Machine Learning

Assignment 7

CHANG Yung-Hsuan (張永璿)

111652004

eiken.sc11@nycu.edu.tw

October 20, 2025

1. Explain the concept of score matching and describe how it is used in score-based (diffusion) generative models.

   Explanation. The core idea of score matching is to learn the shape of a probability distribution without computing its normalization constant. Instead of estimating $p(x)$ itself, we learn its score function $S(x) \coloneqq \nabla_x \log p(x)$, which points toward directions of higher probability density. If the model can reproduce this vector field accurately, it essentially captures the geometry of the data distribution.

   a. Intuition behind the explicit, implicit, and denoising score matching

   - *Explicit score matching.*

     We start with the most direct idea: make our model $S(x;\theta)$ imitate the true score $\nabla_x \log p(x)$. The loss function

     $$L_{\text{ESM}} = \mathbb{E}_{x\sim p(x)} \| S(x;\theta) - \nabla_x \log p(x) \|^2$$

     measures how close they are.

     This is an intuitive modeling: it treats the true score as the target and training data via supervised learning. However, since $p(x)$ is unknown, we usually cannot compute $\nabla_x \log p(x)$ in practice.

   - *Implicit score matching.*

The implicit score matching starts from the same principle but uses integration by parts to remove the dependence on $p(x)$. This gives a new loss that only depends on the model $S(x;\theta)$ and its divergence, allowing learning directly from data samples with the loss function

$$L_{\text{ISM}} = \mathbb{E}_{x\sim p(x)}\|S(x;\theta)\|^2 + 2\nabla_x \cdot S(x;\theta).$$

The key idea is that even **without** knowing $p(x)$, we can still find parameters θ (the same as in ESM) that produce the correct score field.

- *Denoising score matching.*

  The denoising score matching adds small Gaussian noise to data points:

  $$x = x_0 + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right).$$

  The model $S_\sigma(x;\theta)$ then learns to predict the direction **back** to the clean data.

  Because the target is tildeply how to "denoise" a perturbed point, the model can be trained directly using pairs of (x, x_0), much like supervised learning.

  As a result, the denoising score matching avoids the analytical difficulties of the explicit score matching (which requires knowing $p(x)$) and the indirect optimization of the implicit score matching (which needs integration by parts), while still learning the same parameter as in ESM or ISM.

  Intuitively, the denoising score matching teaches the network how to reverse the effect of noise; or equivalently, how to move random samples toward regions of higher probability density.

- *Sliced score matching.*

  The implicit score matching involves a computationally expensive term $\nabla_x \cdot S(x;\theta)$, which scales poorly in high dimensions. The sliced score matching **reduces** this cost using Hutchinson's trace estimator,

  $$\text{tr}(A) = \mathbb{E}_{v\sim p(v)}\left(v^T A v\right),$$

for any random vector v satisfying $E\left(vv^T\right) = I$.

Using this, the trace term $\text{tr}(\nabla_x S(x;\theta))$ can be rewritten as

$$\text{tr}(\nabla_x S(x;\theta)) = \mathbb{E}_{v\sim p(v)}\left(v^T \nabla_x \left(v^T S(x;\theta)\right)\right),$$

giving the sliced score matching loss

$$L_{\text{SSM}(\theta)} = \mathbb{E}_{x\sim p(x)}\|S(x;\theta)\|^2 + \mathbb{E}_{x\sim p(x)}\mathbb{E}_{v\sim p(v)}\left(2v^T \nabla_x \left(v^T S(x;\theta)\right)\right).$$

This avoids explicit computation of the Hessian of $\log p(x;\theta)$ and makes score matching feasible even in high-dimensional settings such as image generation with producing the same parameter, in ESM, ISM, or DSM, in expectation.

ISM looks at the divergence of the score field directly (summing over all coordinate directions), and SSM looks at many random slices of that divergence and averages them; the expectation over all slices is exactly the same as the full divergence.

b. How score matching is used in diffusion models

Score-based generative models extend denoising score matching to a **continuous sequence** of noise levels. They define a forward diffusion process that gradually adds Gaussian noise to the data, forming a Markov chain,

$$p(x_0) \longrightarrow p(x_1) \longrightarrow \cdots \longrightarrow p(x_T),$$

where $p(x_T)$ becomes nearly Gaussian.

The model learns a score function $S_\sigma(x_t;\theta)$ for each noise level σ_t, indicating how to reverse the noise. During generation, we start from random noise $x_T \sim \mathcal{N}(0, I)$ and iteratively **reverse** the diffusion, guided by the learned scores, gradually reducing noise to obtain realistic samples.

In short, score matching teaches the model the gradient of $\log p(x)$, while diffusion models apply denoising (or sliced) score matching across multiple noise scales to learn how to reverse the diffusion process and generate data.

2. There are unanswered questions from the lecture, and there are likely more questions we haven't covered. Take a moment to think about these questions. Write down the ones you find important, confusing, or interesting.

Answer. In the denoising score matching, the data are assumed to follow

$$y_i = f_\theta(x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}\big(0, \sigma^2 I\big).$$

Typically, we fix σ when fitting the model, but in practice, σ controls how much noise we assume in the data. A small σ means we trust the data more, leading to possible overfitting, while a large σ means we treat the data as noisier, often resulting in smoother and more generalizable models.

This made me wonder: instead of fixing a single σ, could we take a range of σ values and average the resulting models? Intuitively, this would mean combining models trained under different assumptions about data noise, which resembles the idea of *Bayesian model averaging*—integrating over different possible noise levels rather than committing to one.

Mathematically, this could be viewed as

$$\hat{f}(x) = \frac{1}{N_\sigma} \sum_{j=1}^{N_\sigma} f_{\theta(\sigma_j)}(x),$$

where each $f_{\theta(\sigma_j)}$ is a model trained under a different assumed noise variance σ_j^2.

Conceptually, this approach might yield a more robust estimate of $f_\theta(x)$, especially when the true noise level is uncertain or varies across the input space. I find this question meaningful because it connects the practical side of the denoising score matching, where we can directly control the level of noise, with theoretical ideas from probabilistic modeling and Bayesian inference.