ML Week8 Assignment

WANG TZU YI

October 28, 2025

1. From Implicit Score Matching (ISM) to Sliced Score Matching (SSM)

We start from the Implicit Score Matching (ISM) objective (Hyvärinen, 2005):

$$\mathcal{L}_{\text{ISM}}(\theta) = \mathbb{E}_{x\sim p(x)}\Big[\, \|\nabla_x s_\theta(x)\|_F^2 + 2\,\text{tr}\big(\nabla_x s_\theta(x)\big)\Big], \tag{1}$$

where $s_\theta(x) = \nabla_x \log p_\theta(x)$ denotes the model score function, and $\|\cdot\|_F$ is the Frobenius norm.

Step 1. Trace identity

For any square matrix $A \in \mathbb{R}^{d\times d}$ and a random vector $v \sim \mathcal{N}(0, I_d)$ (or any isotropic distribution with $\mathbb{E}[vv^\top] = I_d$), we have

$$\mathbb{E}_v\big[v^\top A v\big] = \text{tr}(A). \tag{2}$$

Proof:

$$\mathbb{E}_v[v^\top A v] = \mathbb{E}_v[\text{tr}(v^\top A v)] = \mathbb{E}_v[\text{tr}(A v v^\top)] = \text{tr}(A\,\mathbb{E}_v[vv^\top]) = \text{tr}(A).$$

Step 2. Replace the trace term in ISM

Using the identity above, the ISM objective can be rewritten as

$$\begin{aligned} \mathcal{L}_{\text{ISM}}(\theta) &= \mathbb{E}_{x\sim p(x)}\Big[\, \|\nabla_x s_\theta(x)\|_F^2 + 2\,\mathbb{E}_{v\sim\mathcal{N}(0,I)}\big[v^\top(\nabla_x s_\theta(x))v\big]\Big] && (3)\\ &= \mathbb{E}_{x\sim p(x)}\mathbb{E}_{v\sim\mathcal{N}(0,I)}\Big[\, \|\nabla_x s_\theta(x)\|_F^2 + 2\,v^\top(\nabla_x s_\theta(x))v\Big]. && (4) \end{aligned}$$

Step 3. Replace the Frobenius norm by an isotropic expectation

Similarly, for the Jacobian $J = \nabla_x s_\theta(x)$, we have the identity

$$\mathbb{E}_{v\sim\mathcal{N}(0,I)}\left[\|v^\top J\|^2\right] = \|J\|_F^2, \tag{5}$$

because

$$\mathbb{E}_v[\|v^\top J\|^2] = \mathbb{E}_v[v^\top J J^\top v] = \mathrm{tr}(JJ^\top) = \|J\|_F^2.$$

Step 4. Substitute to obtain the Sliced Score Matching form

Plugging this result back into the ISM objective, we obtain

$$\mathcal{L}_{\mathrm{SSM}}(\theta) = \mathbb{E}_{x\sim p(x)}\mathbb{E}_{v\sim\mathcal{N}(0,I)}\Big[\,\|v^\top s_\theta(x)\|^2 + 2\,v^\top(\nabla_x s_\theta(x))v\Big]. \tag{6}$$

2. Stochastic Differential Equations in Score-Based Generative Models

2.1 From ODE to SDE

An ordinary differential equation (ODE) describes a deterministic system:

$$\frac{dy_t}{dt} = f(t, y_t),$$

where the state y_t evolves deterministically according to the drift function f.

A stochastic differential equation (SDE) extends this by adding a stochastic term:

$$dy_t = f(t, y_t)\,dt + g(t, y_t)\,dW_t,$$

where W_t is a Wiener process (Brownian motion). The term $f(t, y_t)$ is called the *drift*, representing the mean tendency, and $g(t, y_t)dW_t$ is the *diffusion*, introducing random noise.

Thus, while an ODE produces a single deterministic trajectory, an SDE defines a *stochastic process* — a collection of random trajectories whose distribution evolves over time.

2.2 Why Use SDEs in Score-Based Models?

In score-based generative modeling, the idea is to gradually transform the data distribution $p_{\text{data}}(x)$ into a simple Gaussian noise distribution via a forward noising process, and then learn to reverse this process to generate data.

This forward diffusion process can be formulated as an SDE:

$$dx = f(x,t)\,dt + g(t)\,dW_t,$$

where f and g determine how noise is added to the data as time increases.

2.3 Reverse-Time SDE

According to Anderson (1982), every forward SDE

$$dx = f(x,t)\,dt + g(t)\,dW_t$$

has a corresponding reverse-time SDE that describes how to revert this diffusion process:

$$dx = \big[f(x,t) - g(t)^2 \nabla_x \log p_t(x)\big]\,dt + g(t)\,d\bar{W}_t,$$

where $\bar{W}_t$ is a reverse-time Wiener process, and $\nabla_x \log p_t(x)$ is the *score function* of the intermediate distribution $p_t(x)$.

In practice, this score function is unknown and is learned by a neural network $s_\theta(x,t)$ during training.

2.4 Generation via Reverse SDE

Once the score model $s_\theta(x,t)$ is trained, we can simulate the reverse SDE using numerical methods (such as the Euler–Maruyama method) to generate data.

$$x_{t-\Delta t} = x_t + \big[f(x_t,t) - g(t)^2 s_\theta(x_t,t)\big]\Delta t + g(t)\sqrt{|\Delta t|}\,z, \quad z \sim \mathcal{N}(0,I).$$

Starting from Gaussian noise $x_T \sim \mathcal{N}(0,I)$, we iteratively apply this update until $t=0$ to obtain samples from the data distribution.