Statistical Methods

Samy Braik

We denote by p the target distribution and q an easy-to-sample distribution, for example a centered Gaussian.
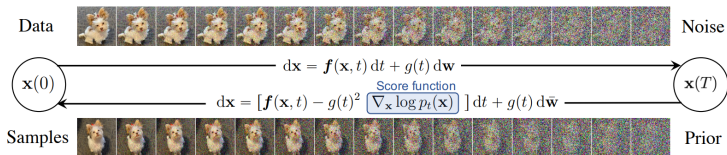
Diffusion

Let $X_0 \sim p$. We want to add noise until we reach pure noise, and denoise it afterward. We choose an horizon of time $T \in \mathbb{N}^*$ and a noise schedule $\beta : [0, T] \to \mathbb{R}^*$, continuous and non decreasing.

Forward process

$$d\overrightarrow{X}_t = \frac{-\beta(t)}{2\sigma^2}\overrightarrow{X}_t dt + \sqrt{\beta(t)}dB_t, \quad \overrightarrow{X}_0 \sim p \qquad (1)$$

Backward process

$$d\overleftarrow{X}_t = \left(\frac{\beta(T-t)}{2\sigma^2}\overleftarrow{X}_t + \beta(T-t)\nabla \log p_{T-t}\left(\overleftarrow{X}_t\right) \right) dt \qquad (2)$$
$$+ \sqrt{\beta(T-t)}dB_t, \quad \overleftarrow{X}_0 \sim p_T$$

We learn the score by using score-matching techniques

Score matching

$$\mathcal{L}_{\mathsf{score}}(\theta) = \mathbb{E}\left[\left\|s_\theta\left(\tau, \overrightarrow{X}_\tau\right) - \log p_\tau\left(\overrightarrow{X}_\tau | X_0\right)\right\|^2\right] \qquad (3)$$

where s_θ is a neural network approximating the score function.

Plug it in the backward process and generate by discretizing the dynamics.

Normalizing flow

Let $X_0 \sim q$ and $f : \mathbb{R}^d \to \mathbb{R}^d$ an invertible and differentiable function an set $X_1 := f(X_0)$ such that $X_1 \sim p$.
We can write the density of X_1 as

$$p_{X_1} = p_{X_0}(f^{-1}(x_1)) \left| \det \frac{\partial f^{-1}}{\partial x_1}(x_1) \right| \qquad (4)$$

$$= p_{X_0}(f^{-1}(x_1)) \left| \det \frac{\partial f}{\partial x_0}(f^{-1}(x_1)) \right|^{-1} \qquad (5)$$

We can then write the log-likelihood as

$$\log p_{X_1}(x_1) = \log p_{X_0}(f^{-1}(x_1)) - \log \left| \det \frac{\partial f}{\partial x_0}(f^{-1}(x_1)) \right| \qquad (6)$$

MINES
ParisTech | PSL★

Normalizing flow

Let $X_0 \sim q$ and $X_1 \sim p$. We want to learn f_θ such that $X_1 \simeq f_\theta(X_0) = Z \sim p_Z$. To do that, we set a structure on f_θ, with $f_1, \ldots, f_k$ simpler function (all parametrized by θ) such that

$$f_\theta = f_1 \circ f_2 \circ \ldots \circ f_k$$

We determine f_θ by minimizing

$$\mathcal{L}_{\mathsf{NF}}(\theta) = \mathbb{E}\left[-\log p_Z(f_\theta(x)) - \log \left|\det \frac{\partial f_\theta}{\partial x}(x)\right|\right]$$
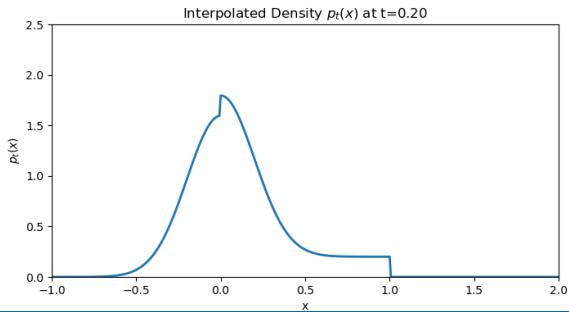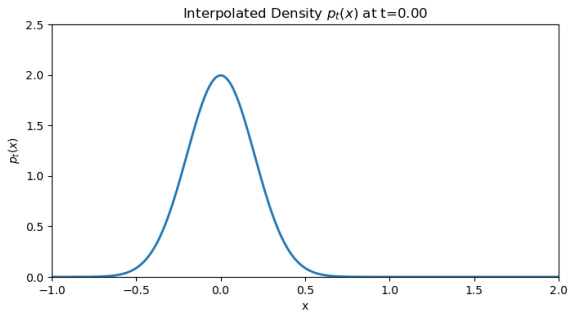
PSL★

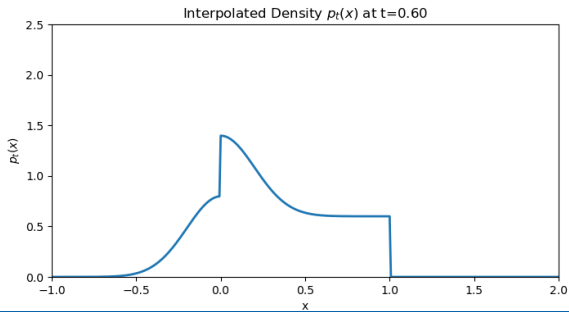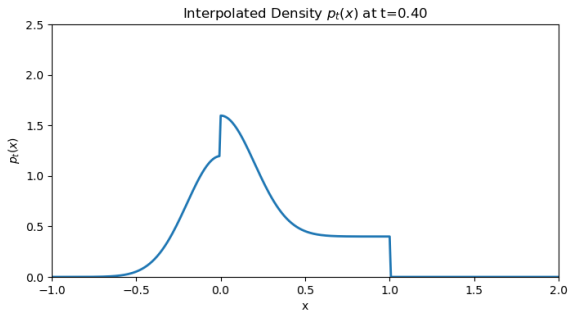# Flow matching

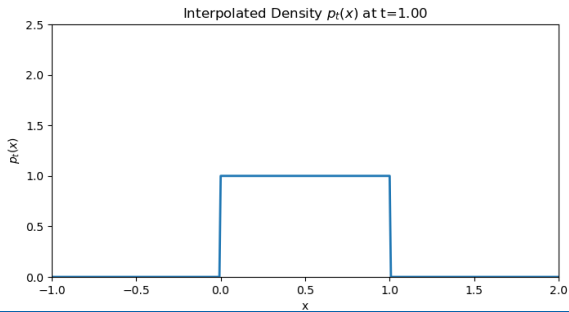We start by defining a probability density path

Probability density path

A probability path $p : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ meaning that for each time t, p_t is density function i.e. $\int p_t(x)dx = 1$.

A simple example of such a path is a path p interpolating two density p_0 and p_1 with $p_t = tp_1 + (1 - t)p_0$

Figure: A probability path interpolating $\mathcal{N}(0, 0.2)$ and $\mathcal{U}([0, 1])$

Interpolated Density $p_t(x)$ at t=0.00

Interpolated Density $p_t(x)$ at t=0.20

Interpolated Density $p_t(x)$ at t=0.40

Interpolated Density $p_t(x)$ at t=0.60

Interpolated Density $p_t(x)$ at t=0.80

Interpolated Density $p_t(x)$ at t=1.00

Flow matching
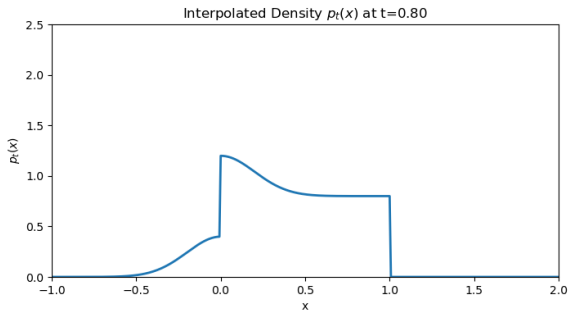
Next we introduce a core object, a time dependent vector field $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ which is used to construct a map $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, called a flow, by the following ODE

$$\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)) \tag{7}$$

$$\phi_0(x) = x$$

The link between the flow and the probability path is given by the change of variable formula

$$p_t(x) = q(\phi_t^{-1}(x)) \det \left[\frac{\partial \phi_t^{-1}}{\partial x}(x) \right] \tag{8}$$

That coincides with the normalizing flow case.

The link between the vector field and the probability path is given by the continuity equation

$$\frac{d}{dt}p_t(x) + \text{div}\left[v_t(x)p_t(x)\right] = 0 \tag{9}$$

It said that the vector field v_t generates the probability path p_t if it satisfies the continuity equation.
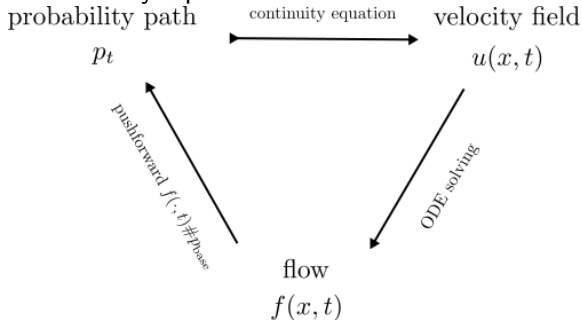

Figure: Link between marginal flow matching objects

Given a probability path p_t and a vector field v_t, the naïve flow matching loss is defined by

$$\mathcal{L}_{\mathsf{FM}}(\theta) = \mathbb{E}_{t, p_t(x)} \left[\|v_t^\theta(x)t - v_t(x)\|^2 \right] \tag{10}$$

but we don't have acces to v_t and p_t. To adress this problem, we introduce new objects.

Given a particular data sample x_1 from q, we introduce a conditional probability path $p_t(x|x_1)$ such that at time $t = 0$ $p_0(x|x_1) = q(x)$ and by marginalizing over x_1 we can recover the marginal probability path

$$p_t(x) = \int p_t(x|x_1)q(x_1)dx_1 \qquad (11)$$

Instead of working with a probability density, we consider samples from our distributions and define how to go from one to the other.

In the same vein, we can define a conditional vector field, assuming $p_t(x) > 0$ for all t and x

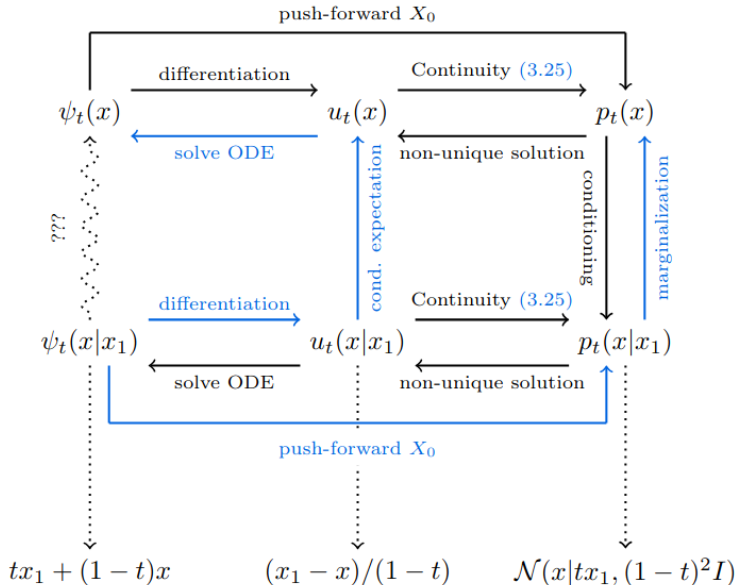$$v_t(x) = \int v_t(x|x_1) \frac{p_t(x|x_1)q(x_1)}{p_t(x)} dx_1 \qquad (12)$$

Figure: Link between all the flow matching objects

The new loss function, called conditional flow mathcing loss, is defined as

$$\mathcal{L}_{\mathsf{CFM}}(\theta) = \mathbb{E}_{t,p_t(x|x_1)} \left[\|v_t^\theta(x|x_1)t - v_t(x|x_1)\|^2 \right] \qquad (13)$$

with the property : $\mathcal{L}_{\mathsf{FM}}(\theta) = \mathcal{L}_{\mathsf{CFM}}(\theta)$ up to a constant independant of θ.

Flow matching

Algorithm Flow matching training

    Input: dataset p, noise q
    Initialized v^θ
    while not converged **do**
        $t \sim \mathcal{U}([0,1])$
        $x_1 \sim p(x_1)$
        $x_0 \sim q(x_0)$
        Gradient step with $\nabla_\theta \| v_t^\theta (x_t - \dot{x}_t{}^2)$
    Output: v^θ

PSL★

Flow matching

Algorithm Flow matching sampling

Input: Trained v^θ

$x_0 \sim q(x_0)$

Solve numerically the ODE $\dot{x}_t = v_t^\theta(x_t)$

Output: x_1

Comparison

Models	Pros	Cons
Diffusion	1.2	Only works with Gaussian
Normalizing flow	Exact density estimation	Computationally intensive
Flow matching	Exact density estimation Simulation free training	test
Kernel estimator	Flexible Easy to exploit	Slow rate of convergence Hard to evaluate at new data point Hard to choose tuning parameters Need a lot of data