑另外两种方法的实现
 - 更新在线 Overleaf 文档
2. 耦合约瑟夫森结
 - 将 MATLAB 模拟代码全部迁移至 Python
 - 考虑简单的 Neural SDE 方法解带参 OU 过程的参数

理论学习

1. 随机过程课程
 - 复习 Poisson 过程和 Markov 过程
2. 随机微分方程
 - 第五章完成

5. 附录

参考文献

- [1] V. D. Bortoli, J. Thornton, J. Heng, and A. Doucet, “Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling.” [Online]. Available: <https://arxiv.org/abs/2106.01357>
- [2] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-Based Generative Modeling through Stochastic Differential Equations,” *CoRR*, 2020, [Online]. Available: <https://arxiv.org/abs/2011.13456>