Deep Dive: Understanding diffusion model from a mathematical view

Pipi Hu

Microsoft Research AI4Science

May 20, 2025

Feynman's words

What I cannot create, I do not understand.

Microsoft Research AI4Science

Outline

Microsoft Research AI4Science

Generate samples form given distribution

Given a distribution $p(x)$, how to sample from the distribution?

- Uniform distribution & Gaussian distribution: easy sampling
- Mixture Gaussian: determine which Gaussian to sample & sample from that Gaussian
- A general sample function?
  - Langevin Markov Chain Monte Carlo sampling (Langevin MCMC)
  - Stein Variational Gradient Descent (SVGD)
  - ...

Microsoft Research AI4Science

Score function is all you need?

A typical process of generating new samples from modeling the data distribution

1. To find the probability function $p(x)$, model $p(x;\theta)$.
2. Normalization distribution $p(x;\theta) = \frac{1}{C(\theta)} e^{-E(x;\theta)}$ with $C(\theta) \equiv \int_x e^{-E(x;\theta)} dx$.
3. Generating samples from above sampling methods.

Recall the Langevin Markov Chain Monte Carlo sampling

$$x_t = x_{t-1} + \frac{\Delta t}{2} \nabla_x \log p(x_{t-1};\theta) + \sqrt{\Delta t}\epsilon, \quad \epsilon \sim N(0,1). \tag{1}$$

Here $\nabla_x \log p(x_{t-1};\theta) = -\nabla_x E(x;\theta)$ is free of the normalization $C(\theta)$. Define the score function

$$S(x;\theta) \equiv \nabla \log p(x;\theta). \tag{2}$$

Microsoft Research AI4Science

More about Langevin MCMC and the score function

Continuous form of the Langevin MCMC

- $\Delta t \to 0$
$$dx = \frac{1}{2}\nabla_x \log p(x) dt + dB_t, \tag{3}$$
where dB_t is a Brownian motion.
- Given $p(x) = \frac{1}{C}e^{-E}$, we have
$$dx = -\frac{1}{2}\nabla_x E dt + dB_t \tag{4}$$

Langevin MCMC is overdamped Langevin dynamics

- Langevin dynamics (MD)
$$\ddot{x} = -\nabla_x E - \gamma\dot{x} + dB_t, \tag{5}$$
- Given $\gamma\dot{x} \gg \ddot{x}$, we have the overdamped Langevin equation
$$dx = -\frac{1}{\gamma}\nabla_x E + \frac{1}{\gamma}dB_t. \tag{6}$$

Microsoft Research AI4Science

Outline

Microsoft Research AI4Science

Matching the score by neural networks

Recall the definition

$$S \equiv \nabla_x \log p(x), \tag{7}$$

The key is to matching the score by a neural network

$$J_{ESM} = \frac{1}{2}\mathbb{E}_p \left[\| s(x;\theta) - \frac{\partial \log p(x)}{\partial x} \|^2 \right], \tag{8}$$

where J_{ESM} is called Explicit Score Matching loss. However, it is not tractable to compute $\frac{\partial \log p(x)}{\partial x}$ from data.

Where should we go?

Microsoft Research AI4Science

Implicit Score Matching (ISM) loss

Implicit Score Matching loss (Aapo, 2005) was developed to make a tractable score matching

$$J_{ISM} \equiv \mathbb{E}_p \left[\frac{1}{2} \| s(x;\theta) \|^2 + \nabla \cdot s(x;\theta) \right] = J_{ESM} - C. \tag{9}$$

Proof.

$$\begin{aligned} J_{ESM} &= \frac{1}{2} \mathbb{E}_p \left[\| s(x;\theta) - \frac{\partial \log p(x)}{\partial x} \|^2 \right] \\ &= \frac{1}{2} \mathbb{E}_p \left[\| s \|^2 \right] - \mathbb{E}_p [s \cdot \frac{\partial \log p}{\partial x}] + \frac{1}{2} \mathbb{E}_p \left[\| \frac{\partial \log p}{\partial x} \|^2 \right] \end{aligned} \tag{10}$$

where

$$\begin{aligned} & - \mathbb{E}_p [s \cdot \frac{\partial \log p}{\partial x}] = - \int_x ps \cdot \nabla \log p dx = - \int_x ps \cdot \frac{\nabla p}{p} dx \\ & = - \int_x s \cdot \nabla p dx = - \int \nabla \cdot (ps) dx + \int p \nabla \cdot s dx = \mathbb{E}_p [\nabla \cdot s]. \end{aligned} \tag{11}$$

So we have $J_{ISM} = J_{ESM} - C$, where $C \equiv \frac{1}{2} \mathbb{E}_p \left[\| \frac{\partial \log p}{\partial x} \|^2 \right]$. □

Denoising Score Matching (DSM) loss

However, the ISM score is not very stable to optimize, with two reasons

1. Expectation is performed on the whole distribution;
2. The loss is negative decreasing to $-C$ with a great quantity, e.g. $-1e5$.

Denoising Score Matching (DSM) loss (Vincent, 2011) is developed to solve this problem

$$J_{DSM} = \frac{1}{2}\mathbb{E}_{p(x,\tilde{x})}\left[\|s(\tilde{x};\theta) - \nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2\right]. \tag{12}$$

And we can prove that $J_{DSM} = J_{ESM} + C$.

Microsoft
Research
AI4Science

Prove the equivalence $J_{DSM} = J_{ESM} + C$

Proof.

We expand the formula and check the items one by one.

$$\begin{aligned} J_{DSM} &\equiv \frac{1}{2}\mathbb{E}_{p(x,\tilde{x})}\left[\|s(\tilde{x};\theta) - \nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2\right] \\ &= \frac{1}{2}\mathbb{E}_{p(\tilde{x})}\left[\|s(\tilde{x})\|^2\right] - \mathbb{E}_{p(x,\tilde{x})}[s(\tilde{x}) \cdot \nabla_{\tilde{x}} \log p(\tilde{x}|x)] + \frac{1}{2}\mathbb{E}_{p(x,\tilde{x})}[\|\nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2] \end{aligned} \tag{13}$$

And we can prove that the cross term of J_{ESM} and J_{DSM} is equal.

$$\begin{aligned} &\mathbb{E}_p(\tilde{x})[s(\tilde{x}) \cdot \nabla_{\tilde{x}} \log p(\tilde{x})] = \int_{\tilde{x}} p(\tilde{x})s(\tilde{x}) \cdot \nabla_{\tilde{x}} \log p(\tilde{x})d\tilde{x} \\ &= \int_{\tilde{x}} s(\tilde{x}) \cdot \nabla_{\tilde{x}} p(\tilde{x})d\tilde{x} = \int_{\tilde{x}}\int_x p(x)s(\tilde{x}) \cdot \nabla_{\tilde{x}} p(\tilde{x}|x)d\tilde{x}dx \\ &= \int_{\tilde{x}}\int_x p(\tilde{x}|x)p(x)s(\tilde{x}) \cdot \nabla_{\tilde{x}} \log p(\tilde{x}|x)d\tilde{x}dx = \mathbb{E}_{p(\tilde{x},x)}\left[s(\tilde{x}) \cdot \nabla_{\tilde{x}} \log p(\tilde{x}|x)\right]. \end{aligned} \tag{14}$$

Finally we have

$$J_{DSM} = J_{ESM} - \frac{1}{2}\mathbb{E}_p\left[\|\nabla_{\tilde{x}} \log p(\tilde{x})\|^2\right] + \frac{1}{2}\mathbb{E}_{p(x,\tilde{x})}[\|\nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2]. \tag{15}$$

□

DSM is all you need?

Problem arises when $\tilde{x} \to x$. We can prove that when Gaussian noised added by $p(\tilde{x}|x) = N(\alpha x, \sigma^2)$,

$$J_{DSM} \to \infty, \text{ if } \tilde{x} \to x. \tag{16}$$

Proof.

Given the Gaussian noise added, we have

$$\nabla_{\tilde{x}} \log p(\tilde{x}|x) = \nabla_{\tilde{x}} \log e^{-\frac{(\tilde{x}-\alpha x)^2}{2\sigma^2}} = -\frac{\tilde{x} - \alpha x}{\sigma^2}. \tag{17}$$

And we can show that the following formula goes to infinity

$$\begin{aligned} &\mathbb{E}_{p(x,\tilde{x})}[\|\nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2] = \mathbb{E}_{p(x,\tilde{x})}[\|\frac{\tilde{x} - \alpha x}{\sigma^2}\|^2] \\ &= \mathbb{E}_{p(x)}\mathbb{E}_{p(\tilde{x}|x)}[\|\frac{\tilde{x} - \alpha x}{\sigma^2}\|^2] = \mathbb{E}_{\epsilon \sim N(0,1)}[\|\frac{\epsilon}{\sigma}\|^2] \to \infty \text{ if } \sigma \to 0. \end{aligned} \tag{18}$$

Finally, we know that

$$J_{DSM} = J_{ESM} + C \to \infty \text{ as } C \to \infty \text{ when } \tilde{x} \to x. \tag{19}$$

□

Outline

Microsoft Research AI4Science

Revisit the adding noise from x to $\tilde{x}$

Gaussian transition kernel for adding noise from x to $\tilde{x}$

$$\tilde{x} = \alpha x + \sigma \epsilon, \tag{20}$$

which equivalent to the Gaussian transition kernel

$$p(\tilde{x}|x) = N(\tilde{x}; \alpha x, \sigma^2), \tag{21}$$

where $\epsilon \sim N(0, 1)$ and

$$N(\tilde{x}; \alpha x, \sigma^2) = Ce^{-\frac{\|\tilde{x}-\alpha x\|^2}{2\sigma^2}}. \tag{22}$$

And

$$\epsilon = \frac{\tilde{x} - \alpha x}{\sigma}. \tag{23}$$

Questions arise: how can we get x given $\tilde{x}$?

Microsoft
Research
AI4Science

Tweedie's Formula

Tweedie's Formula (Bayes statistics) states (Robbins, 1956)

Theorem

Given random variables x, y, $y \sim N(x, \sigma^2 I)$, i.e., $p(y|x) = N(x, \sigma^2)$, the expectation of x could be given by

$$\mathbb{E}[x|y] = y + \sigma^2 \nabla \log p(y). \tag{24}$$

Let

$$x \leftarrow \alpha x, \quad y \leftarrow \tilde{x}, \tag{25}$$

we have

$$s \equiv \nabla_{\tilde{x}} \log p(\tilde{x}) = -\frac{\tilde{x} - \alpha \mathbb{E}[x|\tilde{x}]}{\sigma^2} = -\frac{1}{\sigma}\mathbb{E}[\frac{\tilde{x} - \alpha x}{\sigma}|\tilde{x}] = -\frac{\mathbb{E}[\epsilon|\tilde{x}]}{\sigma}, \tag{26}$$

where we have used the fact $\epsilon = \frac{\tilde{x} - \alpha x}{\sigma}$.

Microsoft Research AI4Science

Another insight: from the definition of the score S

Recall that adding noise by $\tilde{x} = \alpha x + \sigma\epsilon$,

$$p(\tilde{x}|x) = Ce^{-\frac{-\|\tilde{x}-\alpha x\|^2}{2\sigma^2}}. \tag{27}$$

Hence

$$s(\tilde{x}) = \nabla_{\tilde{x}} \log p(\tilde{x}) = \frac{\nabla_{\tilde{x}} p(\tilde{x})}{p(\tilde{x})} = \frac{\int_x \nabla_{\tilde{x}} p(\tilde{x}|x) p(x) dx}{p(\tilde{x})}, \tag{28}$$

From Eq. (27), we have

$$s(\tilde{x}) = \frac{\int_x \nabla_{\tilde{x}} p(\tilde{x}|x) p(\tilde{x}|x) p(x) dx}{p(\tilde{x})} = \int_x \nabla_{\tilde{x}} \log p(\tilde{x}|x) p(x|\tilde{x}) dx, \tag{29}$$

leading to

$$s(\tilde{x}) = \mathbb{E}_x[\nabla_{\tilde{x}} \log p(\tilde{x}|x)|\tilde{x}] = \mathbb{E}_x[-\frac{\tilde{x} - \alpha x}{\sigma^2}|\tilde{x}] = -\frac{\tilde{x} - \alpha\mathbb{E}[x|\tilde{x}]}{\sigma^2}. \tag{30}$$

Microsoft Research AI4Science

Theoretical Support

Theorem

Let X be an integrable random variable. Then for each σ-algebra $\mathcal{V}$ and $Y \in \mathcal{V}$, $Z = \mathbb{E}(X|\mathcal{V})$ solves the least square problem

$$\|Z - X\| = \min_{Y \in \mathcal{V}} \|Y - X\|,$$

where $\|Y\| = \left(\int Y^2 dP\right)^{\frac{1}{2}}$.

Lemma

If Y is $\mathcal{V}$-measurable, and f is a measurable function in the sense that its domain and codomain are appropriately aligned with the σ-algebras, then $f(Y)$ will also be $\mathcal{V}$-measurable

By Theorem 2, the score matching loss can be written as

$$Loss_{score} = \mathbb{E}_t \mathbb{E}_{x_t \sim p(x_t|x_0)} \mathbb{E}_{x_0} \|S_\theta(x_t, t) - \nabla_{\tilde{x}} \log p(\tilde{x}|x)\|^2. \quad (31)$$

Microsoft Research AI4Science

Revisit several models

1. Score model

$$Loss_{score} = \|s_\theta(\tilde{x}) - \frac{\epsilon}{\sigma}\|^2; \tag{32}$$

2. X prediction model (denoising model, $x0$ model)

$$Loss_{x_0} = \|x_\theta(\tilde{x}) - x\|^2, \tag{33}$$

with

$$s(\tilde{x}) = -\frac{\tilde{x} - \alpha x_{\theta^*}(\tilde{x})}{\sigma^2} \tag{34}$$

where $x_\theta(\tilde{x}) \to x_{\theta^*}(\tilde{x}) \equiv \mathbb{E}[x|\tilde{x}]$;

3. ϵ prediction model

$$Loss_\epsilon = \|\epsilon_\theta(\tilde{x}) - \epsilon\|^2, \tag{35}$$

with

$$s(\tilde{x}) = -\frac{\epsilon_{\theta^*}(\tilde{x})}{\sigma}, \tag{36}$$

where $\epsilon_\theta(\tilde{x}) \to \epsilon_{\theta^*}(\tilde{x}) \equiv \mathbb{E}[\epsilon|\tilde{x}]$.

Microsoft Research AI4Science

The consistency of the three modeling

As stated before, we have the following connection in the optimal (θ^*) case

$$s_\theta^*(\tilde{x}) = -\frac{\tilde{x} - \alpha x_{\theta^*}(\tilde{x})}{\sigma^2} = -\frac{\epsilon_{\theta^*}(\tilde{x})}{\sigma}, \tag{37}$$

by the above conditional expectation.
Hence, in the real modeling of designing the loss, we use the connection of three models without reaching the optimal parameter

$$s_\theta(\tilde{x}) = -\frac{\tilde{x} - \alpha x_\theta(\tilde{x})}{\sigma^2} = -\frac{\epsilon_\theta(\tilde{x})}{\sigma}, \tag{38}$$

and substitute each other form to the loss definition above, we can recover all of the losses given by different models.

Microsoft
Research
AI4Science

Outline

Microsoft Research AI4Science

SMLD

Score Matching Langevin Dynamic (SMLD) (Song, 2019) adding noise in a following manner

$$p(\tilde{x}_i|x) = N(\tilde{x}_i; x, \sigma_i^2) \equiv Ce^{-\frac{\|\tilde{x}_i - x\|^2}{2\sigma_i^2}}, \tag{39}$$

where a geometric sequence $\sigma_1 > \sigma_2 > \cdots > \sigma_T \approx 0$ is given to add noise with different level.

In a random variable view,

$$\tilde{x}_i = x + \sigma_i \epsilon \tag{40}$$

for different σ_i to get different noised random variable $\tilde{x}_i$.

Training object J_{DSM} is given by

$$J_{DSM} = \frac{1}{2}\mathbb{E}_{\sigma_i}\mathbb{E}_{p(x)}\mathbb{E}_{p(\tilde{x}_i|x)}\left[\|s_\theta(\tilde{x}_i, \sigma_i) + \frac{\tilde{x}_i - x}{\sigma_i^2}\|^2\right]. \tag{41}$$

Anneled Langevin MCMC for sampling

Microsoft Research AI4Science

$$x_t = x_{t-1} + \frac{\Delta t}{2} S_\theta(x_{t-1}) + \sqrt{\Delta t}\epsilon, \quad \epsilon \sim N(0, 1). \tag{42}$$

DDPM

denoising diffusion probabilistic models (DDPM) (Ho, 2020)

1. Derived from the Evidence Lower BOund (ELBO), Not score matching.
2. First provide adding noise and denoise in the forward and reverse process.

The main procedure

1. Adding noise
$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon; \tag{43}$$
2. Transition kernel
$$p(x_t|x_0) = N(x_t; \alpha_t x_0, \sigma_t^2), \tag{44}$$
where $\alpha_t = \prod_{i=1}^{T} \sqrt{1-\beta_t}$ and $\sigma_t^2 = 1 - \alpha_t$.

Microsoft Research AI4Science

DDPM

The main procedure (Continued)

1. Adding noise

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon; \tag{45}$$

2. Transition kernel

$$p(x_t|x_0) = N(x_t; \alpha_t x_0, \sigma_t^2), \text{ i.e., } x_t = \alpha_t x_0 + \sigma_t \epsilon, \tag{46}$$

   where $\alpha_t = \prod_{i=1}^{T} \sqrt{1-\beta_t}$ and $\sigma_t^2 = 1 - \alpha_t$.

3. Training Losss

$$\mathbb{E}_{t\sim U[0,1]}\mathbb{E}_{p(x)}\mathbb{E}_{p(x_t|x)}\left[\|\epsilon_\theta(x_t,t)-\epsilon\|^2\right], \tag{47}$$

   where we have used the approximation $x_0 = \frac{x_t - \sigma_t\epsilon}{\alpha_t}$.

4. Sampling

$$x_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(x_t + \beta_t s_\theta(x_t,t)) + \sqrt{\beta_t}\epsilon_t, \quad t = T, T-1, \ldots, 1. \tag{48}$$

Microsoft
Research
AI4Science

Outline

1. Preliminary: Generating Samples from Probability Distributions
2. Matching score for training
3. Relations of different models: x_0, ϵ and score model.
4. Diffusion model: from theories to implementation
5. **Continuous version of Diffusion Model**
6. Known and unkown questions

Microsoft Research AI4Science

The continuous version of SMLD

Recall that

$$x_t = x_0 + \sigma_t \epsilon, \tag{49}$$

we can prove that

$$x_t = x_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2}\epsilon. \tag{50}$$

Proof.

We can prove the formula by a Recurrence

$$\begin{aligned} x_t &= x_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2}\epsilon \\ &= x_{t-2} + \sqrt{\sigma_{t-1}^2 - \sigma_{t-2}^2}\epsilon + \sqrt{\sigma_t^2 - \sigma_{t-1}^2}\epsilon \\ &= x_{t-2} + \sqrt{\sigma_t^2 - \sigma_{t-2}^2}\epsilon \\ &= \ldots = x_0 + \sigma_t \epsilon, \qquad \text{where } \sigma_0 \approx 0. \end{aligned} \tag{51}$$

□

The continuous version of SMLD

From

$$x_t = x_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2}\epsilon, \tag{52}$$

we have

$$\begin{aligned}\Delta x_t &= \sqrt{\frac{\sigma_t^2 - \sigma_{t-1}^2}{\Delta t}}\sqrt{\Delta t}\epsilon \\ &= \sqrt{\frac{\sigma_t^2 - \sigma_{t-1}^2}{\Delta t}}\Delta B_t.\end{aligned} \tag{53}$$

Further, let $\Delta t \to 0$, we have

$$dx = \sqrt{\frac{d\sigma_t^2}{dt}}dB_t. \tag{54}$$

Microsoft Research AI4Science

The continuous version of DDPM

Recall that

$$x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t}\epsilon. \tag{55}$$

Similarly

$$x_t \approx (1 - \frac{1}{2}\beta_t) x_{t-1} + \sqrt{\beta_t}\epsilon, \tag{56}$$

hence, we have

$$\begin{aligned} \Delta x_t &= -\frac{1}{2}\tilde{\beta}_t x_{t-1}\Delta t + \sqrt{\tilde{\beta}_t}\sqrt{\Delta t}\epsilon, \quad \text{where } \tilde{\beta}_t = \frac{\beta_i}{\Delta t} \\ &= -\frac{1}{2}\tilde{\beta}_t x_{t-1}\Delta t + \sqrt{\tilde{\beta}_t}\Delta B_t. \end{aligned} \tag{57}$$

Further, let $\Delta t \to 0$, we finally obtain

$$dx = -\frac{1}{2}\tilde{\beta}_t x dt + \sqrt{\tilde{\beta}_t} dB_t. \tag{58}$$

Microsoft Research AI4Science

VE & VP

Summarizing the form of continuous DDPM and SMLD, we summarize the adding noise process as

$$dx = f(x,t)dt + g(t)dB_t, \tag{59}$$

where the corresponding distribution $p(x,t)$ satisfying the following Fokker-Planck equation

$$\frac{\partial p(x,t)}{\partial t} + \nabla \cdot (f(x,t)p(x,t)) - \frac{1}{2}g^2(t)\triangle p(x,t) = 0. \tag{60}$$

Training DSM object

$$J_{DSM} = \mathbb{E}_{t\sim U[0,1]}\mathbb{E}_{p(x_0)}\mathbb{E}_{p(x_t|x_0)}\left[\|s_\theta(x_t,t) - \nabla_{x_t}\log p(x_t|x_0)\|^2\right]. \tag{61}$$

Microsoft
Research
AI4Science

VE & VP adding noise

To training the Denoising Score Match loss J_{DSM} above, the key is to add noise by the transition kernel

$$p(\tilde{x}|x) \equiv p(x(t)|x(0)) = N(x(t); \tilde{\alpha}_t x(0), \tilde{\sigma}_t^2), \tag{62}$$

from which we can obtain the noised data

$$x(t) = \tilde{\alpha}_t x(0) + \tilde{\sigma}_t \epsilon, \quad \epsilon \sim N(0, 1). \tag{63}$$

The question becomes: Given VE & VP

$$dx = \sqrt{\frac{d[\sigma^2]}{dt}} dB_t, \quad dx = -\frac{1}{2}\tilde{\beta}_t x dt + \sqrt{\tilde{\beta}_t} dB_t. \tag{64}$$

how to derive $p(x(t)|x(0))$, i.e., $p(\tilde{x}|x)$ for adding noise?

Microsoft Research AI4Science

VE & VP Transition kernel

The summarizing form of VE & VP is

$$dx = h(t)xdt + g(t)dB_t, \tag{65}$$

where $h(t) = 0$ for VE and $h(t) = -\frac{1}{2}\tilde{\beta}(t)$ for VP.

Lemma

Let $\mu(t) = \mathbb{E}[x(t)]$ and $\Sigma(t) = \mathbb{E}\left[(x-\mu(t))(x-\mu(t))^T\right]$, we have the following formula

$$\frac{d\mu(t)}{dt} = h(t)\mu(t), \quad \frac{d\Sigma(t)}{dt} = 2h(t)\Sigma(t) + g^2(t)I. \tag{66}$$

The proof of the Lemma can be found in a stochastic differential equation textbook such as Applied Stochastic Differential Equations, Särkkä and Solin, 2019.

Microsoft Research AI4Science

VE & VP Transition kernel

Lemma

For VE, the transition kernel is given by the following formula

$$P(x(t)|x(0)) = N(x(t); x(0), \sigma^2(t) - \sigma^2(0)). \tag{67}$$

For VE,

$$\tilde{\alpha}_t \equiv 1, \quad \tilde{\sigma}_t^2 \equiv \sigma^2(t) - \sigma^2(0) \approx \sigma^2(t). \tag{68}$$

Lemma

For VP, the transition kernel is given the following formula

$$p(x(t)|x(0)) = N\left(x(t); x(0)e^{-\frac{1}{2}\int_0^t \tilde{\beta}(s)ds}, 1 - e^{-\int_0^t \tilde{\beta}(s)ds}\right). \tag{69}$$

For VP,

$$\tilde{\alpha}_t \equiv e^{-\frac{1}{2}\int_0^t \tilde{\beta}(s)ds}, \quad \tilde{\sigma}_t^2 \equiv 1 - e^{-\int_0^t \tilde{\beta}(s)ds}. \tag{70}$$

Proof of VE transition kernel

Proof.

For VE, by a simple calculation we have

$$\frac{d\mu(t)}{dt} = 0, \quad \frac{d\Sigma(t)}{dt} = \frac{d[\sigma^2(t)]}{dt}, \tag{71}$$

Hence, it is easy to obtain the following form

$$\mu(t) = \mu(0) = x(0), \quad \Sigma(t) = \sigma^2(t) - \sigma^2(0). \tag{72}$$

Here we have used the fact

$$\mu(0) = x(0), \quad \Sigma(0) = 0 \tag{73}$$

as $x(0)$ is taken as a constant, not a random variable. And we obtain

$$P(x(t)|x(0)) = N(x(t); x(0), \sigma^2(t) - \sigma^2(0)). \tag{74}$$

Proof of VP transition kernel

Proof.

For VP, by a simple calculation we have

$$\frac{d\mu(t)}{dt} = -\frac{1}{2}\beta(t)\mu(t), \quad \frac{d\Sigma(t)}{dt} = -\beta(t)\Sigma(t) + \beta(t). \tag{75}$$

Multiply the exponential term on the both sides, we have

$$\begin{aligned} &\frac{d\mu(t)}{dt}e^{\int_0^t \frac{1}{2}\beta(s)ds} + \frac{1}{2}\beta(t)\mu(t)e^{\int_0^t \frac{1}{2}\beta(s)ds} = 0, \\ &\frac{d\Sigma(t)}{dt}e^{\int_0^t \beta(s)ds} + \beta(t)\Sigma(t)e^{\int_0^t \beta(s)ds} = \beta(t)e^{\int_0^t \beta(s)ds}. \end{aligned} \tag{76}$$

Hence, we have the following form

$$\frac{d}{dt}(\mu(t)e^{\int_0^t \frac{1}{2}\beta(s)ds}) = 0, \quad \frac{d}{dt}(\Sigma(t)e^{\int_0^t \beta(s)ds}) = \frac{d}{dt}e^{\int_0^t \beta(s)ds}. \tag{77}$$

By a direct calculation, we can obtain

$$p(x(t)|x(0)) = N\left(x(t); x(0)e^{-\frac{1}{2}\int_0^t \beta(s)ds}, 1 - e^{-\int_0^t \beta(s)ds}\right). \tag{78}$$

Revisit adding noise

The **equivalence** of the adding noise

1. From the stochastic process, denote the variable x_t, we have

$$dx_t = f(x,t)dt + g(t)dt, \tag{79}$$

with corresponding discrete form (Eular-Maruyama scheme)

$$x_{t+\Delta t} = x_t + f(x_t,t)\Delta t + g(t)\sqrt{\Delta t}\epsilon, \quad \epsilon \sim N(0,1). \tag{80}$$

2. From transition kernel perspective,

$$p(x_t) = \int_{x_0} p(x_0)p(x_t|x_0)dx_0, \tag{81}$$

where

$$p(x_t|x_0) = N(x_t;\mu_t,\Sigma_t) = N(x_t;\tilde{\alpha}_t x_0, \tilde{\sigma}_t^2) = \frac{1}{C}e^{-\frac{\|x_t-\mu_t\|^2}{2\Sigma_t}} \tag{82}$$

Microsoft Research AI4Science

Training recipe

The training for VE follows the following process (VP is in a similar way)

1. Random choose one data $x \sim p_{data}$;
2. Random sample $t \sim U[0, 1]$;
3. Random sample a white noise $\epsilon \sim N(0, 1)$;
4. Adding noise
$$x_t = \tilde{\alpha}_t x + \tilde{\sigma}_t \epsilon, \tag{83}$$
where the mean value $\tilde{\alpha}_t x = x(0)$ and standard deviation $\tilde{\sigma}_t = \sqrt{\sigma^2(t) - \sigma^2(0)} \approx \sigma(t)$ in VE referring to (74). VP is similar with the mean and standard deviation from its transition kernel (78);
5. Compute the Loss by summation the following norm over x, t and ϵ
$$\|s_\theta(x_t, t) - \epsilon/\tilde{\sigma}_t\|. \tag{84}$$

Microsoft Research AI4Science

Inference: Reverse denoising process

The reverse denoising process is given by

$$dx = (f(x,t) - g^2\nabla_x \log p(x,t))dt + g(t)dB_t, \tag{85}$$

or

$$dx = (f(x,t) - \frac{1}{2}g^2\nabla_x \log p(x,t))dt, \tag{86}$$

or

$$dx = (f(x,t) - \frac{3}{2}g^2\nabla_x \log p(x,t))dt + \sqrt{2}g(t)dB_t, \tag{87}$$

$$\cdots \tag{88}$$

Due to obeying the same Fokker-Planck equation for $p(x,t)$

$$\frac{\partial p(x,t)}{\partial t} + \nabla \cdot (f(x,t)p(x,t)) - \frac{1}{2}g^2(t)\triangle p(x,t) = 0. \tag{89}$$

Microsoft Research AI4Science

Inference: Reverse denoising process (proof)

In this slide, we only prove the first reverse sampling form (85) and other cases can be done in a similar approach.

Proof.

The Fokker-Planck equation of the forward process (59) is given by (60) as

$$\frac{\partial p}{\partial t} + \nabla \cdot (f(x,t)p) - \frac{1}{2} g^2 \triangle p = 0. \tag{90}$$

Let $t = T - \tau$, we have

$$\frac{\partial p}{\partial \tau} - \nabla \cdot (f(x, T-\tau)p) + g^2 \triangle p - \frac{1}{2} g^2 \triangle p = 0. \tag{91}$$

By $\nabla \cdot \nabla = \triangle$ and $\nabla \log p = \nabla p / p$, we have

$$\frac{\partial p}{\partial \tau} - \nabla \cdot \Big(\left(f(x, T-\tau) - g^2 \nabla \log p \right) p \Big) - \frac{1}{2} g^2 \triangle p = 0. \tag{92}$$

Hence, we obtain the corresponding SDE form $dx = -(f(x, T-\tau) - g^2 \nabla \log p) d\tau + g dB_\tau$. By a substitution $\tau = T - t$, we finally get $dx = (f(x,t) - g^2 \nabla \log p) dt + g dB_t$. □

Inference recipe

The inference for VE follows the following process (VP is in a similar way)

1. Random generate a noise at time $t = 1$ as $x(1) \sim N(0, \sigma_{max}^2)$;
2. Integrate the reverse sampling equation

$$dx = \big(f(x,t) - g^2(t)s_\theta(x,t)\big)\, dt + g(t)dB_t \tag{93}$$

   or

$$dx = (f - \frac{1}{2}g^2 s_\theta)dt \tag{94}$$

   over the time span $[0, 1]$ to get $x(0)$.
3. Corrector process: at each internal value $x(t)$, we can relax the value $x(t)$ by a corrector, which takes the Langevin MCMC process.

Microsoft Research AI4Science

The Corrector: revisit Langevin MCMC

The Langevin diffusion process is given by

$$x_i = x_{i-1} + \frac{\Delta t}{2} \nabla_x \log p(x_{i-1}) + \sqrt{\Delta t}\epsilon, \quad \epsilon \sim N(0,1). \tag{95}$$

We can prove $\pi(x)$ of the obtained data points $\{x_i\}_{i=1}^N$ converges to $p(x)$.

Proof.

The continuous form of above scheme is given by

$$dx = \frac{1}{2}\nabla \log p(x) dt + dB_t. \tag{96}$$

The corresponding Fokker-Planck equation is written as

$$\frac{\partial \pi(x,t)}{\partial t} + \nabla \cdot (\frac{1}{2}\nabla \log p(x)\pi(x,t)) - \frac{1}{2}\triangle \pi(x,t) = 0. \tag{97}$$

With a enough long time evolution, the distribution converges to a stable distribution where

$$\frac{\partial \pi(x,t)}{\partial t} = 0, \quad t \to \infty. \tag{98}$$

The Corrector: revisit Langevin MCMC

Proof (Cont).

Hence we have

$$\nabla \cdot (\pi \nabla \log p - \nabla \pi) = 0, \quad \forall x. \tag{99}$$

Therefore, we have

$$\nabla \log p = \nabla \log \pi, \tag{100}$$

given that $\pi > 0$ almost true. Hence we have

$$\pi^*(x) = p(x). \tag{101}$$

where π^* is the stable distribution of $\pi(x, t)$.

Microsoft Research AI4Science

Outline

Microsoft Research AI4Science

Questions

If you are interested in the questions, please feel free to write an email to me (pisquare@microsoft.com) with your comments.

1. Please prove that the forms (85)-(87) are equivalent with same marginal distribution $p(x, t)$ given the same initial condition;
2. Please prove that the score function $s(x, t)$ satisfying the following formula
$$\frac{\partial s}{\partial t} = \nabla(-\nabla \cdot f - \langle f, s \rangle + \frac{1}{2}g^2\|s\|^2 + \frac{1}{2}g^2\langle \nabla, s \rangle); \tag{102}$$
3. Why ISM loss is not commonly used like DSM loss in the diffusion model nowadays? Please make your comments.

Microsoft
Research
AI4Science

Welcome inputs

Any comments?

Welcome your inputs!

Microsoft Research AI4Science