

Introduction

◦ 기본 모델의 한계점

- Generative model은 크게 두 가지로 VAE와 같은 likelihood model, GAN이 존재함.

likelihood-based model은 normalized probability model을 위한 통화된 기기해야 필요하거나 surrogate loss를 필요로 함
GAN은 이 문제점을 피하기 위해 adversarial training으로 인해 unstable하다는 고질적인 문제 존재 (자세한 내용은 나중에 자료에)

◦ New Approaches

- 그래서 이 논문에서 generative를 위한 새로운 접근법을 제안함

logarithmic data density의 score를 가지고 estimating과 sampling을 하는 method를 제안

이때 data density는 input data point에서 log-density function의 gradient를 활용함. → 자세한 내용은 노트의 Figure 2에

그리고 더 아래로 흐르면 log data density가 가장 많이 증가하는 방향에서의 vector field를 추정하는 것이 목표임

- 이를 위해 score matching 이런 neural network를 training하고 Langevin dynamic을 이용해 샘플링을 수행

이때 Langevin dynamic이 원래 훈련하기엔 noise를 더해가는 방식인데 노이즈에서 estimated vector field of scores를

제거한 random initial sample을 가지고 high density region으로 이동시켜는 방식으로 작동이 됨



◦ Two main challenges

- 그래서 위의 두 접근법은 상당히 어렵다.

① 만약 data distribution이 low-dimensional manifold에 존재한다면?

위의 가설에 성립한다면 score는 이미 ambient space에서 경계가 멀될 것이다

따라서 Score matching은 일정한 score estimator를 제공하지 못함

② low density region이나 training data의 scarcity 차이면?

예를 들어서 dataset manifold로 하여 멀리 떨어져 있을 것.

low-dimensional manifold란?

고차원 공간에서 저차원으로 휘여져 있는

sub space를 의미. 즉, 실제로 데이터를

표현할 때는 저차원으로 가능한데 우리는

고차원으로 생각하는 것과 비슷한 맥락

◦ Propose New Methods.

- 그래서 다음과 같은 새로운 method들을 제안함

① To perturb the data with random Gaussian noise of various magnitudes.

data에 Gaussian noise를 추가하기 되면 low-dimensional manifold에 빠져들 가능성이 높아짐

즉, 이 노이즈에서는 noise level을 조건으로 하는 single score network를 학습하고 그 noise magnitude에서

score는 estimate하는 것을 제일 잘

② An annealed version of Langevin dynamics

high noise level에 있는 score를 사용하거나 초기에는 original data distribution과 구별이 안 될 정도의 noise level을 사용함

Score-based generative modeling

- 가정: our dataset consists of i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^N$ from an unknown data distribution $P_{\text{data}}(\mathbf{x})$
- 목표: probability density $P(\mathbf{x})$ 의 score $\nabla_{\mathbf{x}} \log P(\mathbf{x})$
- score network $S_{\theta}: \mathbb{R}^D \rightarrow \mathbb{R}^D$ (parameterized by θ) ($P_{\text{data}}(\mathbf{x})$ 의 score를 예측하도록 학습)
- ingredients: score matching and Langevin dynamics

• Score matching for score estimation

- Score matching을 사용해 $\nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 를 추정하기 위해 score network $S_{\theta}(\mathbf{x})$ 를 훈련

따라서 objective function은 $\frac{1}{2} E_{P_{\text{data}}} [\|S_{\theta}(\mathbf{x}) - \nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})\|_2^2]$ 라고 할 수 있음

하지만 우리는 $P_{\text{data}}(\mathbf{x})$ 를 직접 계산할 수 없기 때문에 이의 term이 있는 수식으로 유도를 한다.

위 식은 노트 레퍼런스 [53]에 의해 다음과 같이 유도할 수 있다.

$$E_{P_{\text{data}}} [\text{tr}(\nabla_{\mathbf{x}} S_{\theta}(\mathbf{x})) + \frac{1}{2} \|S_{\theta}(\mathbf{x})\|_2^2]$$

하지만 $\text{tr}(\nabla_{\mathbf{x}} S_{\theta}(\mathbf{x}))$ 은 high dimension을 소유한 영상상이 많아지는 문제가 있음

그렇기 large scale matching의 두 가지 method를 다룬다

① Denoising score matching

$\text{tr}(\nabla_{\mathbf{x}} S_{\theta}(\mathbf{x}))$ 을 피하기 위해 score matching의 이런 변화를 준 method.

먼저 pre-specified noise distribution $\mathcal{G}_0(\tilde{\mathbf{x}}|\mathbf{x})$ 를 이용해 data point \mathbf{x} 를 perturb 한다.

그리고 위 perturbed data distribution $\mathcal{G}_0(\tilde{\mathbf{x}}) = \int \mathcal{G}_0(\tilde{\mathbf{x}}|\mathbf{x}) P_{\text{data}}(\mathbf{x}) d\mathbf{x}$ 의 score를 추정하기 위해

score matching을 활용함

마지막 다음과 같이 objective function을 바꿀 수 있음

$$\frac{1}{2} E_{\mathcal{G}_0(\tilde{\mathbf{x}}|\mathbf{x})} P_{\text{data}}(\mathbf{x}) [\|S_{\theta}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \mathcal{G}_0(\tilde{\mathbf{x}}|\mathbf{x})\|_2^2]$$

그리고 $S_{\theta}^*(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log \mathcal{G}_0(\mathbf{x}) = \nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x}) +$ 고정하기 위해 noise가 굉장히 커야 가능함

Langevin dynamic을 이용할 때 $\nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 의 noise를 고정하기 위해 noise가 커야 함
high noise level에 대처하기 때문에 noise가 커야 함

② Sliced score matching

Sliced score matching은 $\text{tr}(\nabla_{\mathbf{x}} S_{\theta}(\mathbf{x}))$ 을 계산하기 위해 random projection을 사용함

objective는 다음과 같음

$$E_{P_{\theta}} E_{P_{\text{data}}} [V^T \nabla_{\mathbf{x}} S_{\theta}(\mathbf{x}) V + \frac{1}{2} \|S_{\theta}(\mathbf{x})\|_2^2]$$

이때 P_{θ} 는 multivariate standard normal과 같은 random vector의 simple distribution

위 식에서 $V^T \nabla_{\mathbf{x}} S_{\theta}(\mathbf{x}) V$ 는 forward mode auto-differentiation으로 계산이 가능함

sliced score matching은 original unperturbed data distribution에 대한 score estimation이 가능하지만

forward mode auto-differentiation 때문에 나아 더 많은 계산량을 필요로 함

Sampling with Langevin dynamics

- Langevin dynamics는 score function인 $\nabla_{\mathbf{x}} \log P(\mathbf{x})$ 을 사용해 probability density $P(\mathbf{x})$ 로부터 sample을 생성함
- 이 뿐만 아니라 image generation을 위해 gradient descent와 비슷한 алгоритмы를 사용함
즉, 노드가 있는 initial sample에서 data에 대한 gradient를 더하는 과정을 반복해 likelihood가 높아진 data를 찾는 것이다.
gradient ascent과 같은 봐도 되나...?
- 수식으로 표현하면 다음과 같음

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2} \nabla_{\mathbf{x}} \log P(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon} Z_t$$

여기 $\epsilon > 0$ 은 step size, initial value $\tilde{\mathbf{x}}_0 \sim \pi(\mathbf{x})$ 이고 π 는 prior distribution.

그리고 $Z_t \sim N(0, I)$ 은 update를 할 때마다 gradient를 부여하는 term

위의 update equation은 score function $\nabla_{\mathbf{x}} \log P(\mathbf{x})$ 를 필요로 함.

따라서 $P_{\text{data}}(\mathbf{x})$ 의 sample을 얻기 위해 score network를 $S_{\theta}(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 가 되도록 학습을 함.

그리고 나서 $S_{\theta}(\mathbf{x})$ 를 사용해 Langevin dynamics로부터 sample을 얻는다.

Challenges of score-based generative modeling

score-based generative modeling을 하는 데 두 가지 challenge를 볼 수 있다 (Introduction에서 다룬 내용은 더详细介绍)

① manifold hypothesis

data가 low-dimensional manifold에 존재한다는 가정이 성립한다면 그 가치가 크지 않다

- 1) data \mathbf{x} 가 low-dimensional manifold에 국한될 때 data의 score를 정의할 수 없다.

즉, score는 $\nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 으로 gradient를 구할 수 없음 (원래 고차원에 존재하는 data를 저차원으로 축소하면 정확한 gradient를 구하기 어렵음)

2) data \mathbf{x} 가 low-dimensional manifold에서 존재하면 일정한 있는 score estimation을 구하기 어려움

오른쪽 그림을 보면 data에 아무것도 processing도 하지

않고 sliced score matching을 한 경우, loss가 굉장히 불안정한 것을 알 수 있다.

*data를 perturb하는
low dimension에서
high dimension으로
동기는 효과가 있는 듯*

반면 data에 $N(0, 0.0001)$ 을 띠는 noise를

perturb 해주게 되면 loss가 수렴하는 현상을 볼 수 있다. 즉, range [0,1]의 value를 부과하는 것이다.

② Low data density regions

low density region에서 data 샘플은 Langevin dynamics sampling을 수행하는데 어려움이 있다.

크게 두 가지의 어려움이 있다.

1) Inaccurate score estimation with score matching

$P_{\text{data}}(\mathbf{p}) \approx 0$ 인 임의의 region $R \subset \mathbb{R}^D$ 를 고려하자.

그리면 대수율 $\{\mathbf{x}_i\}_{i=1}^N \cap R = \emptyset$ 이 되어

$\mathbf{x} \in R$ 에 대해 $\nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 를 정학하기

추장하지 못할 것이다.

또한 그림에서 high density region인

red box는 비교적 정학하기 흐름을 반영

low density region인 그 외의 영역에 대해서는

data 부족으로 인해 gradient $\nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 를 정학하기 흐름을 못했다.

2) Slow mixing of Langevin dynamics

만약 low density region이 의해 data distribution의 mode가 그대로 분리된다면 Langevin dynamics은

두 mode의 relative weight를 정학하기 찾기 어려울 것이다 true distribution으로 수렴하지 못할 것이다.

$\pi \in (0,1)$ 에 의해 mixture distribution $P_{\text{data}}(\mathbf{x}) = \pi P_1(\mathbf{x}) + (1-\pi) P_2(\mathbf{x})$ 를 고려하자.

i) 식에서 $\nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 는 π 에 의존하지 않는다.

즉, Langevin dynamics은 $\nabla_{\mathbf{x}} \log P_{\text{data}}(\mathbf{x})$ 를 사용해 $P_{\text{data}}(\mathbf{x})$ 의 sample을 얻는데 이 과정을 통해

정학한 π 를 알 수 있고 매우 작은 step size로와 많은 수의 step을 필요로 한다.

그리므로 Langevin dynamics은 다음 그림을 보면 samples의 정학한 분포를 추장하지 못하는 것을 알 수 있다.

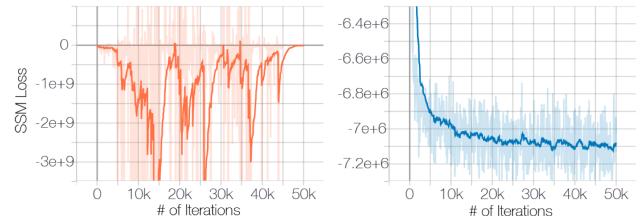


Figure 1: Left: Sliced score matching (SSM) loss w.r.t. iterations. No noise is added to data. Right: Same but data are perturbed with $\mathcal{N}(0, 0.0001)$.

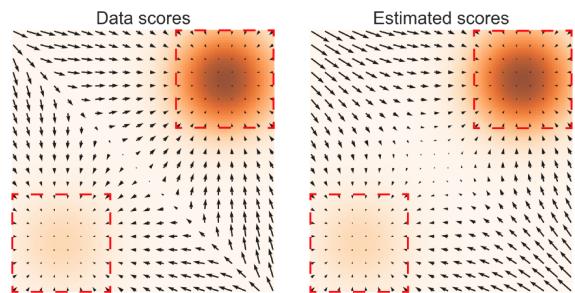


Figure 2: Left: $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$; Right: $s_{\theta}(\mathbf{x})$. The data density $p_{\text{data}}(\mathbf{x})$ is encoded using an orange colormap: darker color implies higher density. Red rectangles highlight regions where $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) \approx s_{\theta}(\mathbf{x})$.

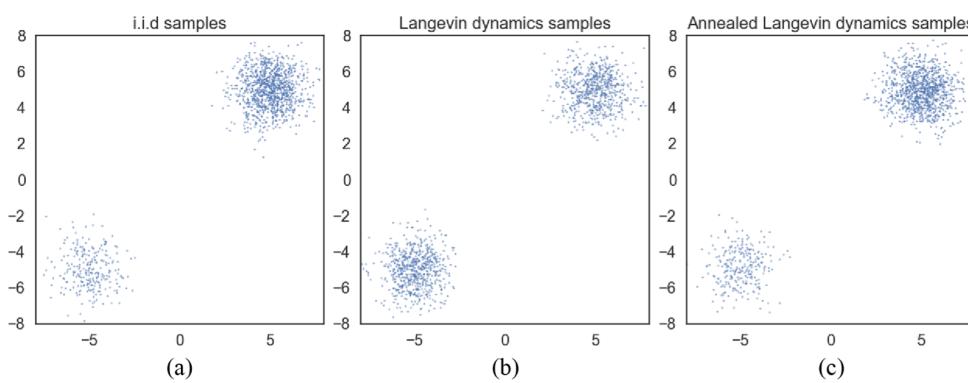


Figure 3: Samples from a mixture of Gaussian with different methods. (a) Exact sampling. (b) Sampling using Langevin dynamics with the exact scores. (c) Sampling using annealed Langevin dynamics with the exact scores. Clearly Langevin dynamics estimate the relative weights between the two modes incorrectly, while annealed Langevin dynamics recover the relative weights faithfully.

Noise Conditional Score Network : learning and inference

score-based generative modeling을 개선하기 위해 두 가지 method를 제안함

① 특정한 noise level의 noise를 사용해 data를 perturbing

② single conditional score network를 학습해 모든 noise level에 해당하는 score들을 동시에 estimating

학습 후 sample을 생성하기 위해 Langevin dynamics를 사용할 때 처음에 large noise에 해당하는 score를 생성하고 점차 noise level을 anneal down하면서 수행

o Noise Conditional Score Networks

$$\frac{\sigma_1}{\sigma_2} = \dots = \frac{\sigma_{n-1}}{\sigma_n} > 1 \text{ 을 만족하도록 } \sum_{i=1}^n \sigma_i^2 = 1 \text{ 을 조작한다.}$$

그리고 $\pi_\sigma(x) = \int p_{\text{data}}(x) N(x | \tilde{x}, \sigma^2 I) d\tilde{x}$ 는 perturbed data의 distribution 이자 핵심.

또한 Ω 은 low-dimensional manifold에서 빠져나올 정도로 충분히 커야 하고 Ω 은 data와 별 차이가 없을 정도로 충분히 작아야 한다.

→ 이를 NCSN (Noise Conditional Score Network)라고 함

우선적으로 모든 $\Omega_{i,j}^L$ 에 대해 $S_\theta(\tilde{x}, \sigma) \approx D_{\tilde{x}} \log \pi_\sigma(\tilde{x})$ 가 되도록 하면 것이다. Ω_i 를 학습한 conditional instance normalization

$S_\theta(\tilde{x}, \sigma)$ 는 dilated or atrous convolution을 포함하는 V-Net 구조로 이루어졌고 instance normalization을 채택함

o Learning NCSNs via score matching

이 논문에서는 denoising score matching의 경우와 유사하게 제작함.

하지만 sliced score matching을 사용할 가능성이.

그리고 noise distribution은 $\pi_\sigma(\tilde{x} | x) = N(\tilde{x} | x, \sigma^2 I)$ 으로 선택함.

$$\nabla_{\tilde{x}} \log \pi_\sigma(\tilde{x} | x) = -\frac{(\tilde{x} - x)^2}{\sigma^2} \text{ 이라고 할 수 있음}$$

그에 대해서 denoising score matching objective는 다음과 같이 정의함.

$$l(\theta; \sigma) = \frac{1}{2} E_{p_{\text{data}}(x)} E_{\tilde{x} \sim N(x, \sigma^2 I)} \left[\| S_\theta(\tilde{x}, \sigma) + \frac{\tilde{x} - x}{\sigma^2} \|_2^2 \right]$$

모든 σ_i 에 대해서 고려하면 다음과 같다.

$$L(\theta; \{\sigma_i\}_{i=1}^L) = \frac{1}{L} \sum_{i=1}^L \lambda(\sigma_i) \cdot l(\theta; \sigma_i)$$

여기서 $\lambda > 0$ 은 σ_i 에 대한 coefficient function이며 이 속성에서는 $\lambda(\sigma) = \sigma^2$ 을 생각

- NCSN inference via annealed Langevin dynamics

sample를 생성하기 위한 method는 annealed Langevin dynamics를 제안함

Algorithm은 다음과 같다.

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^L, \epsilon, T$.

- 1: Initialize \tilde{x}_0
- 2: **for** $i \leftarrow 1$ to L **do**
- 3: $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$ ▷ α_i is the step size.
- 4: **for** $t \leftarrow 1$ to T **do**
- 5: Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
- 6: $\tilde{x}_t \leftarrow \tilde{x}_{t-1} + \frac{\alpha_i}{2} \mathbf{s}_\theta(\tilde{x}_{t-1}, \sigma_i) + \sqrt{\alpha_i} \mathbf{z}_t$
- 7: **end for**
- 8: $\tilde{x}_0 \leftarrow \tilde{x}_T$
- 9: **end for**

return \tilde{x}_T
