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

Lecture 1 – Introduction

Lecturer: Yuansi Chen *Spring 2024*

Key concepts:

- Sampling: two settings
- Sampling in low dimension
- Curse of dimensionality

1.1 Introduction to sampling

We consider the problem of drawing a sample from a probability measure μ in n dimensions:

$$X \sim \mu(dx)$$

where μ is a probability measure on $\mathbb{R}^n$. By an abuse of notation, we also use μ as its density.

We are interested in two main settings.

1.1.1 Setting 1. μ is given in explicit form

In this setting, μ is given by an explicit mathematical expression, up to a normalization constant. It means that we can have access to the probability ratio between two points $\frac{\mu(x_1)}{\mu(x_2)}$ or the gradient of the logarithmic density $\nabla \log \mu(x)$. We often encounter this setting in approximate computation, Bayesian statistics and statistical physics.

Example 1 (Setting 1 in approximate computation)**.** *Given a bounded convex set K on $\mathbb{R}^n$ specified with a membership oracle, where one can access whether any point x is inside K or not, we would like to know the volume of the convex set.*

[DF88, Kha89] show that under the membership oracle, exact volume computation is NP-hard. Even when we are allowed to use a stronger oracle (the separation oracle, where not only we know if any point x is inside K or not, when the point is outside, we also know a hyperplane separating x and K), [Ele86] shows that every deterministic algorithm has to query q times to make a relative error of $\sqrt{\frac{2^n}{q}}$.

The situation is different when we are allowed to use randomized algorithms and are satisfied with results that hold with high probability. The state-of-the-art algorithm [JLLV21] for approximate volume computation achieves constant error with $O(n^3)$ complexity. It builds upon a sampling algorithm to sample from the uniform distribution over the convex set

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

Example 2 (Setting 1 in Bayesian linear regression)**.** *Consider a standard linear regression problem, in which we observe N data points $(x_i, y_i), i = 1, \ldots, N$, and we would like to fit a linear model. In the linear model, we assume*

$$y_i = x_i^\top \beta + \epsilon_i,$$

where $\beta \in \mathbb{R}^p$ is the true parameter, and the ϵ_i are independent and identically distributed (i.i.d.) random variables $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ (σ^2 is given for simplicity). The maximum likelihood principle provides an estimate of β based on the data. Assuming x_i are fixed, the likelihood is

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{X}, \beta) &= \prod_{i=1}^{N} p(y_i \mid x_i, \beta) \\ &\propto \prod_{i=1}^{N} e^{-\frac{(y_i - x_i^\top \beta)^2}{2\sigma^2}} \end{aligned}$$

where $\mathbf{y}$ is N-vector $\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$, $\mathbf{X}$ is the $n \times p$ design matrix with each row being $x_i^\top$. Then the maximum likelihood estimator (MLE) gives

$$\hat{\beta}_{MLE} = \arg\max_{\beta} p(\mathbf{y} \mid \mathbf{X}, \beta).$$

The MLE approach is a frequentist approach and obtaining MLE is usually cast as an optimization problem. In a Bayesian approach, the data are supplemented with a prior belief about the true parameters. Then we are interested in quantify the uncertainty in β estimation given the prior belief about the parameters (see e.g. [Hof09]). Applying the Bayes rule, the posterior distribution over β given prior $p_{prior}(\beta)$ can be expressed as

$$p_{posterior}(\beta \mid \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y} \mid \mathbf{X}, \beta) p_{prior}(\beta)$$

The main goal of Bayesian inference is to characterize the posterior distribution $p_{posterior}(\beta \mid \mathbf{y}, \mathbf{X})$, which is usually done by first sampling the posterior distribution.

1.1.2 Setting 2. μ is given with a collection of i.i.d. samples

In this setting, we don't know the explicit form of μ, but we are given a collection of i.i.d. samples $x_1, \ldots, x_N \overset{\text{i.i.d.}}{\sim} \mu$. This setting is also called "generative modeling" in machine learning. Perhaps the most popular problem of this kind nowadays is the generative modeling of natural images.

Example 3 (Generative modeling of natural images)**.** *Given N cat pictures of size 200×200 pixels each, we are interested in drawing a new picture of a cat from the underlying distribution. In this case, μ is a distribution supported on $\mathbb{R}^{200\times 200}$, and is only given through N i.i.d. samples.*

Take a look at what people can achieve in natural image generation in the past. Compare with what we can do with DALL-E nowadays.

- *Deep Belief Nets in 2006 [HOT06]*
- *Generative Adversarial Nets in 2014 [GPAM+14]*
- *Denoising Diffusion Probabilistic Models in 2020 [HJA20].*

1.2 Reducing the problem in Setting 2 to Setting 1 via density estimation

Although not necessary, we can try to solve the problem in Setting 2 in two steps: first, we estimate an explicit expression of μ from the i.i.d. samples $x_1, \ldots, x_N$; second, we sample from the explicit expression of μ as we do in Setting 1.

The first step is also called *density estimation* in the statistics literature. One of the simplest density estimation method is simply plotting a histogram of the N data points with a fixed small bandwidth parameter. While we discuss several methods as we need, providing a complete overview of density estimation is out of the scope. Interested readers are referred to [Sco15] for an overview.

1.3 Both problems are not difficult in low dimension

Before we delve into sophisticated methods in modern sampling, we show that both the density estimation problem and the sampling problem from an explicit density can be solved via standard techniques in low dimension. By low dimension, we roughly mean that the dimension is much smaller than the sample size, satisfying dimension $n \leq 5$.

We go through a few simple methods for density estimation and sampling. The purpose of the section is to remind us that standard techniques exist and work well if we are just dealing with low dimensional problems.

1.3.1 One dimensional density estimation and sampling

We observe $N = 1000$ i.i.d. samples $x_1, \ldots, x_N \sim \mu$. μ is supported on $\mathbb{R}$. Can we estimate the density? Can we draw samples from μ?

Histogram or approximate the density by piece-wise constant functions

Let $\tau \in \mathbb{N}$ be the number of bins. Create τ uniformly placed bins $(B_i)_{i\in[\tau]}$ in the range of data points. Let

$$p_i = \frac{\sum_{j=1}^{N} \mathbf{1}_{x_j \in B_i}}{N}.$$

We can draw a histogram with count frequencies p_i.

Direct sampling via inverse CDF transform

Let $h : \mathbb{R} \to [0, 1]$ be the cumulative density function (CDF) of μ and suppose the inverse function h^{-1} is known. We can then sample X from μ via the following procedure:

1. Sample U from $U[0, 1]$, the uniform distribution over $[0, 1]$.
2. Output $X = h^{-1}(U)$.

It can be easily proved that $X \sim \mu$ because for any $t \in \mathbb{R}$ we have that

$$\begin{aligned}\mathbb{P}(X \le t) &= \mathbb{P}\left(h^{-1}(U) \le t\right) \\ &= \mathbb{P}(U \le h(t)) = h(t).\end{aligned}$$

This method is exact and is highly efficient when h^{-1} can be easily computed. Additionally, if $\mu(x) > 0, \forall x \in \mathbb{R}$, then h is increasing and h^{-1} always exists. It can also be computed up to numerical errors via the bisection method.

As an example, in the case where μ is a histogram with τ-bins of probability p_i. The inverse CDF transform sampling works as follows

- From the probabilities $p_1, \ldots, p_\tau$, compute the cumulative distribution

  $$F_i = \sum_{k=1}^{i} p_i$$

  for all $i \in [\tau]$
- Draw a uniform random number $U \sim [0, 1]$
- Get the smallest i such that $U \le F_i$. Return any number in the bin B_i.

1.3.2 Low-dimension density estimation and sampling

For $n \geq 2$ but small, histogram still works. More generally, if one wants a smooth density, kernel density estimation is often used.

Kernel density estimation

The density function μ at a point x can be represented as

$$\mu(x) = \lim_{h \to 0} \frac{1}{2h} \mathbb{P}_{X \sim \mu}(x - h < X \leq x + h).$$

We can replace this probability with its numerical estimate

$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{x-h<x_i \leq x+h}.$$

It results in the estimator

$$\hat{\mu}(x) = \frac{1}{Nh} \sum_{i=1}^{n} w\left(\frac{x - x_i}{h}\right),$$

where w is the weight function

$$w(x) = \begin{cases} 1/2 & \text{if } |x| \leq 1, \\ 0 & \text{otherwise .} \end{cases}$$

In general, we don't have pick this weight function, we can use a smoother kernel function $K(\cdot)$. Then we have the definition of the kernel density estimator

$$\hat{\mu}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right),$$
$$K(x) \geq 0, \int_{-\infty}^{\infty} K(x) = 1.$$

One popular choice is the Gaussian kernel

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}.$$

The above idea can be extended to multivariate data by simply switching to multivariate kernel functions. It takes the form

$$\hat{\mu}_{\text{multi}}(x) = \frac{1}{Nh^n} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right).$$

The multivariate Gaussian kernel is

$$K(u) = \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}u^\top u\right).$$

Sampling from $\hat{\mu}_{\text{multi}}$ is not difficult as it is a mixture of simple distributions. We can first draw an index uniformly from $\{1, 2, \ldots, N\}$ and then draw from a simple distribution specified by the kernel function. In general, the sampling of low-dimensional density can be done via rejection sampling.

Rejection sampling

We want to sample from μ on $\mathbb{R}^n$ (with access to unnormalized density $\tilde{\mu}$). Suppose we can sample from an easier measure ν (e.g. uniform on a cube or Gaussian). Let $\tilde{\nu}$ be an unnormalized version of ν such that $\tilde{\nu} \geq \tilde{\mu}$. $\tilde{\nu}$ is also called an upper envelope of $\tilde{mu}$. Then repeat until acceptance:

1. Draw $X \sim \nu$
2. Accept X with probability $\frac{\tilde{\mu}(X)}{\tilde{\nu}(X)}$

We claim that the output of rejection sampling is a sample drawn exactly from μ. Also, the number of samples drawn from ν until a sample is accepted follows a geometric distribution with mean Z_μ/Z_ν, where $Z_\nu := \int \tilde{\nu}$ and $Z_\mu := \int \tilde{\mu}$.

To show that the output X of rejection sampling is drawn exactly according to μ, let $(U_i)_{i=1}^\infty \overset{\text{i.i.d.}}{\sim}$ uniform $[0,1]$ and $(X_i)_{i=1}^\infty \overset{\text{i.i.d.}}{\sim} \nu$ be independent. Then, for any event A,

$$\begin{aligned}
\mathbb{P}(X \in A) &= \sum_{\tau=0}^{\infty} \mathbb{P}\left(X_{\tau+1} \in A, U_i > \frac{\tilde{\mu}(X_i)}{\tilde{\nu}(X_i)}, \forall i \in [\tau], U_{\tau+1} \leq \frac{\tilde{\mu}(X_{\tau+1})}{\tilde{\nu}(X_{\tau+1})}\right) \\
&= \sum_{\tau=0}^{\infty} \mathbb{P}\left(X_{\tau+1} \in A, U_{\tau+1} \leq \frac{\tilde{\mu}(X_{\tau+1})}{\tilde{\nu}(X_{\tau+1})}\right) \mathbb{P}\left(U_1 > \frac{\tilde{\mu}(X_1)}{\tilde{\nu}(X_1)}\right)^\tau \\
&= \sum_{\tau=0}^{\infty} \left(\int_A \frac{\tilde{\mu}}{\tilde{\nu}} \mathrm{d}\nu\right) \left(\int \left(1 - \frac{\tilde{\mu}}{\tilde{\nu}}\right) \mathrm{d}\nu\right)^\tau \\
&= \frac{Z_\mu}{Z_\nu} \mu(A) \sum_{\tau=0}^{\infty} \left(1 - \frac{Z_\mu}{Z_\nu}\right)^\tau \\
&= \mu(A).
\end{aligned}$$

Rejection sampling provides an exact sampling algorithm is one is willing to wait for the number of steps that follows a geometric distribution. However, in practice, if one has to truncate the number of steps to a finite number due to a computational time budget, we would only get approximate sampling.

1.4 Curse of dimensionality

The curse of dimensionality can exhibit in both the density estimation problem and the sampling problem from an explicit density. In the former case, the curse of dimensionality happens when the data size N needs to be at least 2^n in order to estimate the density with a constant error. In the later case, the curse of dimensionality happens when the sampling algorithm takes at least 2^n computational time in order to produce a sample close to μ.

1.4.1 Curse of dimensionality in density estimation

In practice, multivariate kernel density estimation is often restricted to dimension $n \leq 5$. The reason is, that a higher dimensional space will be only very sparsely populated by data points. Or in other words, there will be only very few neighboring data points to any value x in a higher dimensional space, unless the sample size is extremely large.

More specifically, we can recall the classical minimax bounds for density estimation (see e.g. [Tsy09] or Yihong Wu's lecture notes). We say a probability density function (pdf) f *belongs to* $\mathcal{P}_\beta$ with a smooth parameter $\beta > 0$ if

- f is a pdf on $[0,1]^n$ and is upper bounded by a constant, say, 2.
- $f^{(m)}$ is α-Hölder continuous, i.e.

$$\left|f^{(m)}(x) - f^{(m)}(y)\right| \leq |x-y|^\alpha, \forall x, y \in [0,1]^n,$$

  where $\alpha \in (0,1], m \in \mathbb{N},$ and $\beta = \alpha + m$.

For example, when $\beta = 1$, $\mathcal{P}_1$ is simply the set of pdfs which are Lipschitz and bounded by 2.

Theorem 1.4.1 (Minimax risk lower bound on density estimation)**.** *Given N i.i.d. samples $x_1, \ldots, x_N$ from a pdf $f \in \mathcal{P}_\beta$, the minimax risk of an estimation $\hat{f}$ of f under the quadratic loss function $\ell(\hat{f}, f) := \left\|\hat{f} - f\right\|_2^2 = \int_{[0,1]^n} (f(x) - \hat{f}(x))^2 dx$ satisfies*

$$\inf_{\hat{f}} \sup_{f \in \mathcal{P}_\beta} \left\|\hat{f} - f\right\|_2^2 \gtrsim N^{-\frac{2\beta}{n+\beta}}.$$

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

In words, for $\beta = 1$, the number of data points N needs to be as large as 2^{n+1} in order to achieve a quadratic loss of order $\frac{1}{4}$.

Proving the above theorem is out of scope of this course. We present an intuitive proof only for $\beta = 1$. For each data point, draw a small cube with width δ centered

around it, with $0 < \delta \ll 1$. Choose δ such that $1/4 \leq N\delta^n \leq 1/2$. Then the N cubes centered around N data points cover at most $1/2$ of the total volume. In the place where it is not covered, any point is at least $\delta/2$ away from a data point. We don't have any information the true pdf f except the information from its nearest data point via the Lipschitz assumption.

$$|f(x) - f(y)| \leq |x - y| .$$

By designing f adversarially, we make an error of $\delta/2 * 2$ for any x in the part which is not covered. Hence, the minimax risk is lower bounded as

$$\inf_{\hat{f}} \sup_{f \in \mathcal{P}_\beta} \left\| \hat{f} - f \right\|_2^2 \gtrsim \delta^2 \frac{1}{2} \overset{(i)}{\gtrsim} N^{-\frac{2}{n}},$$

where (i) uses the bound $1/4 \leq N\delta^n$. The bound is loose but also illustrates the curse of dimensionality.

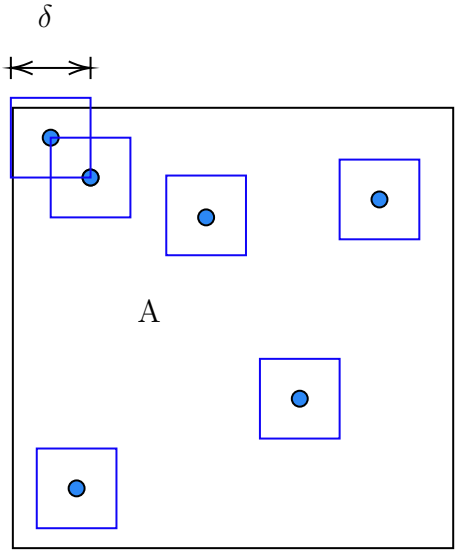

Figure 1.1. Illustration of the partition of the unit cube. Each smaller cube is of width δ. Each blue point represents a data point. A is the space which is not covered. In A, since any data point is at least $\frac{\delta}{2}$ away, the error we make inside A is of order $\delta/2 * 2$.

1.4.2 Curse of dimensionality in rejection sampling

In high dimension, the ratio Z_ν/Z_μ will be very close to 0. Here is an example. Take the unnormalized target density $\tilde{\mu} = \mathbf{1}_{B_2}$, where $B_2 := \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$ is the ℓ_2-unit ball in $\mathbb{R}^n$. Take $\tilde{\nu} = \mathbf{1}_{B_\infty}$, where $B_\infty := \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\}$ is ℓ_∞-unit ball. Since

$\|\cdot\|_2 \geq \|\cdot\|_\infty$, we verify that $\tilde{\nu}$ is an upper envelope $\tilde{\nu} \geq \tilde{\mu}$. Calculate the ratio Z_ν/Z_μ via formula for the volume of an n-ball, we obtain

$$Z_\mu/Z_\nu = \frac{\int \tilde{\mu}}{\int \tilde{\nu}} = \frac{\frac{2(2\pi)^{(n-1)/2}}{n!!}}{2^n} \approx \frac{1}{\sqrt{n\pi}} \left(\frac{\pi e}{2n}\right)^{n/2},$$

where the last approximation is done via Stirling's approximation. As n grows, this rate can be much smaller than 2^{-n}. As a consequence, the rejecting sampling of μ via ν takes at least 2^n iterations.

Bibliography

[DF88] Martin E. Dyer and Alan M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM Journal on Computing*, 17(5):967–974, 1988.

[Ele86] György Elekes. A geometric inequality and the complexity of computing volume. *Discrete & Computational Geometry*, 1:289–292, 1986.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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

[Hof09] Peter D Hoff. *A first course in Bayesian statistical methods*, volume 580. Springer, 2009.

[HOT06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[JLLV21] He Jia, Aditi Laddha, Yin Tat Lee, and Santosh Vempala. Reducing isotropy and volume to KLS: an $O(n^3\psi^2)$ volume algorithm. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 961–974, 2021.

[Kha89] Leonid Genrikhovich Khachiyan. The problem of calculating the volume of a polyhedron is enumerably hard. *Russian Mathematical Surveys*, 44(3):199, 1989.

[Sco15] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[Tsy09] Alexandre B Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.