Diffusion Models and Score Matching methods

Tri Nguyen

Internal reading group
Oregon State University

February 24, 2023

What would be covered

1. The emergent of diffusion model: Jascha Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265
2. The rise of score matching approach: Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems* 32 [2019]

Problem settings

- Given i.i.d *images* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ drawn from unknown $p(\boldsymbol{x})$.
- We want to draw new *images* $\boldsymbol{x} \sim p(\boldsymbol{x})$!

What have been done: VI, VAE, ...

Brief summary on the use of MLE principle $\max \log p(\boldsymbol{x})$. Assuming there is a latent factor $\boldsymbol{z}$,

- Variation inference (VI):

$$\max_{q \in \mathcal{Q}} \log p(\boldsymbol{x}) = \max_{q \in \mathcal{Q}} \{\mathcal{L}(q) + \mathsf{KL}(q(\boldsymbol{z}) || p(\boldsymbol{z}|\boldsymbol{x}))\},$$

$$\mathcal{L}(q) \triangleq \mathop{\mathbb{E}}_{q(\boldsymbol{z})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right]$$

  VI assumes $\mathcal{Q} = \{q(\cdot) : q(\boldsymbol{z}) = \prod_{i=1}^{m} q(\boldsymbol{z}_i)\}$ and analytically derive coupled equations between $\boldsymbol{z}_i$, and often be solved be iterative method.

- VAE: Assume the true joint distribution $p_{\boldsymbol{\theta}^\star}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}^\star}(\boldsymbol{z}) p_{\boldsymbol{\theta}^\star}(\boldsymbol{x}|\boldsymbol{z})$.

$$\max_{\boldsymbol{\theta}, \boldsymbol{\phi}} \log p(\boldsymbol{x}) = \max_{\boldsymbol{\theta}, \boldsymbol{\phi}} \{\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \mathsf{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}))\},$$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} [-\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})],$$

Intention between tractability and model complexity!

Diffusion model: The general goal

- "Deep unsupervised learning using nonequilibrium thermodynamics" aims to simultaneously achieves both flexibility and tractability.
- **[Very informal]** Find a transformation $\mathcal{T}$ such that

$$\boldsymbol{x} \sim p_{\mathsf{data}}(\boldsymbol{x}) \Rightarrow \mathcal{T}(\boldsymbol{x}) \sim p_{\mathsf{nice}}(\boldsymbol{x})$$

  and

$$\boldsymbol{x} \sim p_{\mathsf{nice}}(\boldsymbol{x}) \Rightarrow \mathcal{T}^{-1}(\boldsymbol{x}) \sim p_{\mathsf{data}}(\boldsymbol{x})$$

Deep unsupervised learning using nonequilibrium thermodynamics

- Define a Markov chain (forward):
  $\boldsymbol{x}^0 \to \boldsymbol{x}^1 \to \boldsymbol{x}^2 \to \ldots \to \boldsymbol{x}^{T-1} \to \boldsymbol{x}^T$

$$q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1}) \triangleq \mathcal{N}(\boldsymbol{x}^t; \boldsymbol{x}^{t-1}\sqrt{1-\beta_t}, \beta_t \boldsymbol{I}), \quad 0 \leq \beta_t \leq 1$$

$$q(\boldsymbol{x}^{0\ldots T}) = q(\boldsymbol{x}^0) \prod_{i=1}^{T} q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1})$$

- Then,

$$q(\boldsymbol{x}^t|\boldsymbol{x}^0) = \mathcal{N}(\boldsymbol{x}^t|\sqrt{\overline{\alpha}_t}\boldsymbol{x}^0, (1-\overline{\alpha}_t)\boldsymbol{I}), \qquad \overline{\alpha}_t \triangleq \prod_{i=1}^{t}(1-\beta_i)$$

  which implies $q(\boldsymbol{x}^T|\boldsymbol{x}^0) \approx \mathcal{N}(\boldsymbol{x}^T; \boldsymbol{0}, \boldsymbol{I})$ if $\overline{\alpha}_T \to 0$.
- And also, $q(\boldsymbol{x}^T) \approx \mathcal{N}(\boldsymbol{x}^T; \boldsymbol{0}, \boldsymbol{I})$ when T is large enough (?)

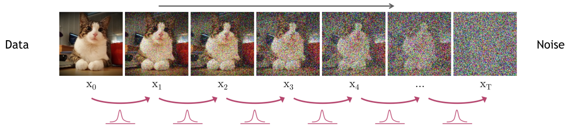


Figure: CVPR 2022 tutorial

Generative Process

Let $q(\boldsymbol{x}^0)$ be data distribution. Given that Markov chain, how to sample from $p(\boldsymbol{x}^0|\boldsymbol{x}^T)$? Note that the forward is fixed, conditional $q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1})$ is known, $q(\boldsymbol{x}^T) \approx \mathcal{N}(\boldsymbol{x}^T|\mathbf{0}, \boldsymbol{I})$.

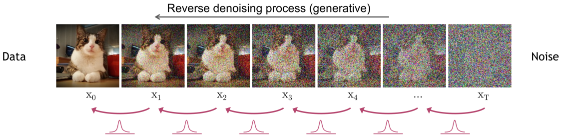


Figure: CVPR 2022 tutorial

A naive but sound strategy:

- Sample $\boldsymbol{x}^T \sim \mathcal{N}(\boldsymbol{x}^T|\mathbf{0}, \boldsymbol{I})$
- Sample $\boldsymbol{x}^{t-1} \sim p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t) \propto p(\boldsymbol{x}^{t-1}, \boldsymbol{x}^t) = q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1})p(\boldsymbol{x}^{t-1}) \Rightarrow$ intractable.

Good news is if β_t in $q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1}) \triangleq \mathcal{N}(\boldsymbol{x}^t; \boldsymbol{x}^{t-1}\sqrt{1-\beta_t}, \beta_t \boldsymbol{I})$ is small enough, then $p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^1)$ is also a normal distribution.

Recipe

- Let $q(\boldsymbol{x}^0)$ denote the unknown data distribution
- Define $\beta_t, 1 \leq t \leq T$ such that

$$q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1}) \triangleq \mathcal{N}(\boldsymbol{x}^t; \boldsymbol{x}^{t-1}\sqrt{1-\beta_t}, \beta_t \boldsymbol{I}), \quad 0 \leq \beta_t \leq 1 \tag{1}$$

$$q(\boldsymbol{x}^T|\boldsymbol{x}^0) \approx \mathcal{N}(\boldsymbol{x}^T; \boldsymbol{0}, \boldsymbol{I}) \tag{2}$$

$$p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t) \text{ is normal} \quad \forall 1 \leq t \leq T \tag{3}$$

Since we know $p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t)$ is normal, it can be parameterized as

$$p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t) \sim \mathcal{N}(\boldsymbol{x}^{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}^t, t), \sigma^2 \boldsymbol{I})$$

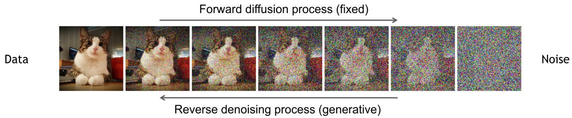


Figure: CVPR 2022 tutorial

There is no assumption on data distribution

Training

- Latent variables $\boldsymbol{x}^{1\dots T}$
- Model probability $p(\boldsymbol{x}^0) = \int p(\boldsymbol{x}^{0\dots T}) d\boldsymbol{x}^{1\dots T}$
- Data distribution $q(\boldsymbol{x}^0)$
- Posterior probability $q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0)$

We try to minimize KL divergence between model probability and the real data distribution (which reduces to MLE),

$$\underset{\boldsymbol{x}^{1\dots T}}{\text{maximize}} \underset{\boldsymbol{x}\sim q(\boldsymbol{x}^0)}{\mathbb{E}} \log p(\boldsymbol{x}^0)$$

$$\begin{aligned} p(\boldsymbol{x}^0) &= \int p(\boldsymbol{x}^{0\dots T}) \frac{q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0)}{q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0)} d\boldsymbol{x}^{1\dots T} \\ &= \int q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0) \frac{p(\boldsymbol{x}^{0\dots T})}{q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0)} d\boldsymbol{x}^{1\dots T} \\ &= \int q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0) p(\boldsymbol{x}^T) \prod_{i=1}^{T} \frac{p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t)}{q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1})} d\boldsymbol{x}^{1\dots T} \\ &= \underset{q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0)}{\mathbb{E}} \left[p(\boldsymbol{x}^T) \prod_{i=1}^{T} \frac{p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t)}{q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1})} \right] \end{aligned}$$

Training

What we want

$$\underset{\boldsymbol{x}^{1\dots T}}{\text{maximize}} \underset{\boldsymbol{x}\sim q(\boldsymbol{x}^0)}{\mathbb{E}} \log p(\boldsymbol{x}^0)$$

What we know is

$$\begin{aligned}\underset{\boldsymbol{x}\sim q(\boldsymbol{x}^0)}{\mathbb{E}} \log p(\boldsymbol{x}^0) &= \underset{q(\boldsymbol{x}^0)}{\mathbb{E}} \log \left(\underset{q(\boldsymbol{x}^{1\dots T}|\boldsymbol{x}^0)}{\mathbb{E}} \left[p(\boldsymbol{x}^T) \prod_{i=1}^{T} \frac{p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t)}{q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1})} \right] \right) \\ &\geq \underset{q(\boldsymbol{x}^{0\dots T})}{\mathbb{E}} \log \left[p(\boldsymbol{x}^T) \prod_{i=1}^{T} \frac{p(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t)}{q(\boldsymbol{x}^t|\boldsymbol{x}^{t-1})} \right]\end{aligned}$$

Estimate un-normalized probability model

Problem setting:

- $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^n$ are drawn i.i.d from $p_{\text{data}}(\boldsymbol{x})$.
- Assume we know that p_{data} belong a distribution class $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = q(\boldsymbol{x}; \boldsymbol{\theta})/Z(\boldsymbol{\theta})$.
- Functional form of $q(\boldsymbol{x}; \boldsymbol{\theta})$ is known, but $Z(\boldsymbol{\theta}) = \int_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{\theta}) d\boldsymbol{x}$ is intractable.
- Goal: We want to use $\boldsymbol{x}_i$'s to estimate $\boldsymbol{\theta}_{\text{data}}$ corresponding to p_{data} (assume it is unique).

[Hyvärinen and Dayan 2005] proposed to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \ \underset{p_{\text{data}}}{\mathbb{E}} \left[\left\| \nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x}) \right\|^2 \right] \tag{4}$$

- Normalization factor plays no role here. $\nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} (\log q(\boldsymbol{x}; \boldsymbol{\theta}) - \log Z(\boldsymbol{\theta})) = \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}; \boldsymbol{\theta})$.
- (1) is surprisingly equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \ \underset{p_{\text{data}}}{\mathbb{E}} \left[\text{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})) + \frac{1}{2} \left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right\|^2 \right]$$

where the so-cal score $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) \triangleq \nabla_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{\theta})$.

Generative Modeling by Estimating Gradients of the Data Distribution

General recipe include 2 ingredients:

- Step 1: Using score match to estimate score of data distribution.
- Step 2: Using Langevin dynamics to draw samples using score function.

  $$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \frac{\epsilon}{2}\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}_{t-1}) + \sqrt{\epsilon}\boldsymbol{z}_t,$$

  where $\boldsymbol{z}_t \sim \mathcal{N}(0, \boldsymbol{I}), \boldsymbol{x}_0 \sim \pi(\boldsymbol{x})$. This would produce $\boldsymbol{x}_t \sim p(\boldsymbol{x})$ when $\epsilon \to 0, t \to \infty$ (in practice, $T = 100, \epsilon = 2e^{-5}$).

Generative Modeling by Estimating Gradients of the Data Distribution

Challenges in step 1: computation complexity

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \; \underset{p_{\text{data}}}{\mathbb{E}} \left[\text{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})) + \frac{1}{2} \left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right\|^2 \right]$$

- Computing the first term $\text{tr}(\cdot)$ (involving Jacobian) is costly for high dimensional data.
  - Solution 1 [Vincent 2011]. Add pre-specified noise to data $q_\sigma(\widetilde{\boldsymbol{x}}|\boldsymbol{x})$, then using score matching to learn score of $q_\sigma(\boldsymbol{x}) = \int_{\boldsymbol{x}} q_\sigma(\widetilde{\boldsymbol{x}}|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}) d\boldsymbol{x}$ (instead of p_{data}).
    It was shown that the objective is equivalent to

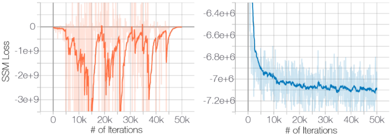
    $$\underset{\widetilde{\boldsymbol{x}} \sim q_\sigma(\cdot)}{\mathbb{E}} \left[\left\| \boldsymbol{s}_\theta(\widetilde{\boldsymbol{x}}) - \nabla_{\widetilde{\boldsymbol{x}}} \log q_\sigma(\widetilde{\boldsymbol{x}}|\boldsymbol{x}) \right\|^2 \right],$$

    and by score matching's result, the optimal solution $\boldsymbol{s}_{\boldsymbol{\theta}^\star}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log q_\sigma(\boldsymbol{x}) \approx p_{\text{data}}(\boldsymbol{x})$.

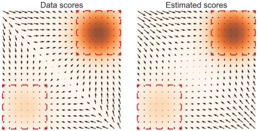
- Solution 2: [Song et al. 2019] Random projection to estimate $\mathrm{tr}(\cdot)$. The objective now become

$$\mathop{\mathbb{E}}_{p_{\boldsymbol{v}}} \mathop{\mathbb{E}}_{p_{\text{data}}} \left[\boldsymbol{v}^\top (\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})) \boldsymbol{v} + \frac{1}{2} \left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right\|^2 \right]$$

Several other challenges are demonstrated in [Song et al. 2020]. In the end, they proposed to add noise with different variance.



(a) Low dimension manifold. Left: train with original MNIST, right: add noise $\mathcal{N}(0, 0.0001)$



(b) In low density region, there is not enough data to learn $\nabla_{\boldsymbol{x}} \log p_{\text{data}}$

Suggestion if anyone's interested

- Jonathan Ho et al. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 [2020], pp. 6840–6851
- Yang Song et al. "Score-based generative modeling through stochastic differential equations". In: *arXiv preprint arXiv:2011.13456* [2020]