Flow Matching for Generative Modeling

Pramook Khungurn

July 23, 2024

This note was written as I read the "Flow Matching for Generative Modeling" paper by Lipman et al. [5].

1 Background

- A data item is denoted by $x = (x^1, x^2, \ldots, x^d) \in \mathbb{R}^d$.
- A **probability density path** is a function $p : [0,1] \times \mathbb{R}^d \to \mathbb{R}^+ \cup \{0\}$ such that each $p(t, \cdot)$ is a probabilty density function on $\mathbb{R}^d$. In other words, it holds that

$$\int p(t,x)\,\mathrm{d}x = 1$$

  for all $t \in [0,1]$.
- For a time dependent function $f : [0,1] \times \mathbb{R}^d \to R$ for some range set R, we may write $f(t,x)$ as $f_t(x)$ to emphasize time dependence. Moreover, we can refer to $f_t : \mathbb{R}^d \to R$ as a function in its own right.
  - With this, we may say that p_t is a probability distribution on $\mathbb{R}^d$.
- A **time-dependent vector field** is a function $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$.
- Given a time dependent vector field v, its **flow** is another vector field $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ defined by the differential equation

$$\frac{\partial}{\partial t}\phi_t(x) = v_t(\phi_t(x)) \tag{1}$$

  and the initial condition $\phi_0(x) = x$.
  - In other words, $\phi_t(x)$ is the position at time t of the particle that starts at x at time 0 and follows the trajectory defined by taking v as the time-dependent vector field.
  - We say that v_t **generates** ϕ_t.
- Chen et al. proposed the **neural ordinary differential equation** model [1]. The idea is to model the vector field v with a neural network $v_t(x;\theta)$. We then train it so that ϕ_1 has the property that we want.
  - If you want a refresher on neural ODE, then read my previous note on it [2].
- A neural ODE can be used to transform a probability distribution to another. Say, we start with a probability distribution p_0 on $\mathbb{R}^d$. Then, we do the following.
  - Sample $x \sim p_0$.
  - Compute $x' = \phi_t(x)$ by integrating the neural ODE from 0 up to t.

Let us denote the probability density of x' by p_t. It follows thet

$$p_t(x') = p_0(\phi_t^{-1}(x'))\ \det\left[\frac{\partial \phi_t^{-1}}{\partial x}(x')\right]. \tag{2}$$

This is the standard formula for tranformation of probability distribution. You can find this in section 3.1 on my notes on the subject [4].

- The formula in Equation (2) is not that great because there is an issue with variable capture. The x in ∂x is not a variable but a shorthard the positiional argument of a function. I previously have introduced a system to deal with this kind of problem [3]. So, let's write the equation using that notation.

  First, we note that $\phi_t(x) = q(t, x)$ is a function that maps a $(d+1)$-dimensional space to a d-dimensional space. So, we can treat it in the same way as a function of signature $\mathbb{R}^{d+1} \to \mathbb{R}^d$. In other words, we can say that ϕ takes $d + 1$ inputs. We can then divide the $d + 1$ inputs into two blocks.

  - The first block is the first argument alone. Using Python slice notation, it is "$1 : 2$." Using my "chapter" notation, it can be abbreviated as §1.
  - The second block is the rest of the arguments. Using Python slice notation, it is "$2 : d+2$." Using my "chapter" notation, it can be abbreviate as §2.

  Hence, using my notation for partial derivatives, we can rewrite the equation as:

  $$p_t(x') = p_0(\phi_t^{-1}(x'))\ \det \nabla_{§2}\phi_t^{-1}(x')$$

  or, to be even briefer

  $$p_t(x') = p_0(\phi_t^{-1}(x'))\ |\nabla_{§2}\phi_t^{-1}(x')|. \tag{3}$$

- Note that we can rewrite Equation (1) using my notation as:

  $$\nabla_1 \phi_t(x) = v_t(\phi_t(x)). \tag{4}$$

- Let $f : \mathbb{R}^d \to \mathbb{R}$ and let $v : \mathbb{R}^d \to \mathbb{R}^d$. A **push-forward** (or a change of variable) of f according to v is a function of $g : \mathbb{R}^d \to \mathbb{R}$ defined by

  $$g(y) = f(v^{-1}(y))\ |\nabla v^{-1}(y)|.$$

  Here, ∇ denotes the derivative operator, which gets you the Jacobian matrix. We denote the push-forward of f according to v as $[v]_* f$.

- In the context of the discussion so far, we have that $p_t = [\phi_t]_* p_0$.

- When a time-dependent vector field v_t generetes a flow ϕ_t and when $p_t = [\phi_t]_* p_0$,
  we says that v_t **generates** p_t.

- When we use a neural ODE to transform a probability distribution from one to another (i.e., transforming p_0 from p_1), we call the resulting model a **continuous normalizing flow** model.

2 Flow Matching

2.1 Flow Matching Objective

- We want to use the above framework to transform a simple noise distribution $p_0 = p_{\text{noise}}$ to a data distribution $p_1 = p_{\text{data}}$.

- p_{noise} is typically a Gaussian distribution $p_{\text{noise}} = \mathcal{N}(0, I)$.
- As in most ML settings, we do not have access to the density function p_{data}, but we only have samples from the distribution.

- Suppose we know a probability path p_t and a time-dependent vector field u_t that has the following property:
  - p_0 is the desired noise distribution, and p_1 is the desired data distribution.
  - u_t generates p_t.

  Suppose again that we want to model u_t with a neural network $v_t(x; \theta)$. Then, we may do it my minimizing the **flow matching objective**:

$$\mathcal{L}_{\text{FM}}(\theta) = E_{\substack{t \sim \mathcal{U}([0,1]), \\ x \sim p_t(x)}} \left[\|u_t(x) - v_t(x; \theta)\|^2 \right]. \tag{5}$$

- The flow maching objective is usable if we know p_t and u_t before hand. However, in our settings, we do not know anything about u_t, and we only know $p_0 = p_{\text{noise}}$ and $p_1 = p_{\text{data}}$ but nothing in between.

2.2 Special Case: Single Item Dataset

- One of the difficulty we are facing right now is that p_{data} can be quite complicated and that we only have access to its samples, not a function that can evaluate the density of sample from the distribution.
- So, let's start with a special case where the distribution can generate exactly one data item. Let us call this item x_{data}.
  - You know where this is going. We will later approximate p_{data} as a mixture of the distributions of individual samples. So, stay tuned and work with this special case first.
- The distribution p_{data} is given by $p_{\text{data}} = \delta(x_{\text{data}})$ where δ is the Dirac delta function.
- We want to derive a vector field u_t that generates a probability distribution p_t so that (1) $p_0 = p_{\text{noise}} = \mathcal{N}(0, I)$ and $p_1 = p_{\text{data}} = \delta(x_{\text{data}})$.
- Unfortunately, I don't think there is a finite-time process that can turn a Gaussian distribution into a delta distribution. So, we will settle for an approximation. We instead require that

$$p_1 = \mathcal{N}(x_{\text{data}}, \sigma_{\text{min}}^2 I)$$

  where σ_{min} is a small positive constant.
- Since u_t and p_t we shall derive is specific to x_{data}, we may write them as "conditional" vector field and probability density, using the notation $u_t(\cdot|x_{\text{data}})$ and $p_t(\cdot|x_{\text{data}})$.
  - Of course, this will be used later when we approximate of p_{data} as a mixture of single item distributions.
- In other words, we want to find a time-dependent vector field $u_t(\cdot|x_{\text{data}})$ that generates a probability path $p_t(\cdot|x_{\text{data}})$ such that
  - $p_0(x|x_{\text{data}}) = \mathcal{N}(x; 0, I)$, and
  - $p_1(x|x_{\text{data}}) = \mathcal{N}(x; x_{\text{data}}, \sigma_{\text{min}}^2 I)$.

- We let $p_t(\cdot|x_{\text{data}})$ take the form

$$p_t(x|x_{\text{data}}) = \mathcal{N}(x; \mu_t(x_{\text{data}}), \sigma_t(x_{\text{data}})^2 I) \tag{6}$$

  where $\mu : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : [0,1] \times \mathbb{R} \to \mathbb{R}^+$. We will specify these two functions later, but there are many choises of them.

- To satsify the requirement on p_0, it must be the case that
  - $\mu_0(x_{\text{data}}) = 0$ for all x_{data}, and
  - $\sigma_0(x_{\text{data}}) = 1$ for all x_{data}.

  Morever, to satisfy the requirment on p_1, it must be the case that
  - $\mu_1(x_{\text{data}}) = x_{\text{data}}$ for all x_{data}, and
  - $\sigma_1(x_{\text{data}}) = \sigma_{\min}$ for all x_{data}.

- Now that we have specified the form of $p_t(\cdot|x_{\text{data}})$, it is now time to figure out the vector field $u_t(\cdot|x_{\text{data}})$ that generates it.

- We do so by first specifying a flow ψ_t such that $p_t(\cdot|x_{\text{data}}) = [\psi_t]_* p_0(\cdot|x_{\text{data}})$. Then, we can define u_t according to the equation

$$\nabla_1 \psi_t(x) = u_t(\psi_t(x)|x_{\text{data}}).$$

  In other words,

$$u_t(x'|x_{\text{data}}) = \nabla_1 \psi_t(\psi_t^{-1}(x')). \tag{7}$$

- Now, let's specify ψ_t. We use a very simple flow:

$$\psi_t(x) = \sigma_t(x_{\text{data}})x + \mu_t(x_{\text{data}})$$

  Let's do some sanity check.
  - At $t = 0$, we have that $\phi_0(x) = x$. So, ϕ_t satisfies the initial condition. The distribution p_0 is the distribution of x, which is $\mathcal{N}(0, I)$ as required.
  - At $t = 1$, we have that $\phi_1(x) = \sigma_{\min} x + x_{\text{data}}$. Because $x \sim \mathcal{N}(0, I)$, we have that $p_1 \sim \mathcal{N}(x_{\text{data}}, \sigma_{\min}^2 I)$ as required too.
  - At other values of t, we have that $\phi_t(x) = \sigma_t(x_{\text{data}})x + \mu_t(x_{\text{data}})$. Again, because $x \sim \mathcal{N}(0, I)$, we have that $\phi_t(x) \sim \mathcal{N}(\mu_t x_{\text{data}}, \sigma_t(x_{\text{data}})^2 I)$ as required again.

- Let's derive $u_t(x'|x_{\text{data}})$.
  - First, we need to derive $\psi_t^{-1}(x')$. Let $x = \psi_t^{-1}(x')$. We have that

$$x' = \psi_t(x) = \sigma_t(x_{\text{data}})x + \mu_t(x_{\text{data}}).$$

    So,

$$x = \frac{x' - \mu_t(x_{\text{data}})}{\sigma_t(x_{\text{data}})}.$$

    In other words,

$$\psi_t^{-1}(x') = \frac{x' - \mu_t(x_{\text{data}})}{\sigma_t(x_{\text{data}})}. \tag{8}$$

– Second, we need to derive the time-derivative $\nabla_1 \psi_t(x)$. This is also simple:

$$\begin{aligned}\nabla_1 \psi_t(x) &= \frac{\partial}{\partial t}\big[\sigma_t(x_{\text{data}})x + \mu_t(x_{\text{data}})\big] \\ &= \left(\frac{\partial}{\partial t}\sigma_t(x_{\text{data}})\right)x + \frac{\partial}{\partial t}\mu_t(x_{\text{data}}) \\ &= x\nabla_1\sigma_t(x_{\text{data}}) + \nabla_1\mu_t(x_{\text{data}}). \end{aligned} \tag{9}$$

- Substituting (8) and (9) into (7), we have that

$$\begin{aligned} u_t(x'|x_{\text{data}}) &= \left(\frac{x' - \mu_t(x_{\text{data}})}{\sigma_t(x_{\text{data}})}\right)\nabla_1\sigma_t(x_{\text{data}}) + \nabla_1\mu_t(x_{\text{data}}) \\ &= \frac{\nabla\sigma_t(x_{\text{data}})}{\sigma_t(x_{\text{data}})}(x' - \mu_t(x_{\text{data}})) + \nabla_1\mu_t(x_{\text{data}}). \end{aligned}$$

- **Theorem 1.** *Suppose that we are given the following functions.*
  - *Let $\sigma : [0,1] \times \mathbb{R}^d \to \mathbb{R}^+$ be a differentable function such that $\sigma_0(x) = 1$ and $\sigma_1(x) = \sigma_{\min}$ for all x.*
  - *Let $\mu : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ be a differentiable time-dependent vector field such that $\mu_0(x) = 0$ and $\mu_1(x) = x$ for all x.*

  Then, the vector field

$$u_t(x|x_{\text{data}}) = \frac{\nabla\sigma_t(x_{\text{data}})}{\sigma_t(x_{\text{data}})}(x - \mu_t(x_{\text{data}})) + \nabla_1\mu_t(x_{\text{data}})$$

  generates the flow

$$\phi_t(x) = \sigma_t(x_{\text{data}})x + \mu_t(x_{\text{data}})$$

  and a probablity path $p_t(\cdot|x_{\text{data}})$ such that that

$$p_t(\cdot|x_{\text{data}}) \sim \mathcal{N}(\mu_t(x_{\text{data}}), \sigma_t(x_{\text{data}})^2 I)$$

  for all t. In particular, we have that

  1. *$p_0(\cdot|x_{\text{data}}) \sim \mathcal{N}(0, I)$, and*
  2. *$p_1(\cdot|x_{\text{data}}) \sim \mathcal{N}(x_{\text{data}}, \sigma_{\min}^2 I)$.*

  So, $u_t(\cdot|x_{\text{data}})$ transforms a Gaussian noise distribution into an approximation of a single data distribution that only contains x_{data}.

2.3 From single item distribution to multi-item distribution

- Now, we get back to the case where p_{data} is not a distrbution that output only a single item.
- Using the law of total proability, we can define the **marginal probability path** as

$$p_t(x) = \int p_t(x|x_1)p_{\text{data}}(x_{\text{data}})\,\mathrm{d}x_{\text{data}}. \tag{10}$$

- It follows that $p_0 = \mathcal{N}(0, I)$, and

$$p_1 = p_{\text{data}} * \mathcal{N}(0, \sigma_{\min}^2 I) \approx p_{\text{data}}.$$

  where $*$ is the convolution operation. So, we can use p_1 in place of p_{data} in many cases.

- Our task is now to find a time-dependent vector field u_t that generates p_t. The paper argues that the following **marginal vector field**,

$$u_t(x) = \int u_t(x|x_{\text{data}}) \frac{p_t(x|x_{\text{data}}) p_{\text{data}}(x_{\text{data}})}{p_t(x)} \, \mathrm{d}x_{\text{data}}, \tag{11}$$

works.

- **Theorem 2.** *The marginal vector field u_t defined in Equation* (11) *generates the marginal probability path p_t in Equation* (10)*.*

Proof. This proof makes heavy use of the continuity equation (12) and Theorem 4.

We showed in the last section that the conditional vector field $u_t(\cdot|x_{\text{data}})$ generates the condtional probability path $p_t(\cdot|x_{\text{data}})$. To make derivation easier, we shall write $p_t(x|x_{\text{data}})$ as $p_{|x_{\text{data}}}(t,x)$ and $u_t(x|x_{\text{data}})$ as $u_{|\text{data}}(t,x)$. With this, we have that these two functions satisfy the continuity equation

$$\nabla_1 p_{|x_{\text{data}}}(t,x) + \sum_{i=1}^{d} \nabla_{i+1}(p_{|x_{\text{data}}} u^i_{|x_{\text{data}}})(t,x) = 0.$$

In other words,

$$\nabla_1 p_{|x_{\text{data}}}(t,x) = -\sum_{i=1}^{d} \nabla_{i+1}(p_{|x_{\text{data}}} u^i_{|x_{\text{data}}})(t,x).$$

Now, recall the definition of $p(t,x)$.

$$p(t,x) = \int p_{|x_{\text{data}}}(t,x) p_{\text{data}}(x_{\text{data}}) \, \mathrm{d}x_{\text{data}}.$$

Differentiating both sides with respect to the first argument (t), we have that

$$\begin{aligned}
\nabla_1 p(t,x) &= \int \nabla_1 p_{|x_{\text{data}}}(t,x) p_{\text{data}}(x_{\text{data}}) \, \mathrm{d}x_{\text{data}} \\
&= \int \Big(-\sum_{i=1}^{d} \nabla_{i+1}(p_{|x_{\text{data}}} u^i_{|x_{\text{data}}})(t,x) \Big) p_{\text{data}}(x_{\text{data}}) \, \mathrm{d}x_{\text{data}} \\
&= -\sum_{i=1}^{d} \int \nabla_{i+1}(p_{|x_{\text{data}}} u^i_{|x_{\text{data}}})(t,x) p_{\text{data}}(x_{\text{data}}) \, \mathrm{d}x_{\text{data}} \\
&= -\sum_{i=1}^{d} \nabla_{i+1} \Big(\int p_{|x_{\text{data}}}(t,x) u_{|x_{\text{data}}}(t,x) p_{\text{data}}(x_{\text{data}}) \, \mathrm{d}x_{\text{data}} \Big) \\
&= -\sum_{i=1}^{d} \nabla_{i+1} \Big(\int p_t(x|x_{\text{data}}) u_t(x|x_{\text{data}}) p_{\text{data}}(x_{\text{data}}) \, \mathrm{d}x_{\text{data}} \Big).
\end{aligned}$$

By Equation (11), we have that

$$u_t(x) p_t(x) = \int p_t(x|x_{\text{data}}) u_t(x|x_{\text{data}}) p_{\text{data}}(x_{\text{data}}) \, \mathrm{d}x_{\text{data}}.$$

As a result,

$$\nabla_1 p(t,x) = -\sum_{i=1}^{d} \nabla_{i+1}\big(u_t(x)p_t(x)\big)$$

$$\nabla_1 p(t,x) = -\sum_{i=1}^{d} \nabla_{i+1}(up)(t,x)$$

$$\nabla_1 p(t,x) + \sum_{i=1}^{d} \nabla_{i+1}(up)(t,x) = 0.$$

This shows that u_t and p_t satisfies the continuity equation, which implites that u_t generates p_t. □

2.4 Conditional Flow Matching

- We have just identified the vector field u_t that we can used in the flow matching loss (5). However, the problem is that u_t is defined as an integral, and we do not want to compute it directly.
- Instead, we optimize the following **conditional flow matching objective** where we sample x_{data} and trying to match $v_t(x;\theta)$ against $u_t(x|x_{\text{data}})$. Here, x is sampled from the conditional distribution $p_t(x|x_{\text{data}})$.

$$\mathcal{L}_{\text{CFM}}(\theta) = E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\|u_t(x|x_{\text{data}}) - v_t(x;\theta)\|^2].$$

- We can sample x_{data} easily because we can sample uniformly from the collections of samples we have at hand.
- We can also sample from $p_t(x|x_{\text{data}})$ easily because $p_t(x|x_{\text{data}}) = \mathcal{N}(x; \mu_t\text{data}(x_{\text{data}}), \sigma_t(x_{\text{data}})^2 I)$.
- The only concern is whether the conditional flow matching objective $\mathcal{L}_{\text{CFM}}(\theta)$ would yield the same θ as $\mathcal{L}_{\text{FM}}(\theta)$ after optimization. The answer is yes.
- **Theorem 3.** *Assuming that $p_t(x) > 0$ for all $x \in \mathbb{R}^d$ and $t \in [0,1]$, then,*

$$\mathcal{L}_{\text{FM}}(\theta) = \mathcal{L}_{\text{CFM}}(\theta) + C$$

where C is a constant independent of θ. As a result,

$$\nabla_\theta \mathcal{L}_{\text{FM}}(\theta) = \nabla_\theta \mathcal{L}_{\text{CFM}}(\theta).$$

Proof. We assume that all functions are well-behaved so that we can say that all integrals exist use the standard trick such as exchanging the order of integration (Fubini's theorem).

We have that

$$\|v_t(x;\theta) - u_t(x)\|^2 = \|v_t(x;\theta)\|^2 - 2\langle v_t(x;\theta), u_t(x)\rangle + \|u_t(x)\|^2$$
$$\|v_t(x;\theta) - u_t(x|x_{\text{data}})\|^2 = \|v_t(x;\theta)\|^2 - 2\langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle + \|u_t(x|x_{\text{data}})\|^2.$$

So,

$$\mathcal{L}_{\text{FM}}(\theta) = E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}[\|v_t(x;\theta)\|^2] - 2E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}[\langle v_t(x;\theta), u_t(x)\rangle] + E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}[\|u_t(x)\|^2]$$
$$\mathcal{L}_{\text{CFM}}(\theta) = E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\|v_t(x;\theta)\|^2] - 2E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle] + E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\|u_t(x|x_{\text{data}})\|^2].$$

Note that $u_t(x)$ and $u_t(x|x_{\text{data}})$ does not depend on θ. As a result, we can treat them as constants. In other words,

$$\mathcal{L}_{\text{FM}}(\theta) = E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}[\|v_t(x;\theta)\|^2] - 2E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}[\langle v_t(x;\theta), u_t(x)\rangle] + C_1$$
$$\mathcal{L}_{\text{CFM}}(\theta) = E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\|v_t(x;\theta)\|^2] - 2E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle] + C_2.$$

Next, note that

$$E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}[\|v_t(x;\theta)\|^2] = E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\|v_t(x;\theta)\|^2]$$

because we still sample x from the same distribution $p_t(x)$. On the LHS, we sample x directly, but, on the RHS, we sample x_{data} before sampling x given x_{data}.

Lastly,

$$\begin{aligned}
&E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}[\langle v_t(x;\theta), u_t(x)\rangle] \\
&= E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}\left[\left\langle v_t(x;\theta), \int u_t(x|x_{\text{data}})\frac{p_t(x|x_{\text{data}})p_{\text{data}}(x_{\text{data}})}{p_t(x)}\,\mathrm{d}x_{\text{data}}\right\rangle\right] \\
&= E_{\substack{t\sim\mathcal{U}([0,1]),\\ x\sim p_t(x)}}\left[\int \langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle\frac{p_t(x|x_{\text{data}})p_{\text{data}}(x_{\text{data}})}{p_t(x)}\,\mathrm{d}x_{\text{data}}\right] \\
&= E_{t\sim\mathcal{U}([0,1])}\left[\int p_t(x)\left(\int \langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle\frac{p_t(x|x_{\text{data}})p_{\text{data}}(x_{\text{data}})}{p_t(x)}\,\mathrm{d}x_{\text{data}}\right)\mathrm{d}x\right] \\
&= E_{t\sim\mathcal{U}([0,1])}\left[\int\int \langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle p_t(x|x_{\text{data}})p_{\text{data}}(x_{\text{data}})\,\mathrm{d}x_{\text{data}}\,\mathrm{d}x\right] \\
&= E_{t\sim\mathcal{U}([0,1])}\left[\int p_{\text{data}}(x_{\text{data}})\left(\int p_t(x|x_{\text{data}})\langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle\,\mathrm{d}x\right)\mathrm{d}x_{\text{data}}\right] \\
&= E_{\substack{t\sim\mathcal{U}([0,1]),\\ x_{\text{data}}\sim p_{\text{data}},\\ x\sim p_t(x|x_{\text{data}})}}[\langle v_t(x;\theta), u_t(x|x_{\text{data}})\rangle].
\end{aligned}$$

We are done. □

3 Conditional Vector Fields

- We said earlier that there are multiple ways to define μ_t and σ_t in Equation (6). We discuss them in this section.

3.1 Diffusion Conditonal Vector Fields

- For a diffusion model, time is the reverse of what we have been using in this note.
  - p_0 is p_{data} or an approximation of it.
  - p_1 is a noise distribution.

  However, to make things simple, we will reverse the time so that it complies with what we have in this note.

- Variance-exploding case.

– Given x_{data}, we have that $p_t(x|x_1) = \mathcal{N}(x; x_{\text{data}}, \sigma_)$

- Variance-preservin case

  –

A Continuity Equation

- The continuity equation is an equation from fluid dynamics that shows up a lot in fields that involves transport pheonomena.
- Here, we start with a probability density function p_0, which tells us how the probability mass is distributed over $\mathbb{R}^d$. Then, we have a vector field v_t that gives us the time-dependent velocity field that governs how the mass at each point should move. The velocity field generates a probability path p_t.
- Note that, because the velocity field just move the mass around, there is no new mass added or no new mass being dropped. It follows that the total probability mass remains constant. It is *conserved*. This means that p_t is a probability distribution for all t. If you integrate it over $\mathbb{R}^d$, you should get 1.
- Given a probability path p_t and a time-dependent velocity field v_t that generates it, the **flux density** $j : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is defined as:

  $$j_t(x) = p_t(x)v_t(x).$$

  It is just the velocity field weighted by the probability density.
- Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be a vector field. The **divergence** of f is a scalar function $\nabla \cdot f : \mathbb{R}^d \to \mathbb{R}$ define by

  $$(\nabla \cdot f)(x) = \sum_{i=1}^{d} \nabla_i f^i(x)$$

  where $f^i(x)$ denotes the ith component of $f(x)$.
- The **continuity equation** of p_t and v_t is given by

  $$\frac{\partial}{\partial t} p_t(x) + (\nabla \cdot j_t)(x) = 0.$$

  In the above equation, we view j_t with t fixed as a vector field of signature $\mathbb{R}^d \to \mathbb{R}^d$.
- Let's rewrite the continuity equation with my notation and taking into account correct indexing with no ambiguity whatsoever. We have

  $$\nabla_1 p(t,x) + \sum_{i=1}^{d} \nabla_{i+1}(pv^i)(t,x) = 0. \tag{12}$$

- In our context, the continuity equation is useful because of the following theorem.

  Theorem 4. *Let $p : [0,1] \times \mathbb{R}^d \to \mathbb{R}^+ \cup \{0\}$ be a differentiable probability path. Let $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ be a differentiable time-dependent vector field. Then, v_t generates p_t if and only if the continuity equation holds.*

- We shall not prove the theorem in this note, but I will be studying it more.

References

[1] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations, 2019.

[2] Khungurn, P. Neural ordinary differential equations. https://pkhungurn.github.io/notes/notes/ml/neural-ode/neural-ode.pdf. Accessed: 2024-07-19.

[3] Khungurn, P. Notation for multivariable derivatives. https://pkhungurn.github.io/notes/notes/math/multivar-deriv-notations/multivar-deriv-notations.pdf. Accessed: 2024-07-22.

[4] Khungurn, P. Probability under transformation. https://pkhungurn.github.io/notes/notes/gfx/pdf-transform/pdf-transform.pdf. Accessed: 2024-07-22.

[5] Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling, 2023.