401-4634-24L: Diffusion Models, Sampling and Stochastic Localization

Lecture 5 – Diffusion model and its convergence

Lecturer: Yuansi Chen *Spring 2024*

Key concepts:

- Denoising diffusion probabilistic modeling (DDPM)
  - Forward process + reverse process
  - Score estimation in DDPM
  - Discretization and implementation
- Convergence analysis

The material of this lecture is based on [CCL+22].

5.1 Denoising diffusion probabilistic modeling

In the previous lecture, we explain the idea of annealed Langevin algorithms with L noise levels. When the number of noise levels tends to infinity, we essentially perturb the data distribution with continuously growing levels of noise. It is natural to first study the convergence of the continuous analogue of the annealed Langevin algorithms, which is a continuous-time stochastic process.

In particular, we focus on the denoising diffusion probabilistic modeling from [SSDK+20]. It has a forward process which generates the perturbed data distribution, and a reverse process which transform noise into new samples from μ.

To be consistent with the notation in [CCL+22], we use both $q := \mu$ and μ for the target measure, and $x_1, \ldots, x_N$ for the i.i.d. samples from q.

Forward process. The forward process is specified via a stochastic differential equation (SDE).

$$d\bar{X}_t = -\bar{X}_t dt + \sqrt{2} dB_t, \quad \bar{X}_0 \sim q, \tag{5.1}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion in $\mathbb{R}^n$. This type of process is also called the Ornstein-Uhlenbeck (OU) process. In practice, one may consider the time-rescaled OU process to adjust the amount of added noise as a function of time: $d\bar{X}_t = -g(t)^2 \bar{X}_t dt + \sqrt{2} g(t) dB_t$, with a positive smooth function $g : \mathbb{R}_+ \to \mathbb{R}_+$. For simplicity, we stick with Eq. (5.1) and the choice $g = 1$.

The forward process has the interpretation of transforming samples from the data distribution q into pure noise. Intuitively, $\sqrt{2}dB_t$ part of the SDE adds noise and the $-\bar{X}_t dt$ part adjusts the magnitude of $\bar{X}_t$ so that its magnitude does not go to infinity. We can have a closed-form solution of Eq. (5.1) as follows. Let $\bar{Y}_t := e^t \bar{X}_t$. Then

$$\begin{aligned} d\bar{Y}_t &= e^t \bar{X}_t dt + e^t d\bar{X}_t \\ &\overset{(i)}{=} e^t \bar{X}_t dt + e^t \left[-\bar{X}_t dt + \sqrt{2} dB_t \right] \\ &= \sqrt{2} e^t dB_t, \end{aligned}$$

where (i) replaces $d\bar{X}_t$ using Eq. (5.1). Then for any $t \geq 0$, $\bar{Y}_t$ is a Gaussian random variable. It suffices to calculate its mean and variance. We have

$$\begin{aligned} \mathbb{E}[\bar{Y}_t] &= 0 \\ \mathbb{E}[\bar{Y}_t \bar{Y}_t^\top] &= 2\mathbb{E}\left[\left(\int_0^t e^s dB_s \right) \left(\int_0^t e^s dB_s \right)^\top \right] \\ &\overset{(i)}{=} 2 \int_0^t (e^{2s}) ds \mathbb{I}_n \\ &= \left(e^{2t} - 1 \right). \end{aligned}$$

(i) follows from Itô's isometry. Hence, we may write

$$\bar{Y}_t = \bar{Y}_0 + \left(e^{2t} - 1 \right)^{\frac{1}{2}} Z,$$

where $Z \sim \mathcal{N}(0, \mathbb{I}_n)$. And

$$\bar{X}_t = e^{-t} \bar{X}_0 + \left(1 - e^{-2t} \right)^{\frac{1}{2}} Z. \tag{5.2}$$

In words, $\bar{X}_t$ is a linear combination of the measure q and noise, with weights $(e^{-t}, (1 - e^{-2t})^{\frac{1}{2}})$ that have their squares sum to 1.

Reverse process. If we reverse the forward process (5.1) in time, then we obtain a process that transforms noise into samples from q, which is what we desire in generative modeling. In general, suppose we have an SDE of the form

$$d\bar{X}_t = a(\bar{X}_t, t) dt + b_t dB_t.$$

Under mild conditions on the process, the process can be reversed, and the reverse process also admits an SDE description. Fix terminal time $T > 0$, define the reverse process

$$\bar{X}_t^{\leftarrow} := \bar{X}_{T-t}, \qquad \text{for } t \in [0, T],$$

then the process $(\bar{X}_t^{\leftarrow})_{t\in[0,T]}$ satisfies the following reverse SDE

$$d\bar{X}_t^{\leftarrow} = a^{\leftarrow}(\bar{X}_t^{\leftarrow}, t)dt + b_{T-t}dW_t, \quad \bar{X}_0^{\leftarrow} \sim q_T,$$

where W_t is the reversed Brownian motion, and the reverse drift satisfies

$$a(x,t) + a^{\leftarrow}(x, T-t) = b_t b_t^{\top} \nabla \log q_t, \text{ where } q_t := \text{law}(\bar{X}_t).$$

For simplicity, we don't distinguish the forward Brownian motion B_t and the reversed Brownian motion W_t and as a consequence, we always state that the law of the $\bar{X}_t^{\leftarrow}$ defined through the reverse SDE is the same as that of $\bar{X}_{T-t}$ instead of stating almost sure equality.

Applying the above result to the forward process (5.1), we obtain the reverse process in DDPM

$$d\bar{X}_t^{\leftarrow} = \left[\bar{X}_t^{\leftarrow} + 2\nabla \log q_{T-t}(\bar{X}_t^{\leftarrow})\right] dt + \sqrt{2}dB_t, \quad \bar{X}_0^{\leftarrow} \sim q_T, \tag{5.3}$$

where $(B_t)_{t\in[0,T]}$ is the reversed Brownian motion. Note that $\nabla \log q_t(\cdot)$ is the score function of the measure q_t, which according to Eq. (5.2) has the law of a linear combination of q and Gaussian noise. Since q is not explicitly known and is only known via its samples $x_1, \ldots, x_N$, in order to implement the reverse process, we need to estimate the score function at any time $t \in [0,T]$ via the samples.

5.1.1 Score estimation in DDPM

As we have explained in the previous lecture, one popular way of estimating score function is via denoising score matching. We review the basics here. We want to estimate the score function $\nabla \log q_t$ by minimizing the Fisher divergence between the density induced by the score function and q_t,

$$\min_{s_t \in \mathfrak{F}} \mathbb{E}_{X\sim q_t} \left\| s_t(X) - \nabla \log q_t \right\|_2^2,$$

where $\mathfrak{F}$ is a function class where we search for s_t, which could be a parameterized class of neural networks. The denoising score matching transforms the above objective to the following equivalent problem, via smart applications of integration by parts,

$$\min_{s_t \in \mathfrak{F}} \mathbb{E} \left\| s_t(\bar{X}_t) + \left(1 - e^{-2t}\right)^{-\frac{1}{2}} Z_t \right\|_2^2 \tag{5.4}$$

where $Z_t \sim \mathcal{N}(0, \mathbb{I}_n)$ independent of $\bar{X}_0$ and $\bar{X}_t = e^{-t}\bar{X}_0 + (1-e^{-2t})^{\frac{1}{2}} Z_t$ according to Eq. (5.2). The main advantage of the formulation (5.4) is that it can be easily replaced

with it empirical counterpart. This allows us to estimate the score on the samples $x_1, \dots, x_N$.

$$\min_{s_t \in \mathfrak{F}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{E} \left\| s_t(\bar{X}_t^{(i)}) + \left(1 - e^{-2t}\right)^{-\frac{1}{2}} Z_t^{(i)} \right\|_2^2, \tag{5.5}$$

where $Z_t^{(i)}$ is independent standard Gaussian noise, $\bar{X}_t^{(i)}$ can be obtained by adding noise to x_i. The above problem has the interpretation of predicting the added noise $Z_t^{(i)}$ from the noisy data $\bar{X}_t^{(i)}$.

In practice, since the denoising score matching formulation (5.5) share some similarity across all $t \in [0, T]$, it is common to use a shared neural network architecture [HJA20] which takes input the noisy data $\bar{X}_t^{(i)}$ and a transformation of time t, and outputs the noise $Z_t^{(i)}$. However, since the score functions at small t and large t are likely to be very different, the statistical advantage of having a shared neural network architecture other than its evident computational advantage is unclear.

5.1.2 Discretization and Implementation

First, in the score estimation phase, given samples $x_1, \dots, x_N$ from q, we generate the noisy samples and train a neural network to estimate the score functions at all time levels via denoising score matching.

Second, we wish to run the reverse process to generate new samples starting from noise. Once we have the score estimate s_t from denoising score matching, we can replace $\nabla \log q_{T-t}$ in Eq. (5.3) by s_t. However, for a general s_t, it is still hard to integrate the reverse process Eq. (5.3) in continuous time.

Let $h > 0$ be the step-size of the discretization. We discretize the reverse SDE as follows, for $t \in [kh, (k+1)h]$,

$$dX_t^{\leftarrow} = [X_t^{\leftarrow} + 2s_{T-kh}(X_{kh}^{\leftarrow})]\, dt + \sqrt{2} dB_t. \tag{5.6}$$

Note that we fixed the argument of $s_{T_{kh}}$ to be the value of $X^{\leftarrow}$ at the beginning of the segment $X_{kh}^{\leftarrow}$, in order to result in a linear SDE which can be integrated in closed form.

Finally, ideally, we would like to run the reverse process starting from q_T. However, we do not have access to q_T directly. Taking advantage of $q_T \approx \gamma^n$ for large T, we instead initialize the algorithm at $X_0^{\leftarrow} \sim \gamma^n$, i.e., pure Gaussian noise.

Let $p_t := \text{law}(X_t^{\leftarrow})$ denote the law at time t. Taking into account of the three implementation details above, running the reverse SDE (5.6) using estimated score and starting from γ^n to generate new samples from q would make three types of errors

1. Score estimation error. This error is mostly statistical in nature, depending on the sample size N, the size of the function class $\mathfrak{F}$ and its closeness to the true score function.

2. The discretization of the reverse process, which depends on the step-size h.

3. The error made at initialization, γ^n used instead of q_T.

5.2 Convergence analysis

[CCL+22] analyzed the total variation distance between the law of generated samples p_T and the target measure q for large T, $d_{\text{TV}}(p_T, q)$ as a function of the three types of errors mentioned above, under the following three assumptions

1. (Lipschitz score). For any $t \geq 0$, the score $\nabla \log q_t$ is L-Lipschitz.

2. (Second moment bound). Assume that $M_2^2 := \mathbb{E}_{X \sim q} \|X\|_2^2 < \infty$.

3. (Score estimation error bound). For $k = 1, \ldots, K$,

$$\mathbb{E}_{q_{kh}} \|s_{kh} - \nabla \log q_{kh}\|_2^2 \leq \epsilon_{\text{score}}^2.$$

Theorem 5.2.1 (DDPM convergence in [CCL+22]). *Under the three assumptions above. Let p_T be the output of the DDPM algorithm (5.6) at time $T > 0$, with $h = T/K$ and K the number of steps, suppose $h \lesssim 1/L$, then*

$$d_{TV}(p_T, q) \lesssim \underbrace{\sqrt{KL(q \parallel \gamma^n)} \exp(-T)}_{\text{convergence of forward process}} + \underbrace{(L\sqrt{nh} + LM_2 h)\sqrt{T}}_{\text{discretization error}} + \underbrace{\epsilon_{score}\sqrt{T}}_{\text{score estimation error}}.$$

Remarks

1. Unlike previous results on Ball walk or Langevin algorithms, the above theorem does not assume any type of "bottleneck" condition such as Poincaré inequality or isoperimetric inequality. It means that DDPM can efficiently sample from multi-modal target measures as long as the score estimation is good.

2. Even though the KL divergence term $\text{KL}(q \parallel \gamma^n)$ between q and γ^n might be large (even exponentially in dimension n), the contraction of the forward process creates a $\exp(-T)$ term which can make the first term small.

3. The discretization depends on the Lipschitz parameter of the score, which typically appears in discretization of continuous processes.

Proof ideas.

- First, we can apply triangle inequality to isolate the distance between the outcome of the discrete reverse process starting from γ^n and from q_T.
- Second, it remains to compare the distance between the discrete reverse process started from q_T with estimated score and the continuous reverse process started from q_T with the true score. It reduces to the comparison of two stochastic processes with slightly different drift terms. Applying Girsanov's theorem is one of the typical ways to control their KL divergence.

Proof. See the proof of Theorem 2 in [SSDK+20] for a complete proof. **YC — TODO. Will fill in the main proof steps soon** □

Bibliography

[CCL+22] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[SSDK+20] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.