Paper summary

AI VISION Lab

1. 공부한 논문의 제목, 게재된 학회 혹은 저널 등 논문 기본 정보를 적으세요.
   - **A. 이름: Maximum Likelihood Training of Score-Based Diffusion Models**
   - **B. 저널: NeurIPS**
   - **C. 도메인: Diffusion**
   - **D. 출판연도: 2021**
   - **E. 저자: Yang Song, Stefano Ermon**

2. 논문에서 제안한 알고리즘 및 프레임워크에 대해 본인이 이해한대로 다이어그램을 그려보세요. **논문 Figure를 그대로 따라 그리면 안됩니다.**
   A. [선행연구] 2021 Yang Song 논문에서는, DDPM과 SMLD를 Score Based Diffusion Model 형태로 결합한 프레임워크를 제시했음.

   B. [선행연구] 그러나, Score Based Model의 본질적인 문제는, Score Matching을 Loss로 사용하여 최적화를 한다는 점인데, 이는 Likelihood에 영향을 주지 못한다는 것임.

   C. [선행연구] 따라서, Likelihood가 필요한 Task를 해결할 수 없었을 뿐 더러, 정확한 Likelihood 값을 계산하지 못하는 문제로 인해 다른 Generative Model과의 비교가 어렵다는 한계가 있었음.

   D. 이 논문에서는 Score Based Diffusion Model의 Loss를 크게 벗어나지 않고도, Likelihood를 maximum하게 할 수 있는 Likelihood Weight를 제안함.

   E. 기존의 Score Matching에서는 Weighted Combination을 사용했는데, 이것의 역할은 모든 noise에 대해서 동일한 Weight를 주는 것을 방지하고(만약, 모든 noise에 대해서 동일한 Weight를 설정하는 경우에는 Variance가 커질 수 있음), bias를 방지하는 것임. Weighted Combination: $\lambda(t)$

   F. KL-Divergence를 Minimizing 하는 것이 Likelihood를 Maximizing하는 것과 동일하며, $\lambda(t) = \sigma^2$ 일 때, KL-Divergence는 Score Matching에 대한 Upper Bound로 작용한다는 것을 증명함.

G. 결과적으로(실험 + 증명), Likelihood를 증가시킬 수 있는 구조를 입증함

3. 본인이 생각하는 이 논문의 장점이 무엇이라고 생각하나요? **논문 Contribution bullet을 그대로 따라 적으면 안됩니다.**
   A. Score Matching을 Loss로 사용하는 경우 생기는 문제점은, Model과 특정 Distribution의 Score 값의 차이를 줄이는 것을 목표로 한다는 것임. 다시 말하자면, 해당 값을 최적화한다고 하더라도, Likelihood에는 큰 영향을 주지 못한다는 한계가 있다는 것이고, Likelihood를 직접적으로 활용하는 Task를 해결하지 못한다는 것을 뜻함.

   B. 그러나 해당 논문에서는, Likelihood Weight를 접목하여, Upper Bound를 기반으로, 해당 값을 최적화하면, Likelihood 또한 상승하도록 하는 구조를 제안함.

   C. 결과적으로, Score Matching이라는 loss의 큰 틀은 벗어나지 않은 채, Likelihood를 상승시킬 수 있는 방안을 제시함.

4. 이 논문을 읽으면서 느낀 점, 혹은 배운 점이 있으면 적어보세요.
   A. 수식 유도과정 확인

5. 이 논문의 한계점이 있다면 무엇이라고 생각하나요?
   A. Sampling을 하기 위해서는 여러 번 Model을 호출할 필요가 있음

6. 본인의 연구에 접목시켜볼 점이 있을지 생각하고 적어보세요.
   A. Diffusion 선행연구

7. 본 Summary를 작성하는 과정에서 생성형AI를 사용했나요?
   A. 아니요

날짜: 2025-07-07

이름: 신준원

Maximum Likelihood Training of Score-based Diffusion models 01/07.

Introduction

Score-based diffusion model

- Diffusing process (SDE)
  ⇒ p_{data} → $\mathcal{N}$ (noise distribution)

- reverse diffusion process (SDE)
  ⇒ SDE + score function . ⇒ CNF (continuous normalizing flow)
  ↳ 반전되는 시간변화 흐름 (random x ⇒ ODE 사용가능!)

★ tractable. likelihood 계산

* Score function

$\nabla \log p(x)$, → Gradient field (= Score) 라고

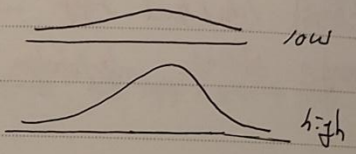
(= 방향)

real data. ← 방향 (score) → noise data

p_0 p_T

~~방향 = high score~~

~~세기 = low score~~



ex) $p(x) = \mathcal{N}(0, \sigma^2 I)$, normal distribution.

$p(x)$의 해당 Score 값.

↳ $\log p(x) = -\frac{1}{2\sigma^2}\|x\|^2 + C$

$\nabla_x \log p(x) = -\frac{x}{\sigma^2}$ ⇒ 차 크기 = score 크기 ~~방향~~

차가 작다 = score 크기 작다 ~~⇒ 세기 약~~

크기 → 절댓값

⊕ weight scale 이 필요한 이유

$N C \leq N$ → 모든 noise level에 대한 분포를 학습하기 위해선 noise마다 다른 loss update가 필요.

weighting ⇒ σ에 대해 (scale) 작은 sigma의 경우 ⇒ error가 커짐 $(\frac{1}{\sigma})$

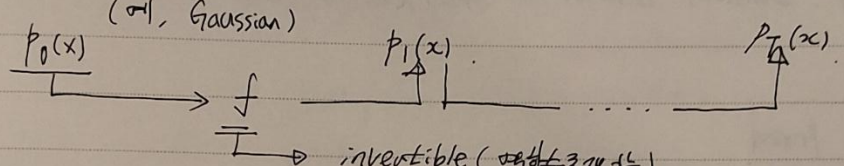
큰 sigma의 경우 ⇒ error가 작아짐 $(\frac{1}{\sigma})$

상황에 따라 보정해줄 필요가 있음 ⇒ $\lambda(\sigma)$

CNF (continuous normalization flow)

: PDF 에 대하여, 연속적인 시간변화로 모델링

Normalizing Flow (이산적인 변수 → $t, t+1 \dots$) → Continuous (★)



간단한 $p_0(x)$ distribution

⇒ 복잡한 $p_T(x)$ 분포로 변환 (with 변환 f)

↳ 복잡한 분포로 만들어가는 과정에서 필요한 것은 ODE-solver.

↳ ODE 문제

즉, 매 step 마다 적분, 매번 ODE-solver 필요.

계산량 ↑. 그러나, 정확한 계산을 기반으로.

정확한 likelihood 계산에 이용가능

그렇다면 Score matching Loss도 ?

→ 훌륭한 효율적인 학습가능 (그러나 → 정확한 likelihood 계산 불가)

* likelihood 를 사용하는 (필요) task에 적용이 어렵다는 한계존재.

결론 → weighted Combination 사용 ⇒ likelihood와 무관하게 학습하된 SM_Loss

⇒ likelihood 값을 추정적 사용가능!

denotation

$p(x)$: dataset distribution ($p_0 = p$)

$p_t(x)$: marginal distribution of $x(t)$

$p_{0t}(x'|x)$: transition distribution from $x(0)$ to $x(t)$

Diffusing process

목표 : $p(x) \longrightarrow$ noise distribution.

식표현 : SDE (stochastic differential equation)

$$dx = \underbrace{f(x,t)}_{\text{drift}}dt + \underbrace{g(t)}_{\text{diffusion}}dw \quad \text{(dw: Wiener)}$$

Coefficient

$p(x) + dx \rightarrow \{x(t)\}_{t\in[0,T]} \longrightarrow \pi(x) \sim N(0, I)$.

tractable, reverse를 간단히!

⇒ 따라서, 분포를 완벽히 복원. ($p(x)$)

* Diffusion process

→ $f(x,t)$, $g(t)$, T 를 수정 가능

이에, 어떻게 수정하느냐에 따라서

VE

VP

sub-VP

) 3 가지이다.

Sampling with Reverse SDE

reverse time diffusion process

$$dx = \left[f(x,t) - g(t)^2 \nabla_x \log p_t(x) \right] dt + g(t) \cdot d\bar{w}$$

$d\bar{w}$: wiener process (reverse)

diffusion $x_0 \to x_1 \to x_2 \to \cdots \to x_T$

reverse $x_0 \leftarrow x_1 \leftarrow x_2 \leftarrow \cdots \leftarrow x_T$

reverse → target : $\nabla_x \log p_t(x)$ 추론하기

→ $S_\theta(x,t)$: model. (Score-based model)

잘 학습된 모델의 경우 $\nabla_x \log p_t(x) \approx S_\theta(x,t)$

model 을 parameterize
- likelihood-base : 분포 O
- Score-function base : 분포 X ⇒ 확률 구할 수 없음.

일반적인 SBM 의 Loss

$$J_{SM}(\theta;\lambda(\cdot)) := \frac{1}{T}\int_0^T E_{p_t(x)}\left[\lambda(t) \left\| \nabla_x \log p_t(x) - S_\theta(x,t) \right\|_2^2 \right] dt$$

$\lambda(t)$: (Positive) weighting function

$\underline{\arg\min J_{SM}(\theta;\lambda(\cdot))}$ ⇒ training SBM.

→ dx = affine 형태 ⇒ $p_{0t}(x'|x)$ = Gaussian.

→ SM 적용 가능

→ sampling reverse-SDE 사용

(with SDE-solver)

Likelihood of SBDM

Diffusion process
Reverse process
→ likelihood 정의해보기.

① $p_\theta^{SDE}(x)$

→ $d\hat{x} = \left[f(\hat{x}, t) - g(t)^2 s_\theta(\hat{x}, t) \right] dt + g(t) d\bar{w}$ ~~$\hat{x}$~~ $\hat{x}_\theta(T) \sim \pi$.

$p_\theta^{SDE}(x) \Rightarrow \hat{x}_\theta(0) \sim p_\theta^{SDE}$ (with. SDE-solver)

SDE의 경우 정확한 계산이 불가능함.
⇒ Lower-bound 이용

② $p_\theta^{ODE}(x)$ → (SDE); ~~주로 부터 파생~~ = $p_t(x)$ 공유가능.
→ ~~data~~ → marginal.

$$\frac{dx}{dt} = f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x)$$

특징 : deterministic ★
→ $f(x)$ — $f^{-1}(x)$ ⇒ 정확한 계산 가능) 정확한 likelihood 계산가능
(계산량 ↑, T)

Bounding likelihood

① p_θ^{SDE} → Likelihood 계산 x
② p_θ^{ODE} → Likelihood 계산 o but expensive.

⇓

Score match loss → 계산 easy, 그러나 likelihood 높이지 못함.

KL divergence

maximizing Likelihood = minimizing D_{KL} * main idea

정리 1

$$D_{KL}(p \,\|\, p_\theta^{SDE}) \le J_{SM}(\theta; g(\cdot)^2) + D_{KL}(p_T \,\|\, \pi)$$

upper bound — diffusion 계수, $\lambda(t) = g(t)^2$ 일 때

증명 ① denote μ : joint distribution of diffusion process (dx) $x(t) \to \{x(t)\}_{t\in[0,T]}$
ν : joint distribution of process $\hat{x}_\theta(t) \to \{\hat{x}_\theta(t)\}_{t\in[0,T]}$

② $D_{KL}(\mu \,\|\, \nu) \ge D_{KL}(p \,\|\, p_\theta^{SDE})$
모든 경로 — 일부 of 경로

③ $D_{KL}(\mu \,\|\, \nu) = D_{KL}(p_T \,\|\, \pi) + E_{p_T(z)}[D_{KL}(\mu(\cdot \mid x(T)=z) \,\|\, \nu(\cdot \mid \hat{x}_\theta(T)=z))]$

요소(결과) ① π 고정.

→ (π 고정인 경우)

결과 $-E_{p(x)}[\log p_\theta^{SDE}(x)] \le J_{SM}(\theta; g(\cdot)^2) + C_1 = J_{DSM}(\theta; g(\cdot)^2) + C_2$.

⇒ $\lambda(t) = g(t)^2$ = likelihood weighting

정리 1

$p(x)$, $q(x)$, p_t, q_t, $q_T = \pi$, $s_\theta(x,t) = \nabla_x \log q_t(x)$ 이면

$$p_\theta^{SDE} = p_\theta^{ODE} = q$$

$$D_{KL}(p \| p_\theta^{SDE}) = J_{SM}(\theta; q(\cdot,T)^2) + D_{KL}(p_T \| \pi)$$

증명 $s_\theta(x,t) = \nabla_x \log q_t(x)$ 일 때, $\rightarrow p_\theta^{SDE} = q$

즉, $\nabla_x \log q_t(x)$ 이므로, $p_\theta^{SDE} = p_\theta^{ODE} = q$

⊕ Fokker-Planck equation

그래서, $s_\theta(x,t) = \nabla_x \log q_t(x)$ 여러움

⊕ p^{SDE}, p^{ODE} 에만 좋은 likelihood 가 좋은지 보장 못함.

⇒ p_θ^{SDE} 와 p_θ^{ODE} 는 거의 비슷 ($s_\theta(x,t) \approx \nabla_x \log p_t(x)$ 잘 추정한 경우)

⊕ 장점) training with likelihood weight.

⇒ p^{ODE}, p^{SDE}, p 의 likelihood 높음 +

datapoint 에서 log-likelihood

$J_{SM} \rightarrow \nabla_x \log p_x(x)$ 에 대한 식 $\Rightarrow$ $J_{DSM}(\theta; \lambda(\cdot))$ 사용

따라서, Theorem 1 $\Rightarrow$ training에서만 이용 $\Rightarrow$ datapoint 에서 추정하는 방법

Theorem 1 (대체 가능)

$$-\log p_\theta^{SDE}(x) \leq L_\theta^{SM}(x) \doteq L_\theta^{DSM}(x)$$

증명 목표 $p \rightarrow -E_{p(x)}[\log p_\theta^{SDE}(x)] = D_{KL}(p \| p_\theta^{SDE}) + H(p)$

Numerical stability (수치적 안정성)

instability 한 경우, 작은 오차 $\Rightarrow$ 큰 영향력 존재.

$t \rightarrow 0$ (작은 noise) 인 때, instability 하다고 할 수 있음.

해결방안 $\rightarrow$ 0이 아닌, 0에서 시작. 즉 $t \in [\varepsilon, T]$