2025 年 9 月 8 日至 9 月 14 日周报

何瑞杰
中山大学, 大湾区大学

1. 文献阅读

1.1. Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song, Stefano Ermon | NeurIPS 2019 | https://arxiv.org/abs/1907.05600

1.1.1. Score matching

1.1.1.1. 生成模型和 Score matching 动机

生成模型的目的是获取所需要生成范畴中的对象（如图片）的隐藏分布。生成的过程就是从该分布中采样。假设有从一个未知分布 $p_{\text{data}}(\boldsymbol{x})$ 中采样得到的数据集 $\{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^n$。我们要尝试估计该分布。一个自然的假设是

$$p_{\text{data}}(\boldsymbol{x}) = \frac{\exp(-f_\theta(\boldsymbol{x}))}{Z(\theta)}$$

其中 $f_\theta : \mathbb{R}^D \to \mathbb{R}$ 是某个函数，$Z(\theta)$ 是归一化因子。若不加其他考虑，直接处理 $p_{\text{data}}(\boldsymbol{x})$ 将会不可避免地遇到计算 $Z(\theta)$ 的困难。因此 Score matching 的一个核心思路是转而去估计分布的 Score function，其“几何直观”的意义是指向概率密度增加的方向：

$$s_\theta(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x}) - \underbrace{\nabla_{\boldsymbol{x}} \log Z(\theta)}_{=0} = -\nabla_{\boldsymbol{x}} f_\theta(\boldsymbol{x}).$$

1.1.1.2. 以方便计算为目的的 Score matching 目标函数变换

自然地，我们有 Score matching 的原始目标函数：

$$J(\theta) := \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\|s_\theta(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\|].$$

但由于我们不知道原始数据的分布，因此我们无法求得 $\nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})$，可以做下面的变换

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|s_\theta(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\|] \\
=& \frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|s_\theta(\boldsymbol{x})\|] + \cancel{\mathbb{E}_{p_{\text{data}}}[\|\nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\|]} - \mathbb{E}_{p_{\text{data}}}[\langle s_\theta(\boldsymbol{x}), \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\rangle] \\
=& \frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|s_\theta(\boldsymbol{x})\|] + \int p(\boldsymbol{x})\langle s_\theta(\boldsymbol{x}), \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\rangle \mathrm{d}x + \text{constant} \\
=& \frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|s_\theta(\boldsymbol{x})\|] + \int \langle s_\theta(\boldsymbol{x}), \nabla_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x})\rangle \mathrm{d}x + \text{constant} \\
=& \frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|s_\theta(\boldsymbol{x})\|] + \left(\cancel{\int_{\partial\mathbb{R}^D} s_\theta(\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}) \mathrm{d}x} + \int_{\mathbb{R}^D} p_{\text{data}}(\boldsymbol{x}) \nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x})\right) + \text{constant} \\
=& \frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|s_\theta(\boldsymbol{x})\|] + \int p(\boldsymbol{x})\ \mathrm{div}(s_\theta(\boldsymbol{x}))\mathrm{d}x + \text{constant} \\
=& \mathbb{E}_{p_{\text{data}}}\left[\frac{1}{2}\|s_\theta(\boldsymbol{x})\| + \mathrm{tr}(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x}))\right]
\end{aligned}$$

1.1.1.3. 降低目标函数计算成本

上式中的 $\mathrm{tr}(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x}))$ 计算成本还是太高。庆幸我们可以对原数据增加 Gaussian 噪声，从而将其经过一个条件分布 $q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x}) \sim N(0, \sigma^2 I)$ 得到加噪声后的数据 $\bar{\boldsymbol{x}}$。经计算后我们能得到更加实际的目标函数。我们可以从扰动后的数据向量 $\bar{\boldsymbol{x}} \sim q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x})$ 对应的损失函数开始推导：

$$
\begin{aligned}
J(\theta) &= \frac{1}{2}\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}\left[\|s_\theta(\bar{\boldsymbol{x}}) - \nabla_{\bar{\boldsymbol{x}}} \log p_\sigma(\bar{\boldsymbol{x}})\|_2^2\right] \\
&= \frac{1}{2}\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}\left[\|s_\theta(\bar{\boldsymbol{x}})\|_2^2\right] + \cancel{\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}\left[\|\nabla_{\bar{\boldsymbol{x}}} \log p_{\mathrm{data}}(\bar{\boldsymbol{x}})\|_2^2\right]} - \mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}[\langle s_\theta(\bar{\boldsymbol{x}}), \nabla_{\bar{\boldsymbol{x}}} \log p_\sigma(\bar{\boldsymbol{x}})\rangle] \\
&= \frac{1}{2}\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}\left[\|s_\theta(\bar{\boldsymbol{x}})\|_2^2\right] - \int p_\sigma(\bar{\boldsymbol{x}})\langle s_\theta(\bar{\boldsymbol{x}}), \nabla_{\bar{\boldsymbol{x}}} \log p_\sigma(\bar{\boldsymbol{x}})\rangle \mathrm{d}\bar{\boldsymbol{x}} + \text{constant} \\
&= \frac{1}{2}\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}\left[\|s_\theta(\bar{\boldsymbol{x}})\|_2^2\right] - \int \langle s_\theta(\bar{\boldsymbol{x}}), \nabla_{\bar{\boldsymbol{x}}} p_\sigma(\bar{\boldsymbol{x}})\rangle \mathrm{d}\bar{\boldsymbol{x}} + \text{constant} \\
&= \frac{1}{2}\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}\left[\|s_\theta(\bar{\boldsymbol{x}})\|_2^2\right] - \int \langle s_\theta(\bar{\boldsymbol{x}}), \nabla_{\bar{\boldsymbol{x}}} \int q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x}) p_{\mathrm{data}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\rangle \mathrm{d}\bar{\boldsymbol{x}} + \text{constant} \\
&= \frac{1}{2}\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}})}\left[\|s_\theta(\bar{\boldsymbol{x}})\|_2^2\right] - \iint p_\sigma(\bar{\boldsymbol{x}}) p_{\mathrm{data}}(\boldsymbol{x}) \langle s_\theta(\bar{\boldsymbol{x}}), \nabla_{\bar{\boldsymbol{x}}} q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x})\rangle \mathrm{d}\boldsymbol{x} \mathrm{d}\bar{\boldsymbol{x}} + \text{constant} \\
&= \mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}}), \boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}\left[\frac{1}{2}\|s_\theta(\bar{\boldsymbol{x}})\|_2^2 - \langle s_\theta(\bar{\boldsymbol{x}}), \nabla_{\bar{\boldsymbol{x}}} q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x})\rangle\right] + \text{constant} \\
&= \frac{1}{2}\mathbb{E}_{\bar{\boldsymbol{x}}\sim p_\sigma(\bar{\boldsymbol{x}}), \boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}\left[\|s_\theta(\bar{\boldsymbol{x}})\|_2^2 - 2\langle s_\theta(\bar{\boldsymbol{x}}), \nabla_{\bar{\boldsymbol{x}}} q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x})\rangle + \underbrace{\|\nabla_{\bar{\boldsymbol{x}}} q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x})\|_2^2}_{\text{constant w.r.t. } \theta}\right] + \text{constant} \\
&= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}, }\left[\|s_\theta(\bar{\boldsymbol{x}}) - \nabla_{\bar{\boldsymbol{x}}} \log q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x})\|_2^2\right]
\end{aligned}
$$

注意此时 $\nabla_{\bar{\boldsymbol{x}}} \log q_\sigma(\bar{\boldsymbol{x}}|\boldsymbol{x})$ 还可以写为 $-\dfrac{1}{\sigma}\cdot\varepsilon$，其中 ε 是从 $N(0, I)$ 中采样得到的噪声，原目标函数就变成

$$
J(\theta) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}, }\left[\left\|s_\theta(\bar{\boldsymbol{x}}) + \frac{1}{\sigma}\cdot\varepsilon\right\|_2^2\right]
$$

换句话说，score function 在这个意义下正在预测加入到训练数据中的噪声。在实际操作中，我们需要权衡 ε 的取值，如果太小，该方法起不到明显效果；如果太大，加噪声后的分布和原分布的区别大，难以学习原分布的特征。

1.1.2. Langevin 动力学

假如我们已经训练好 $s_\theta(\boldsymbol{x})$，我们应该如何做“生成”这个动作呢？根据 score function 拟合概率的对数梯度 $\nabla_{\boldsymbol{x}} \log p_{\mathrm{data}}(\boldsymbol{x})$，它指向概率密度上升的方向。我们可以按照自然地想法做梯度上升，这样迭代就可以得到很有可能属于该分布的点：

$$
\boldsymbol{x}_{\{t+1\}} \leftarrow \boldsymbol{x}_t - \varepsilon s_\theta(\boldsymbol{x}_t)
$$

但这一方法总会使得若干步后的采样结果收敛于原始分布之密度函数的若干极大值点，与生成模型的多样性目标不符。因此为解决这一问题，我们可以用 Langevin Dynamics 来采样。其与原来方法的区别在于在每一步中加入噪声，最后迭代得到的结果将会服从原始分布 $p_{\mathrm{data}}(\boldsymbol{x})$：

$$
\boldsymbol{x}_{\{t+1\}} \leftarrow \boldsymbol{x}_t - \frac{\varepsilon}{2} s_\theta(\boldsymbol{x}_t) + \sqrt{\varepsilon} z_t
$$

其中 $z_t \sim N(0, I)$。

1.1.3. Score-based 生成模型的问题

1. 一是流形假设造成的问题。我们所处世界中的高维数据往往分布在一个低维流形上。但上文中提到的对数据分布密度函数在低维流形的环绕空间 $\mathbb{R}^D$ 中求梯度是没有意义的。
2. 二是低概率密度区域的估计问题。如果原分布是一个混合分布，且两个“峰”中间存在一个低概率密度的区域，模型学习时将难以获取该区域的信息，最后训练的结果在该区域的表现将会很差。

如果我们使用 Gauss 分布对原分布做扰动，则得到的新分布的支持集将会是整个环绕空间，而不是流形。另一方面，恰当强度的扰动也会使得低概率密度区域的概率密度增加，从而更容易采样到该区域中的点，为模型训练提供更多的信息。

1.1.4. 方法

上文中提到，对数据做扰动时，大强度的扰动会使得训练变得简单，但扰动后的分布与原分布相差很大；小强度的扰动使得扰动后分布近似原分布，但会有诸如低概率密度区域训练点不足的问题。

文中提出一个整合两者有点的方法，即不考虑单个扰动强度 σ，而是考虑一个序列 $\{\sigma_i\}_{i=1}^n$ 其中 σ_n 是一个足够小的数 (例如 0.01)，σ_1 是一个足够大的数 (例如 25)。我们训练一个条件 score function $s_\theta(\boldsymbol{x}, \sigma)$ 预测不同扰动强度下的噪声方向。此时目标函数变为

$$\ell(\theta, \sigma) := \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}), \bar{\boldsymbol{x}} \sim N(\boldsymbol{x}, \sigma^2 I)}\left[\left\| s_\theta(\bar{\boldsymbol{x}}) + \frac{\bar{\boldsymbol{x}} - \boldsymbol{x}}{\sigma^2} \right\|_2^2\right]$$

$$\ell(\theta, \{\sigma_i\}_{i=1}^n) := \frac{1}{n}\sum_{i=1}^n \lambda(\sigma_i)\ell(\theta, \sigma_i)$$

其中 $\lambda(\sigma_i)$ 是权重函数，常取为 $\lambda(\sigma_i) = \sigma_i^2$，以平衡不同扰动强度下 score function 的范数。在采样时，我们就做类似模拟退火的采样动作。首先选取最高的扰动强度 σ_1 然后再该噪声强度下迭代若干次，然后选取次高的扰动强度 σ_2 再该噪声强度下迭代若干次，以此类推。这样就可以综合利用大扰动强度和小扰动强度的有点，从而更好的采样。在实际实验中，作者提出的方法在 CIFAR-10 数据集上取得了不错的效果。整个过程如下面的算法所示。

Algorithm 1: Anneled Langevin Dynamics

procedure ANNELED LANGEVIN DYNAMICS($\{\sigma_i\}_{i=1}^n$, ε, T)
  ▷ Initialize the search range
  $\bar{\boldsymbol{x}}_0 \leftarrow \boldsymbol{v}$
  for $t \leftarrow 1, ..., L$ **do**
    ▷ Set step size α_i
    $\alpha_i \leftarrow \varepsilon \cdot \sigma_i^2 / \sigma_L^2$
    for $t \leftarrow 1, ..., T$ **do**
      Draw $\boldsymbol{z}_t \sim N(0, I)$
      $\bar{\boldsymbol{x}}_t \leftarrow \bar{\boldsymbol{x}}_{t-1} + \alpha_i/2 \cdot s_\theta(\bar{\boldsymbol{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\boldsymbol{z}_t$
    end
    $\bar{\boldsymbol{x}}_0 \leftarrow \bar{\boldsymbol{x}}_T$
  end
  return $\bar{\boldsymbol{x}}_T$
end

2. 项目进展

2.1. 使用神经网络学习生命游戏的演化动力学

3. 学习进度

3.1. 随机过程

进行了概率论部分的简单回顾。

3.2. 随机微分方程

进度推进至 Itô 积分的定义，它的思路比较长。

首先有一个 Paley-Wiener-Zygmund 随机积分定义，它需要要求被积函数 g 是一个确定的函数，无法满足 $\int_0^t \boldsymbol{B}(\boldsymbol{X}, s)\mathrm{d}\boldsymbol{W}$ 这样的情形。因此我们考虑从 Riemann 和的角度逐步推广。

首先对于一维 Brown 运动 W 和区间 $[0,T]$ 上的一个划分 P，我们可以定义 $\int_0^T W\mathrm{d}W$ 的 Riemann 和估计

$$R = R(P,\lambda) = \sum_{k=0}^{m-1} W(\tau_k)[W(t_{k+1}) - W(t_k)]$$

接着我们就研究当 P 的细度趋于零时这个 Riemann 和是否收敛。我们首先证明了一维 Brown 运动在 $[a,b]$ 的二次变差在 $L^2(\Omega)$ 中趋于 $b-a$ —— 这也侧面说明其在任意该区间上的变差几乎必然无限 —— 然后我们可以用此结果证明上面的 Riemann 和估计在划分变细时有极限

$$\lim_{|P|\to 0} R(P,\lambda) = \frac{W(T)^2}{2} + \left(\lambda + \frac{1}{2}\right)T$$

看似自然的取法是令 $\lambda = \dfrac{1}{2}$ 这将得到 Stranovich 积分。而 Itô 积分的取法是 $\lambda = 0$，也就是划分小区间的中点取得是小区间的左端点，这将在后续的处理中带来便利。

接着我们开始研究可以作为被积的随机过程。我们考虑的是在 $[0,T]$ 上二次可积的循序可测随机过程空间 $\mathbb{L}^2(0,T)$。和 Lebesgue 积分的定义类似，我们先从“简单”的循序可测过程开始，也就是阶梯过程。类似阶梯函数，阶梯过程 $G \in \mathbb{L}^2(0,T)$ 是这样的随机过程：存在 $[0,T]$ 上的一个划分 $P = \{0 = t_0 < t_1 < \cdots < t_m = T\}$，使得对任意 $t \in [t_k, t_{k+1})$，都有 $G(t) \equiv G(t_k) = G_k$。其中 $G(t_k)$ 是 $\mathcal{F}(t_k)$-可测的。有了阶梯过程的定义，我们容易给出其随机积分的形式

$$\int_0^T G\mathrm{d}W \coloneqq \sum_{k=0}^{m-1} G_k[W(t_{k+1}) - W(t_k)]$$

接着，任意 $G \in \mathbb{L}^2(0,T)$ 都可以被有界阶梯过程逼近，从而可以形成 Itô 积分的良好定义。

3.3. 流形上的微积分