拡散モデルと確率微分方程式

榎本 拓実

August 4, 2025

目標

- 拡散モデル及びそれと SDE との関係を速習する
- 拡散モデルへの Malliavin 解析の応用例を理解する
(次回以降...)

生成 AI とは

拡散モデル: 生成モデル (生成 AI) の 1 つ

生成 AI でのタスク

1. 実データ $D = \{x_0, x_1, \cdots, x_n\} \subset \mathbb{R}$ からその発生分布 $p(x)$ を学習
2. その分布から効率的にサンプリング

統計学と何が違う??

- データ空間 X が巨大 (ex: MNIST データでは 28×28 次元)
- 想定される分布の形状も非常に複雑
- 分布の台が非常に小さい (多様体仮説)

生成モデルの例

以下が基本モデル. ここから様々な改善が提案されている.

- Variational Auto Encoder (VAE)
- Genetative Adversal Network (GAN)
- Transformer
- **拡散モデル**

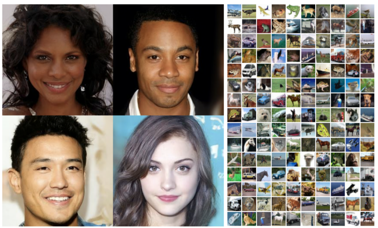

Figure 1: 生成された画像例[1]

[1]出典: [2] Fig1

拡散モデルとは

- データからノイズに変換する (拡散過程, 推論過程)
- その過程を学習することで, ノイズ入力からノイズ除去 (逆拡散過程, 生成過程)

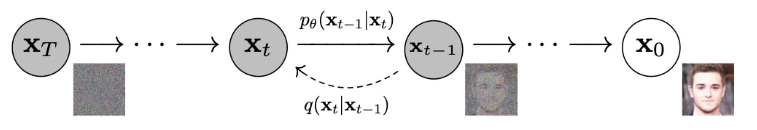

Figure 2: 拡散・逆拡散プロセス[2]

[2]出典: [2] Fig2

拡散モデル速習: DDPM を通じて

Denoising Diffusion Probabilistic Model ([2], NeurIPS2020)

拡散過程: 以下のような潜在変数モデル

元データ x_0 にノイズを徐々に加えていく

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}),$$
$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t I)$$

$0 < \beta_1 < \beta_2 < \cdots < \beta_T < 1$: 分散の大きさを制御

$\alpha_t = 1 - \beta_t$

$q(x_T|x_0) \approx \mathcal{N}(x_T; 0, I)$ なので, $q(x_T) \approx \mathcal{N}(x_T; 0, I)$

拡散過程により, 最終的には完全なノイズになる

逆拡散過程

デノイジングのプロセスをモデル化

各ステップでのガウス分布の平均・分散を NN で表現する

$$
\begin{aligned}
p_\theta(x_{0:T}) &= p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t),\\
p_\theta(x_{t-1}|x_t) &= \mathcal{N}(x_t; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)),\\
p(x_T) &= \mathcal{N}(x_T; 0, I)
\end{aligned}
$$

Remark.

逆拡散過程は必ずしも正規分布ではないが, $\beta_t \ll 1$ であれば正規分布で近似できる

(後述の SDE 表現に対する逆拡散過程 SDE を見よ)

ELBO による学習

パラメータ θ を最尤推定で求めたい

... が, 尤度 $p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$ の現実的には計算困難

ELBO (Evidence Lower BOunds) の利用: 最大化したい尤度の下限

$$\begin{aligned} \log p_\theta(x_0) &= E_{r(z)}\left(\log \frac{p_\theta(x_0, z)}{r(z)}\right) + D_{KL}(r(z) \| p(z|x)) \\ &\geq E_{r(z)}\left(\log \frac{p_\theta(x_0, z)}{r(z)}\right) =: \mathrm{ELBO(x_0)} \end{aligned}$$

新たな分布 r は任意.

ELBO のメリット: "sum-log"(連続なので int-log?) の形なので解析しやすい

DDPM での評価関数

$q = q(x_{1:T|x_0})$ と略記

$$
\begin{aligned}
L(\theta, x_0) &= \mathrm{ELBO}(x_0) \\
&= E_q\left(\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right) \\
&= E_q\left(\log \frac{p(x_T)}{q(x_T|x_0)} + \sum_{t>1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} + \log p_\theta(x_0|x_1)\right) \\
&= C + E_q\left(\sum_{t>1}^{T} D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) + \log p_\theta(x_0|x_1)\right)
\end{aligned}
$$

D_{KL} 部分は正規分布/正規分布なので解析的に計算できる

拡散モデルと SDE

拡散モデル = 拡散による推論 + 逆拡散による生成

拡散を SDE を使ってモデル化 [6]

$$dx_t = f(x_t, t)dt + G(x_t, t)dw_t$$

逆拡散過程を別の SDE から与えることができる [1]

Theorem
f, G が十分滑らか, かつ分布 $p_t(x)$ も十分滑らかと仮定. このとき,

$$\begin{aligned} dx_\tau = &\Big\{ f(x, \tau) - \nabla_x \cdot [G(x_\tau, \tau)G(x_\tau, \tau)^\top] \\ &- G(x_\tau, \tau)G(x_\tau, \tau)^\top \nabla_x \log p_t(x_\tau) \Big\} d\tau + G(x_\tau, \tau)d\bar{w}_\tau \end{aligned}$$

ただし, $\tau = T - t$, $\bar{w}_t$ は時刻逆向きの標準 *Wiener* 過程.

例: DDPM を連続化

$x_i = \sqrt{1-\beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1}$

$\hat{\beta}_i = \beta_i \Delta t$ とスケーリング. $\lim_{\Delta t \to 0} \hat{\beta}_i = \beta(t)$ とする.

更新式で極限を取れば,

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

より

$$
dx_t = -\frac{1}{2}\beta(t) x_t dt + \sqrt{\beta(t)} dw_t
$$

スコアマッチング

分布関数 $p(x) = \exp(-f(x))/Z$ と表記してみると, p が大きい $\Leftrightarrow$ f が小さい $\rightsquigarrow$ f は" エネルギー"

(統計力学からのアナロジー. Z は分配関数)

エネルギーを学習するには, Z の計算が困難 (大域的な量)

更に微分をとる: スコア $s(x) = \nabla_x \log p(x)$ (局所的な量)

スコアが学習できれば, Langevin MC でサンプリング可能

$\rightsquigarrow$ スコアを NN でモデリング = スコアマッチング, Score based

明示的スコアマッチング:

$$J_{\mathrm{ESM}}(p, \theta) = \frac{1}{2} E_{p(x)}(\|\nabla_x \log p(x) - s_\theta(x)\|^2)$$

デノイジングスコアマッチング

多くのケースでは $\log p(x)$ が計算できない $\rightsquigarrow$ ノイズで摂動する

データ x にノイズ $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ を加える: $\tilde{x} = x + \varepsilon$

摂動後の分布:

$$p_\sigma(\tilde{x}) = \int p_\sigma(\tilde{x}|x) p(x) dx$$

デノイジングスコア:

$$J_{\mathrm{ESM}}(p_\sigma, \theta) = \frac{1}{2} E_{p_\sigma(\tilde{x})}(\|\nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) - s_\theta(\tilde{x})\|^2)$$
$$J_{\mathrm{DSM}}(p_\sigma, \theta) = \frac{1}{2} E_{p_\sigma(\tilde{x})}(\|\nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) - s_\theta(\tilde{x})\|^2)$$

$J_{\mathrm{ESM}}(p_\sigma, \theta)$ と $J_{\mathrm{DSM}}(p_\sigma, \theta)$ は定数差なので, J_{DSM} で効率よく学習できる

スコアベース SDE 拡散モデル

サンプリングには逆拡散過程を用いる

$$dx_\tau = \Big\{ f(x,\tau) - \nabla_x \cdot [G(x_\tau,\tau)G(x_\tau,\tau)^\top] \\ - G(x_\tau,\tau)G(x_\tau,\tau)^\top \nabla_x \log p_t(x_\tau) \Big\} d\tau + G(x_\tau,\tau) d\bar{w}_\tau$$

$\rightsquigarrow$ スコアが学習できれば良い

評価関数

$$J(\theta) = E_{\lambda(t)} E_{p_{\mathrm{data}}(x_0)} E_{p(x_t|x_0)} (\|\nabla_{x_t} \log p(x_t, t|x_0, 0) - s_\theta(x_t, t)\|^2)$$

Remark.

s_θ の引数に t を含めることで, 各時刻での NN を連続濃度用意するのではなく時刻も含めて 1 つだけの NN で表現

SDE 化のご利益

- ロバスト性等の理論解析につながる
    - [5] (SDE 関連ではないが)Attention 機構での Lipshitz 性を議論
- SDE のアイデアを活用
    - [3] Malliavin 微分により特異な評価関数を利用可能に
- 不均一サンプリングを自然に実現 (SDE というより連続化のご利益)
    - [4] Neural ODE 等

[1] B. D. O. Anderson.
Reverse-time diffusion equation models.
Stochastic Processes and their Applications, 12(3):313–326, 1982.

[2] J. Ho et al.
Denoising diffusion probabilistic models.
In *Proc. NeurIPS*, 2020.

[3] J. Pidstrigach et al.
Conditioning diffusions using malliavin calculus.
In *Proc. ICML*, 2025.

[4] R. T. Q. Chen et al.
Neural ordinary differential equations.
In *Proc. NeurIPS*, 2018.

[5] V. Castin et al.
How smooth is attention?
In *Proc. ICML*, 2024.

[6] Y. Song et al.
Score-based generative modeling through stochastic differential equations.
In *Proc. ICLR*, 2021.