Diffusion Models as Plug-And-Play Priors

Sunday, November 13, 2022 3:42 AM

- This note is written as I read "Diffusion models and plug-and-play priors" by Graikos et al.
- Paper link: https://arxiv.org/abs/2206.09012

- We have a prior $p(x)$ on high-dimensional data such as images.
- We are given a constraint $c(x,y)$.
- We want to perform "inference" on a model that involve $p(x)$ and $c(x,y)$.
- That is, we want to find an approximation to $p(x|y) \propto p(x)\,c(x,y)$.
- In our case, the prior is a DDPM: $p(\underbrace{x^{(0)}}_{\text{"}x\text{"}}, \underbrace{x^{(1)}, x^{(2)}, \dots, x^{(T)}}_{h})$

  pretrained and cannot be changed

  treated as latent variables

- What is $c(x,y)$? It can be:

  $\Rightarrow$ $c(x,y)$ = probability that image x has class y.

  $\Rightarrow$ $c(x,y)$ = a constraint so that image x matches a segmentation labeling y

---

Variation Inference

- We are given a posterior distribution

$$p(x|y) = \frac{p(x)\,c(x,y)}{Z(y)}$$

where $Z(y) = \int p(x)\,c(x,y)\,dx$ is a normalization constant.

- We want to approximate $p(x|y)$ with a tractable distribution $q(x)$
- So, we seek to minimize $D_{KL}(q(x) \,\|\, p(x|y))$, which is given by

$$\begin{aligned} D_{KL}(q(x) \,\|\, p(x|y)) &= \int q(x) \log \frac{q(x)}{p(x,y)}\,dx \\ &= E_{x\sim q(x)}\left[\log q(x) - \log p(x,y)\right] \end{aligned}$$

constant

$$E_{x\sim q(x)}[\ \cdots\ ]$$

$$= -E_{x\sim q(x)}\left[\log p(x) + \log C(x,y) - \overbrace{\log Z(y)} - \log q(x)\right]$$

$$= \underbrace{-E_{x\sim q(x)}\left[\log p(x) + \log C(x,y) - \log q(x)\right]}_{F \,=\, \text{free energy}} + C$$

- When $q(x)$ is expressive enough to capture $p(x,y)$, we have that $q(x) = p(x|y)$ when F is optimized.
- Otherwise, $q(x)$ will capture a mode-seeking approximation to $p(x|y)$.
  - ↳ If $q(x)$ is Dirac delta, then the optimized $q(x)$ will be located at the mode of $p(x|y)$.

Variational inference with latent variable model

- We consider the special case where $p(x)$ is a latent variable model $p(x,h)$ with $p(x) = \int p(x,h)\,dh = \int p(x|h)\,p(h)\,dh$.
- The problem now is that $p(x)$ becomes hard to approximate.
- The solution is to introduce another component to the tractable distribution.
- Let $q(h|x)$ be a tractable approximate to the posterior $p(h|x)$.
- In this way,

$$p(x) = \int p(x|h)\,p(h)\,dh$$

$$= \int q(h|x)\,\frac{p(x|h)\,p(h)}{q(h|x)}\,dh$$

$$= E_{h\sim q(h|x)}\left[\frac{p(x|h)\,p(h)}{q(h|x)}\right].$$

- By Jensen's inequality,

$$\log p(x) = \log E_{h\sim q(h|x)}\left[\frac{p(x|h)\,p(h)}{q(h|x)}\right] = \log E_{h\sim q(h|x)}\left[\frac{p(x,h)}{q(h|x)}\right]$$

$$\log p(x) = \log E_{h\sim q(h|x)}\left[\frac{p(x|h)\,p(h)}{q(h|x)}\right] = \log E_{h\sim q(h|x)}\left[\frac{p(x,h)}{q(h|x)}\right]$$

$$\geq E_{h\sim q(h|x)}\left[\log \frac{p(x,h)}{q(h|x)}\right]$$

$$= E_{h\sim q(h|x)}\left[\log p(x,h) - q(h|x)\right]$$

- Now, consider the free energy F

$$F = -E_{x\sim q(x)}\left[\log p(x) + \log c(x,y) - \log q(x)\right]$$

$$= -E_{x\sim q(x)}\left[\log p(x) - \log q(x)\right] - E_{x\sim q(x)}\left[\log c(x,y)\right]$$

$$\leq -E_{x\sim q(x)}\left[E_{h\sim q(h|x)}\left[\log p(x,h) - \log q(h|x)\right] - \log q(x)\right]$$
$$- E_{x\sim q(x)}\left[\log c(x,y)\right].$$

$$= -E_{x\sim q(x)}\left[E_{h\sim q(h|x)}\left[\log p(x,h) - \log q(h|x)\right] - E_{h\sim q(h|x)}\left[\log q(x)\right]\right]$$
$$- E_{x\sim q(x)}\left[\log c(x,y)\right]$$

$$= -E_{x\sim q(x),\, h\sim q(h|x)}\left[\log p(x,h) - \log q(h|x) - \log q(x)\right] - E_{x\sim q(x)}\left[\log c(x,y)\right]$$

$$= -E_{x\sim q(x),\, h\sim q(h|x)}\left[\log p(x,h) - \log q(h|x)\,q(x)\right] - E_{x\sim q(x)}\left[\log c(x,y)\right]$$

- Equation (2) of the paper says that F = RHS above.

However, this is not the case!!! The paper was sloppy.

It should be

$$F \leq \underbrace{-E_{x\sim q(x),\, h\sim q(h|x)}\left[\log p(x,h) - \log q(x)\,q(h|x)\right] - E_{x\sim q(x)}\left[\log c(x,y)\right]}_{\text{ELBO}(x) \text{ or } \text{VLB}(x)}$$

"evidence lower bound" "variational lower bound"

Application to DDPM

- For a DDPM, we have that

$$p(x^{(T)}, x^{(T-1)}, \ldots, x^{(0)}) = p(x^{(T)}) \prod_{t=1}^{T} p(x^{(t-1)} | x^{(t)})$$

where

$$p(x^{(t-1)} | x^{(t)}) = N\Big(x^{(t-1)};\ \underbrace{\mu_\theta(x^{(t)}, t)}_{\text{neural network}},\ \underbrace{\Sigma_\theta(x^{(t)}, t)}_{\sigma_t^2 I}\Big), \quad 1 \le t \le T$$

$$p(x^{(T)}) = N(x^{(T)};\ 0, I)$$

$\sigma_t^2 I$ ↳ fixed function of t.

- The DDPM is trained so that

$$q(x^{(t)} | x^{(t-1)}) = N(x^{(t)};\ \sqrt{1-\beta_t}\, x^{(t-1)}, \beta_t I).$$

So, we should use

$$q(h = (x^{(T)}, x^{(T-1)}, \dots, x^{(1)}) | x) = \prod_{t=1}^{T} q(x^{(t)} | x^{(t-1)}).$$

- The simplest approximation $q(x)$ that we can use is the Dirac delta

$$q(x) = \delta(x - \eta).$$

which simply sets $x^{(0)} = \eta$.

- So, with $q(x) = \delta(x-\eta)$

$$-E_{x\sim q(x),\, h\sim q(x)}\big[\log p(x,h) - \log q(x) q(h|x)\big] - E_{x\sim q(x)}\big[\log c(x,y)\big]$$

$$= \underbrace{-E_{h\sim q(\eta)}\big[\log p(\eta, h) - \log q(h|\eta)\big]}_{L} - \log c(\eta, y)$$

- We now must optimize η so that $L - \log c(\eta, y)$ is minimized.

- From the standard derivation of the ELBO of a DDPM, we have that

$$L = \underbrace{-E_{x^{(T)}\sim q(x^{(T)}|\eta)}\left[\log \frac{p(x^{(T)})}{q(x^{(T)}|\eta)}\right]}_{L_T}$$

$$\underbrace{- E_{x^{(1)}\sim q(x^{(1)}|\eta)}\big[\log p(\eta | x^{(1)})\big]}_{L_0}$$

$$- \sum_{t=2}^{T} \underbrace{E_{x^{(t)}\sim q(x^{(t)}|\eta)}\big[D_{KL}(q(x^{(t-1)}|x^{(t)}, \eta) \,\|\, p(x^{(t-1)}|x^{(t)}))\big]}_{L_{t-1}}$$

- Now, following the (non-standard) derivation in my note

- Now, following the (non-standard) derivation in my note (https://pkhungurn.github.io/notes/notes/ml/ddpm/ddpm.pdf), we have that

$$\underbrace{L_0 + \sum_{t=2}^{T} L_{t-1}}_{L_{t-1}} = \sum_t w(t)\, E_{\xi \sim N(0,I)}\left[\|\xi - \xi_\theta(x^{(t)}, t)\|^2\right]$$

where $x^{(t)} = \sqrt{\bar{\alpha}_t}\,\eta + \sqrt{1-\bar{\alpha}_t}\,\xi$, $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{t=1}^{T} \alpha_t$, and $w(t)$ is a weight function.

- The paper then makes more simplifications
  1. It drops the L_T term even though it has dependency on η
  2. It drops the $w(t)$ term to simplify the loss as Ho et al. does in their famous DDPM paper.

So, $$L - \log c(\eta, y) \approx \underbrace{\left(\sum_{t=1}^{T} E_{\xi \sim N(0,I)}\left[\|\xi - \xi_\theta(x^{(t)}, t)\|^2\right]\right) - \log c(\eta, y)}_{L_{simple}}$$

- To repeat, we perform inference by optimizing L_{simple} with respect to η.
- The paper gives a sample optimization algorithm.

  Input: pretrained $\xi_\theta(\cdot,\cdot)$, data y, contraint $c(\cdot,\cdot)$, learning rate λ

  Initialized $x \sim N(0, I)$

  for $i = T$ downto 1

  Sample $\xi \sim N(0, I)$

  $x_{t_i} \leftarrow \sqrt{\bar{\alpha}_{t_i}}\, x + \sqrt{1-\bar{\alpha}_i}\, \xi$

  $x \leftarrow \lambda \nabla_x \left(\|\xi - \xi_\theta(x^{(t_i)}, t_i)\|^2 - \log c(x, y)\right)$

  end for

  output $\eta = x$

Not that the timestep is not simple $t = T, T-1, \dots, 1$, but $t = t_T, t_{T-1}, \dots, t_1$

So, there's a flexibility in choosing the timestep.

- The above algorithm can be changed in various ways.
  - ⇒ How to initialize x
  - ⇒ The number of time steps T
  - ⇒ The timesteps used $t_T, t_{T-1}, t_{T-2}, \ldots, t_1$
  - ⇒ How x is accumulated.
    - ↳ The above is based on SGD, but we may do something else.

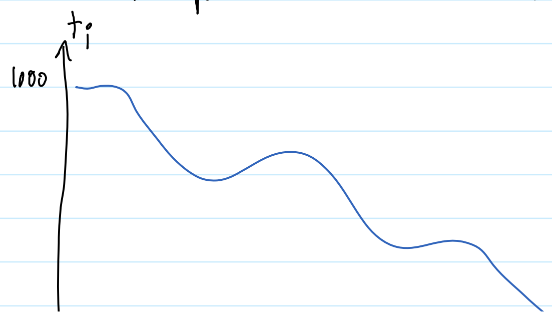
- Note that the optimization is stochastic.
  Each time it is run it produces a different output.

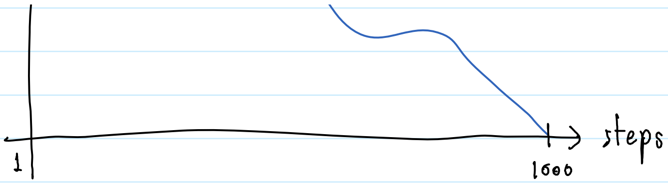
- On the timestep used.
  - ⇒ The paper observed that, when optimizing for all timestep at once, it is hard for the sample to reach a mode.
  - ⇒ So, they suggest annealing: t_i should generally be decreasing.
    - ↳ First few iterations should coarsely explore the search space.
    - ↳ Later iterations should be at lower temperature to home in on a local minimum.

Conditional image generation ①: MNIST

the "DDPM beats GAN" paper

- The paper trains Dhariwal and Nichol (2021) on MNIST.
- They experimented with a number of constraints $c(x, y)$
  - ⇒ "thin digit" = negative of average density of image
  - ⇒ "thick digit" = average intensity.
  - ⇒ "vertical symmetry", "horizontal symmetry" = negative distance between two halves
  - ⇒ "class" = "generate digit 3"
- Inference was performed with the ADAM optimizer with learning rate 10^{-2}
- The timesteps is cosine-modulated linearly decreasing.





- It is also important to reduce the weight of the constraint $\log c(x,y)$ as we progress through the algorithm. So, they set the weight for that term to linearly decrease from $w_T = 10^{-2}$ to $w_1 = 0$. Not doing so results in poor sample quality.

Conditional image generation ② : FFHQ

- Use ① DDPM for 256×256 FFHQ from Baranchuk et al. (https://arxiv.org/abs/2112.03126)

  ② pretrained ResNet-18 face attribute classifier on CelebA (https://arxiv.org/abs/1411.7766)

- Attributes classified included "no beard", "smiling", "blonde hair", "male"

- The paper select a collection of attributes $y = \{y_1, y_2, .., y_k\}$ and enforce the attributes with constraint

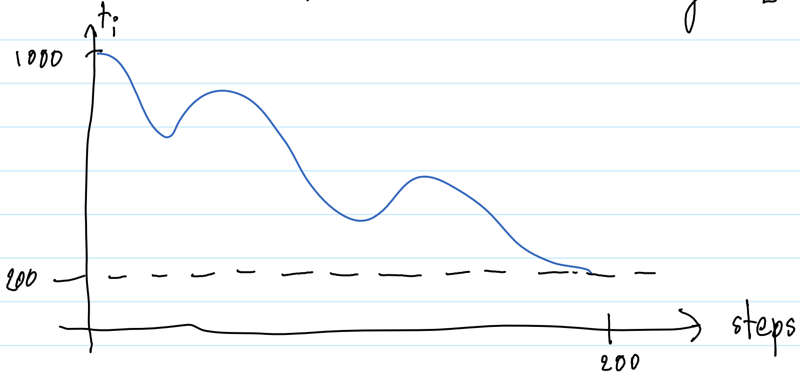
$$c(x,y) = \prod_{i=1}^{k} \log \overbrace{p(y_i \mid x)}^{\text{classifier for attribute } y_i}$$

- Inference algorithm

  ⇒ Run optimization with Adamax with $\beta_1 = 0.9$, $\beta_2 = 0.999$ for 200 steps

  ⇒ Decrease learning rate from 1 to 0.5

  ⇒ Timestep schedule is again cosine modulated linearly decresing but the t values are in the range [200, 1000].



  ⇒ Balancing diffusion loss (L) with $\log c(x,y)$ was difficult.

200

⇒ Balancing diffusion loss (L) with $\log c(x,y)$ was difficult.

⇒ So, clip the gradient norm of $\log c(x,y)$ to half of diffusion loss's gradient norm.

⇒ After the 200 Adamax step, the image is still noisy, so the paper just run the DDPM from $t=200$ to denoise the sample.

Comments

- The paper also provides experiments on semantic segmentation and solving TSP. However, I did not read them in details.

- The framework introduced by the paper is versatile
  ↳ Can use off-the-shelf DDPM.
  ↳ Constraints can be anything including off-the-shelf classifier
  ↳ as long as it is differentiable

- However, it is hard to apply.
  ↳ Need to finetune algorithm for each problem instance.

- No guarantees on generated sample quality.