Sampling and Diffusion Model

Handstein Wang

Institute of Computational Mathematics and Scientific/Engineering Computing
Academy of Mathematics and Systems Science
Chinese Academy of Sciences, China

November 7, 2025

Outline

1. Two Settings of Sampling with Applications

2. Density Estimation

3. Generative Model

4. Score Matching

5. Diffusion Model

Outline

1. **Two Settings of Sampling with Applications**

2. Density Estimation

3. Generative Model

4. Score Matching

5. Diffusion Model

Sampling

Target: We want to sample from a distribution μ on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ which admits a density w.r.t. Lebesgue measure m, also denoted by $\mu(x)$.

There are two settings of sampling problems:

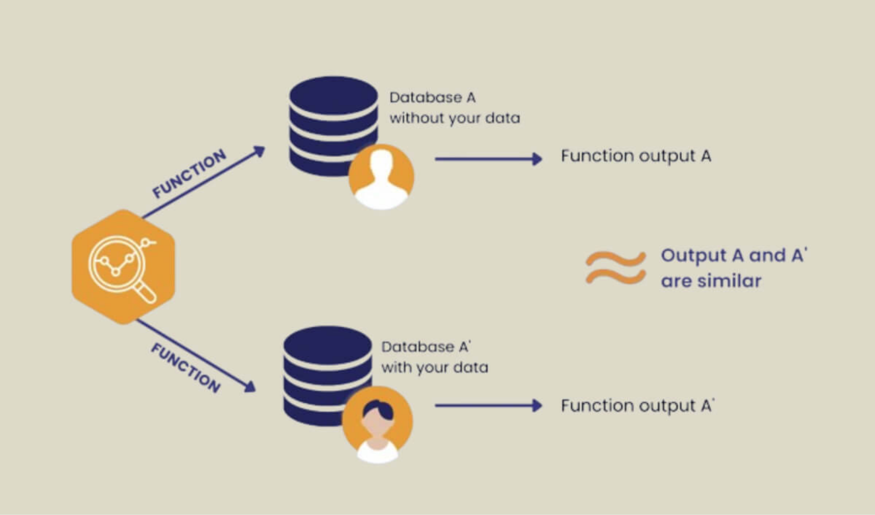
Setting 1: μ is given in explicit form up to a normalization constant.

Applications: Bayesian inference, inverse problem, finance, computation of high dimensional integral, differential privacy, approximate computation, uncertainty quantification, etc.

Setting 2: μ is given with a collection if i.i.d. samples.

Applications: generative models (diffusion models, GANs, etc).

Differential Privacy



Differential Privacy

- Let $(\mathcal{X}, \mathscr{X})$ be a measurable space of individual records. For a fixed $n \in \mathbb{N}$, datasets live in $(\mathcal{X}^n, \mathscr{X}^{\otimes n})$.
- Fix a adjacency relation $\sim \subseteq \ \mathcal{X}^n \times \mathcal{X}^n$ specifying which dataset pairs differ by "one individual".
- Let $(Y, \mathcal{Y})$ be a measurable output space equipped with a σ-finite base measure μ. A randomized algorithm (mechanism) is a Markov kernel

$$K : \ \mathcal{X}^n \times \mathcal{Y} \to [0,1], \qquad (D, A) \ \mapsto \ K(D, A),$$

i.e., for each D, $K(D, \cdot)$ is a probability measure on $(Y, \mathcal{Y})$, and for each A, the map $D \mapsto K(D, A)$ is $\mathscr{X}^{\otimes n}$-measurable.

Definition ($\varepsilon-$DP)

The mechanism K is ε-*differentially private* (w.r.t. $\sim$) if for all adjacent datasets $D \sim D'$ and all measurable events $A \in \mathcal{Y}$,

$$K(D, A) \ \leq \ e^{\varepsilon} K(D', A).$$

Differential Privacy

- A *score function* is a measurable function $u : \mathcal{X}^n \times Y \to \mathbb{R}$.
- The *global sensitivity* of u w.r.t. $\sim$ is

$$\Delta u \;:=\; \sup_{D \sim D'} \; \operatorname*{ess\,sup}_{y \in Y} \left| u(D, y) - u(D', y) \right| \;\in (0, \infty).$$

Definition (Exponential Mechanism)

Fix $\varepsilon > 0$ and set $\alpha := \varepsilon/(2\Delta u)$. For each dataset $D \in \mathcal{X}^n$, the *Exponential Mechanism* is the Markov kernel $K_{\exp}$ given by

$$K_{\exp}(D, A) \;:=\; \frac{\int_A \exp\big(\alpha\, u(D, y)\big)\, \mu(dy)}{\int_Y \exp\big(\alpha\, u(D, y)\big)\, \mu(dy)}, \qquad A \in \mathcal{Y}.$$

Equivalently, $K_{\exp}(D, \cdot)$ has density proportional to $\exp(\alpha\, u(D, \cdot))$ w.r.t. μ.

Differential Privacy

Theorem

The exponential mechanism K_{exp} is ε-differentially private, namely, for all adjacent datasets $D \sim D'$ and all measurable events $A \in \mathcal{Y}$,

$$K_{\mathrm{exp}}(D, A) \;\leq\; e^{\varepsilon}\, K_{\mathrm{exp}}(D', A).$$

By the theorem above, it suffices to sample from the distribution

$$\pi(dy) \propto \exp(\alpha\, u(D, y))\mu(dy)$$

which falls under Setting 1.

Approximate Computation

Example

Let $K \subset \mathbb{R}^n$ be a bounded convex set given via a *membership oracle*, i.e., for any query point $x \in \mathbb{R}^n$ we can decide whether $x \in K$. The task is to approximate $\mathrm{Vol}(K)$.

- It is known that, under a membership oracle, exact volume computation is computationally intractable: computing the volume of a polytope is NP-hard [Dyer and Frieze, 1988].
- Formulate this as sampling from the uniform law on K with unnormalized density

$$\mu(x) \propto \mathbf{1}_K(x),$$

by using some sampling technique, the approximate volume computation achieves constant error with $O(n^3)$ complexity [Jia et al., 2021].

Outline

1. Two Settings of Sampling with Applications
2. **Density Estimation**
3. Generative Model
4. Score Matching
5. Diffusion Model

Reducing the Problem in Setting 2 to Setting 1

In setting 2, μ is given with a collection if i.i.d. samples $x_1, x_2, \cdots, x_N$.

Density Estimation

We can estimate an explicit expression of μ from the i.i.d. samples $x_1, x_2, \cdots, x_N$ by *density estimation*.

There are many methods for density estimation, see [Scott, 2015] for an overview. By density estimation, we can reduce the Problem in Setting 2 to Setting 1.

But density estimation can NOT avoid the curse of dimensionality!

Curse of Dimensionality of Density Estimation

Let $n \in \mathbb{N}$ be the dimension and $\beta > 0$ the smoothness parameter. Write $\beta = \alpha + m$ with $m \in \mathbb{N}$ and $\alpha \in (0, 1]$. Define the class of probability densities $\mathcal{P}_\beta$ on $[0,1]^n$ by those f satisfying:

(1) f is a probability density on $[0,1]^n$ and is bounded, say 2.

(2) The m-th derivative $f^{(m)}$ is α-Hölder continuous, i.e., for all $x, y \in [0,1]^n$,

$$\left| f^{(m)}(x) - f^{(m)}(y) \right| \leq \|x - y\|^\alpha.$$

Theorem ([Tsybakov, 2009])

Given N i.i.d. samples $x_1, \ldots, x_N$ from a pdf $f \in P_\beta$, the minimax risk of an estimation $\hat{f}$ of f under the quadratic loss function $\ell(\hat{f}, f) := \|\hat{f} - f\|_2^2 = \int_{[0,1]^n} (f(x) - \hat{f}(x))^2 \, dx$ satisfies

$$\inf_{\hat{f}} \sup_{f \in P_\beta} \|\hat{f} - f\|_2^2 \; \gtrsim \; N^{-\frac{2\beta}{n+\beta}}.$$

The infimum is taken over all estimators $\hat{f}$ built on the data $x_1, \ldots, x_N$.

Outline

1. Two Settings of Sampling with Applications

2. Density Estimation

3. **Generative Model**

4. Score Matching

5. Diffusion Model

Solving Sampling Problems in Setting 2

In 2021, Song and Ermon [Song and Ermon, 2019] categorized the existing generative modeling techniques into two major categories: **likelihood-based models** and **implicit generative models**.

Likelihood-based models, which directly models the distribution's probability density function via (approximate) maximum likelihood. That is, we model

$$\mu(x) = p_\theta(x),$$

where $p_\theta(x)$ is the probability density parametrized by some parameter θ. We learn θ by maximizing the log-likelihood of the data

$$\max_\theta \sum_{i=1}^{N} \log p_\theta(x_i).$$

This category includes autoregressive models, normalizing flow models, energy- based models, and variational auto-encoders (VAEs).

Solving Sampling Problems in Setting 2

Implicit generative models, which implicitly represent the probability distribution via a model of sampling process. That is, we say the target distribution is close to a transformation of a Gaussian

$$g(Z),\ Z \in \mathcal{N}(0, I_m),$$

where $g : \mathbb{R}^m \to \mathbb{R}^n$ is a measurable mapping to be learned. Then with a distance between two measures $\mathrm{dist}(\cdot, \cdot)$, we want to find g to minimize

$$\mathrm{dist}\left(\mathrm{Law}(g(Z)),\ \frac{1}{N}\sum_{i=1}^{n} \delta_{x_i}\right).$$

The most prominent example is generative adversarial networks (GANs), where new samples are synthesized by transforming a random Gaussian vector with a neural network g. The parameters of the neural network g are learned via minimizing the adversarial loss between newly generated images and the empirical measure.

Solving Sampling Problems in Setting 2

- **Likelihood-based models:** Either impose strong architectural restrictions to make the normalizing constant tractable for likelihood computation, or rely on surrogate objectives that only approximate maximum likelihood.
- **Implicit generative models:** Typically require adversarial training, which is notoriously unstable and prone to mode collapse.

Here we introduce another way to represent probability distributions that

- is an iterative sampling process, which does not rely on a good distance between two measures,
- models the score function, rather than the likelihood, avoiding the normalizing constant.

The key idea is to model the gradient of the log probability density function, a quantity often known as the **(Stein) score function**. Such score-based models are not required to have a tractable normalizing constant, and can be directly learned by score matching.

Outline

1. Two Settings of Sampling with Applications

2. Density Estimation

3. Generative Model

4. **Score Matching**

5. Diffusion Model

Score Matching

Definition (Score function)

Let μ be a probability measure on $\mathbb{R}^d$ with density also denoted by $\mu(x)$ with respect to Lebesgue measure satisfying $\mu \in W^{1,1}_{\mathrm{loc}}(\mathbb{R}^d)$ and $\mu(x) > 0$ a.e. Then the *score* of μ is the vector field defined by

$$s(x) := \nabla \log \mu(x) \quad a.e.,$$

where $\nabla \log \mu$ is to be interpreted in the sense of *weak derivatives*.

Equivalently, $s(x) = (s_1(x), \cdots, s_d(x))$ is the unique vector field satisfying, for all $\varphi \in C_c^\infty(\mathbb{R}^d)$ and each $i = 1, \cdots, d$,

$$\int_{\mathbb{R}^d} \partial_i \varphi \, d\mu \;=\; -\int_{\mathbb{R}^d} \varphi \, s_i \, d\mu.$$

Even if we only know μ up to a normalizing constant, $\mu \propto e^{-f}$, we can compute the score function without knowing the normalizing constant

$$\nabla \log \mu(x) \;=\; \nabla \log e^{-f} \;=\; -\nabla f(x).$$

Score Matching

Now we assume the score of μ exists and $\nabla \log \mu \in L^2(\mu)$.

To estimate the score function, it is natural to parametrize the score function by a parameter θ via $s_\theta(x)$, and try to minimize its distance to the true score function over data sampled from the target measure:

$$\min_\theta \ \mathbb{E} \left\| \nabla \log \mu(X) - s_\theta(X) \right\|_2^2, \quad X \sim \mu. \tag{SM}$$

In general, it is difficult to evaluate the objective function above when μ is represented via N samples. $\nabla \log \mu(x)$ is difficult to evaluate with only data samples, for this, a family of methods called score matching is introduced.

Denoising Score Matching

Instead of solving score matching problem (SM), we consider the following denoising score matching problem

$$\min_{\theta} \quad DSM(s_\theta) := \mathbb{E} \left\| \nabla \log \mu_Y(Y) - s_\theta(Y) \right\|_2^2, \quad , \tag{DSM}$$

where $Y = aX + \sigma Z$, $a \in \mathbb{R}, \sigma > 0$,$X \sim \mu$, $Z \sim N(0, I_d)$, Z is independent of X and μ_Y is the density of Y w.r.t. Lebesgue measure .

Theorem

Under the above assumptions and define

$$DSM'(s_\theta) = \mathbb{E}\left[\left\| s_\theta(Y) + \frac{1}{\sigma} Z \right\|_2^2 \right].$$

Then

$$DSM'(s_\theta) = DSM(s_\theta) + \frac{1}{\sigma^2} \mathbb{E} \left\| Z - \mathbb{E}[Z|Y] \right\|_2^2.$$

Denoising Score Matching

The main idea of the proof is using the Stein's lemma to show the following lemma.

Lemma

$$\nabla \log \mu_Y(Y) = -\frac{1}{\sigma}\mathbb{E}[Z|Y] \quad a.s.$$

By the previous theorem,

$$\arg\min_\theta DSM(s_\theta) = \arg\min_\theta DSM'(s_\theta) = \arg\min_\theta \mathbb{E}\left[\left\| s_\theta(Y) + \frac{1}{\sigma}Z \right\|_2^2\right].$$

Then, the objective in the right hand side can be replaced with an empirical version

$$\min_\theta \ \frac{1}{N}\sum_{i=1}^{N}\left\| s_\theta(ax_i + \sigma z_i) + \frac{1}{\sigma}z_i \right\|_2^2$$

Remark: The score we learned here is the score of $\mu_Y = a_{\#}\mu \ * \ N(0, \sigma^2 I_d)$ rather than the score of μ, which provides a good estimate of μ when σ is small enough.

Score-based Langevin Monte Carlo Algorithm

Recall the Langevin Monte Carlo algorithm for sampling $\mu \propto e^{-f}$,

$$X_{k+1} = X_k - h\nabla f(X_k) + \sqrt{2h}\,\xi_k,$$

where $h > 0$ is step-size and $\xi_k \overset{\text{i.i.d.}}{\sim} N(0, I_d)$. Replacing the $-\nabla f(\cdot)$ with estimated score leads to the following.

Score-based Langevin Monte Carlo Algorithm

$$X_{k+1} = X_k + hs_\theta(X_k) + \sqrt{2h}\,\xi_k,$$

where $s_\theta(x)$ is an estimate of score function $\nabla \log \mu(x) = -\nabla f(x)$.

As long as the score estimate is good ($s_\theta \approx \nabla \log \mu(\cdot)$) and h is small, we expect that the law of μ_K to be close to μ for K large enough.

Main Challenges

- Estimated score function is inaccurate in low density regions.

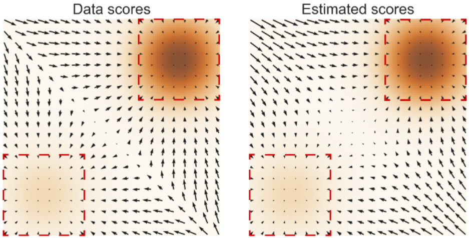


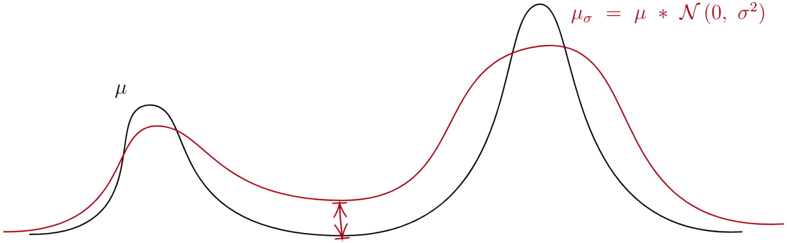
Figure 2: **Left**: $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$; **Right**: $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$. The data density $p_{\text{data}}(\mathbf{x})$ is encoded using an orange colormap: darker color implies higher density. Red rectangles highlight regions where $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) \approx \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$.

- Langevin algorithms mix slowly when faced a bottleneck.

Annealed Langevin Algorithm

A key observation here is that once we smooth μ by convolving it with large Gaussian noise, $\mu_\sigma := \mu * N(0, \sigma^2 I_d)$, then both problems are gone for the problem of sampling μ_σ. However, to sample from μ, it is not enough to say that we can sample from μ_σ with large enough noise σ. [Song and Ermon, 2020] proposed to improve Langevin algorithms with estimated score by

1. perturbing the data using various levels of noise
2. estimating score and run Langevin algorithms, at all noise levels.



Annealed Langevin Algorithm

Inspired by the idea of annealing , the annealed Langevin algorithm is proposed. Set L levels of noise from large to small, $\sigma_1, \ldots, \sigma_L$, and $h_L > 0$

- For i from 1 to L run
  - Set step-size $h_i = h_L \sigma_i^2 / \sigma_L^2$.
  - Run unadjusted Langevin algorithm for K steps

$$X_{k+1} = X_k + h_i s_\theta(X_k, \sigma_i) + \sqrt{2h_i}\, Z_k,$$

  where Z_k is independent standard Gaussian noise, and $s_\theta(X_k, \sigma_i)$ is the estimated score function for $\mu * N(0, \sigma_i^2 I_d)$.

Questions:

- How to set noise levels $\sigma_1, \cdots, \sigma_L$ in practice?
- Does the annealed Langevin algorithm converges to the target measure as our intuition suggest?

Outline

1. Two Settings of Sampling with Applications

2. Density Estimation

3. Generative Model

4. Score Matching

5. **Diffusion Model**

Diffusion Model: Forward Process

Forward Process

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

$$dX_t = -X_t dt + \sqrt{2}\, dB_t, \quad X_0 \sim q_0 := \mu, \qquad \text{(forward process)}$$

where $(B_t)_{t \geq 0}$ is a standard Brownian motion in $\mathbb{R}^d$.

- The solution of this SDE is also called the Ornstein-Uhlenbeck (OU) process.
  (1) The solution is
  $$X_t = e^{-t} X_0 + \sqrt{2} \int_0^t e^{-(t-s)} dB_s.$$
  (2) $\mathbb{E}\, X_t = e^{-t}\, \mathbb{E}\, X_0$, $\mathrm{Var}(X_t) = 1 - e^{-2t}$, hence we can write $X_t = e^{-t} X_0 + \left(1 - e^{-2t}\right)^{1/2} Z_t$, where $Z_t \sim N(0, I_d)$.
  (3) $N(0, I_d)$ is an invariant measure of OU process.
- In practice, one may consider the time-rescaled OU process : $d\bar{X}_t = -g(t)^2 \bar{X}_t dt + \sqrt{2}\, g(t)\, dB_t$, with a positive smooth function $g : \mathbb{R}_+ \to \mathbb{R}_+$.

Diffusion Model: Reverse Process

In general, suppose we have an SDE of the form

$$dX_t = a(X_t, t)dt + b_t dB_t.$$

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

$$X_t^{\leftarrow} := X_{T-t}, \quad \text{for } t \in [0, T],$$

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

$$dX_t^{\leftarrow} = a^{\leftarrow}(X_t^{\leftarrow}, t)dt + b_{T-t}dW_t, \quad X_0^{\leftarrow} \sim q_T,$$

where W_t is the reversed Brownian motion, for simplicity, we don't distinguish B_t and W_t. We need to choose the reverse drift $a^{\leftarrow}(x, T-t)$ such that

$$\boxed{\mathrm{Law}(X_t^{\leftarrow}) = \mathrm{Law}(X_{T-t})}$$

By **Fokker-Planck equation**, we can choose

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

Diffusion Model: Reverse Process and Score Matching

Applying the result to the forward process, we obtain the reverse process in DDPM

Reverse Process

$$dX_t^{\leftarrow} = \left[X_t^{\leftarrow} + 2\nabla \log q_{T-t}(X_t^{\leftarrow})\right] dt + \sqrt{2}\, dB_t, \quad X_0^{\leftarrow} \sim q_T, \qquad \text{(reverse process)}$$

where $(B_t)_{t\in[0,T]}$ is the reversed Brownian motion.

Since q_0 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 of q_t at any time $t \in [0, T]$ via the samples.

Diffusion Model: Score Matching

By the properties of OU process, we know for any given t, X_t can be written as a linear combination of X_0 and independent noise

$$X_t = e^{-t} X_0 + \left(1 - e^{-2t}\right)^{1/2} Z_t, \quad Z_t \sim N(0, I_d).$$

By the theorem of Donising Score Matching, the score matching problem for X_t

$$\min_{s_t \in \mathcal{F}} \mathbb{E} \left\| \nabla \log q_t(X_t) - s_t(X_t) \right\|_2^2$$

is equivalent to

$$\min_{s_t \in \mathcal{F}} \mathbb{E} \left[\left\| s_t(X_t) + \frac{1}{\sqrt{1 - e^{-2t}}} Z_t \right\|_2^2 \right],$$

where $\mathcal{F}$ could be, e.g., a class of neural networks.

Diffusion Model: Score Matching

The objective can be replaced with an empirical version and estimated on the basis of samples $x_0^{(1)}, \dots, x_0^{(N)}$ from q_0, leading to the finite-sample problem

$$\min_{s_t \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \left\| s_t\left(x_t^{(i)}\right) + \frac{1}{\sqrt{1 - \exp(-2t)}} z_t^{(i)} \right\|^2, \tag{DDPM-SM}$$

where $(z_t^{(i)})_{i \in [N]}$ are i.i.d. standard Gaussian samples independent of the data $(x_0^{(i)})_{i \in [N]}$.

Hence, we can learn the score of q_t for all $t \in [0, T]$, and we assume

$$\mathbb{E}[\|s_t(X_t) - \nabla \log q_t(X_t)\|_2^2] \leq \varepsilon_{\text{score}}^2.$$

Diffusion Model

Forward process:

$$dX_t = -X_t dt + \sqrt{2}\, dB_t, \quad X_0 \sim q_0 := \mu$$

Reverse process:

$$dX_t^{\leftarrow} = \left[X_t^{\leftarrow} + 2\nabla \log q_{T-t}(X_t^{\leftarrow})\right] dt + \sqrt{2}\, dB_t, \quad X_0^{\leftarrow} \sim q_T \approx \gamma^d := \text{law of } N(0, I_d),$$

Partition : Partition the interval $[0, T]$ to $[kh, (k+1)h]$, $k = 0, 1, \cdots, K-1$ with $h > 0$, $K = T/h$. Integrate the reverse process from $[kh, t]$, $t \in [kh, (k+1)h]$,

$$X_t^{\leftarrow} = X_{kh}^{\leftarrow} + \int_{kh}^{t} X_s^{\leftarrow} ds + \int_{kh}^{t} 2 \underbrace{\nabla q_{T-s}(X_s^{\leftarrow})}_{\approx s_{T-kh}(X_{kh}^{\leftarrow})} ds + \sqrt{2}(B_t - B_{kh}).$$

Score matching and integral approximation:

$$dX_t^{\leftarrow} = \left\{ X_t^{\leftarrow} + 2\, s_{T-kh}(X_{kh}^{\leftarrow}) \right\} dt + \sqrt{2}\, dB_t, \qquad t \in [kh, (k+1)h],$$

which is a linear SDE and can be integrated in closed form.

Diffusion Model: Convergence Analysis

Let $p_t := \mathrm{Law}(X_t^{\leftarrow})$, DDPM has mainly three types of errors.

1. The error made at initialization of reverse process, γ^d used instead of q_T.
2. The score matching error, which on the sample size N, the size of the function class $\mathcal{F}$ and its closeness to the true score function.
3. The discretization of the reverse process, which depends on the step-size h.

Assumption 1 (Lipschitz score)

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

Assumption 2 (Second moment bound)

Assume that $M_2^2 := \mathbb{E}_{X \sim q_0} \|X\|_2^2 < \infty$.

Assumption 3 (Score estimation error bound)

For $k = 1, \ldots, K$,

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

Diffusion Model: Convergence Analysis

Theorem ([Chen et al., 2023])

Under the three previous assumptions. Let p_T be the output of the DDPM algorithm at time $T > 0$, with $h = T/K$ and K the number of steps, suppose $h \lesssim 1/L$, then

$$d_{\mathrm{TV}}(p_T, q_0) \lesssim \underbrace{\sqrt{\mathrm{KL}(q_0 \,\|\, \gamma^d)}\, e^{-T}}_{\textit{convergence of forward process}} + \underbrace{\left(L\sqrt{d\,h} + LM_2 h\right)\sqrt{T}}_{\textit{discretization error}} + \underbrace{\epsilon_{\mathrm{score}}\sqrt{T}}_{\textit{score estimation error}} .$$

Remark:

1. Unlike Langevin algorithms, this theorem does not assume any type of “bottleneck” condition such as log concave target distribution. **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 $\mathrm{KL}(q \,\|\, \gamma^d)$ might be large (even exponentially in dimension d), the contraction of the forward process creates a $\exp(-T)$ term which can make the first term small.

References I

Chen, S., Chewi, S., Li, J., Li, Y., Salim, A., and Zhang, A. R. (2023).
Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions.

Dyer, M. E. and Frieze, A. M. (1988).
On the complexity of computing the volume of a polyhedron.
SIAM Journal on Computing, 17(5):967–974.

Jia, H., Laddha, A., Lee, Y. T., and Vempala, S. (2021).
Reducing isotropy and volume to kls: an o*(n 3 ψ 2) volume algorithm.
In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 961–974.

Scott, D. W. (2015).
Multivariate density estimation: theory, practice, and visualization.
John Wiley & Sons.

References II

Song, Y. and Ermon, S. (2019).
Generative modeling by estimating gradients of the data distribution.
Advances in neural information processing systems, 32.

Song, Y. and Ermon, S. (2020).
Generative modeling by estimating gradients of the data distribution.

Tsybakov, A. (2009).
Introduction to nonparametric estimation. springer series in statistics. springer, new york.

Thanks!