How to Train Unnormalized Probabilistic Models

Presenter: Zijing Ou

June 27, 2021

Unnormalized Probabilistic Model

Unnormalized Probabilistic Models (UPMs) specify probabilistic desity or mass functions up to an unknown normalizeing constant

$$p_\theta(x) = \frac{1}{Z_\theta} E_\theta(x),$$

where Z_θ, known as the partition function, is defined as

$$Z_\theta = \int E_\theta(x) dx.$$

Since without placing a restriction on the normalizing constant, UPMs are more flexible and can model a more expressive family of probability distributions.

Training UPMs

Given a set of training data $X = \{x_1, \ldots, x_N\}$, the model parameters θ can be optimized by minimizing the negative log likelihood

$$\mathcal{L}(\theta) := \log Z_\theta - \frac{1}{N} \sum_{i=1}^{N} \log E_\theta(x_i). \tag{1}$$

The difficulty rises here is how to handle with the normalizing constant $\log Z_\theta$. To address this issue, we focus on

- Contrastive Divergence
- Noise Contrastive Estimation
- Score Matching

Maximum Likelihood Training with MCMC

The gradient of (1) can be written as

$$\begin{aligned}\nabla_\theta \mathcal{L}(\theta) &= \nabla_\theta \log Z_\theta - \nabla_\theta \frac{1}{N}\sum_{i=1}^{N} \log E_\theta(x_i) \\ &= \int \frac{E_\theta(x)}{Z_\theta} \nabla_\theta \log Z_\theta dx - \mathbb{E}_{p_d(x)}\left[\nabla_\theta \log E_\theta(x)\right] \\ &= \mathbb{E}_{p_\theta(x)}\left[\nabla_\theta \log E_\theta(x)\right] - \mathbb{E}_{p_d(x)}\left[\nabla_\theta \log E_\theta(x)\right].\end{aligned}$$

The hardness arises at sampling from $p_\theta(x)$, since we cannot obtain its close form due to the notorious partition function.

Maximum Likelihood Training with MCMC

For sampling from $p_\theta(x)$, we resort to MCMC sampling. Fpr examples, we can use Langevin MCMC

$$x^{k+1} \leftarrow x^k + \frac{\epsilon^2}{2} \underbrace{\nabla_x \log p_\theta(x^k)}_{=\nabla_x \log E_\theta(x)} + \epsilon z^k, k = 0, 1, \ldots, K-1.$$

When $\epsilon \to 0$ and $K \to \infty$, x^K is guaranteed to distributed as $p_\theta(x)$ under some regularity conditions.

However, running MCMC till convergence to obtain a samle $x \sim p_\theta(x)$ can be computationally expensive.

Contrastive Divergence

Objective.

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_\theta(x)} \left[\nabla_\theta \log E_\theta(x)\right] - \mathbb{E}_{p_d(x)} \left[\nabla_\theta \log E_\theta(x)\right]$$

Contrastive Divergence (CD) is a popular approximation method to make MCMC-based learning practical.

- In CD, the initial sample is from the empirical data distribution: $x^{(0)} \sim p_d(x)$.
- Then we apply k-step MCMC iteration to generate $x^{(k)}$ for $p_\theta(x)$, which has been turn out that $\lim_{k \to \infty} x^{(k)} \sim p_\theta(x)$.
- The gradient can be approximated by CD-k estimator

$$\nabla_\theta \mathcal{L}(\theta) \approx \nabla_\theta \log E_\theta(x^{(k)}) - \nabla_\theta \log E_\theta(x^{(0)}).$$

Actually, CD-1 works well.

Understanding Contrastive Divergence

Maximizing likelihood is equivalent to minimizing the KL divergence between $p_d(x)$ and $p_\theta(x)$, because

$$\begin{aligned} -\mathbb{E}_{x\sim p_d(x)}\left[\log p_\theta(x)\right] &= KL(p_d(x)||p_\theta(x)) - \mathbb{E}_{x\sim p_d(x)}\left[\log p_d(x)\right] \\ &= KL(p_d(x)||p_\theta(x)) - \text{constant}. \end{aligned}$$

Why CD works? Why CD is called divergence?

- Denoting $p_\theta^{(k)}(x)$ is the distribution at k-th MCMC iteration, then in CD, we have $p_\theta^{(0)}(x) = p_d(x)$, $p_\theta^{(\infty)}(x) = p_\theta(x)$.
- Optimizing the objective of CD-k is equivalent to

$$\begin{aligned} &\operatorname*{argmin}_\theta \mathbb{E}_{x^{(0)}\sim p_d(x)}\left[\log E_\theta(x^{(k)}) - \log E_\theta(x^{(0)})\right] \\ \Leftrightarrow &\operatorname*{argmin}_\theta \underbrace{KL(p_\theta^{(0)}(x)||p_\theta^{(\infty)}(x)) - KL(p_\theta^{(k)}(x)||p_\theta^{(\infty)}(x))}_{\text{contrastive divergence}}. \end{aligned}$$

Unbiased Contrastive Divergence

Objective.

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_\theta(x)}\left[\nabla_\theta \log E_\theta(x)\right] - \mathbb{E}_{p_d(x)}\left[\nabla_\theta \log E_\theta(x)\right]$$

The MCMC approximation of $\mathbb{E}_{\mathcal{M}}[f(x)]$ is biased, where $\mathcal{M}$ is $p_\theta(x)$ and $f(x) = \nabla_\theta \log E_\theta(x)$ in CD. The question here is

Can we construct an unbiased MCMC estimation?

This can be achieved by introducing another Markov chain.

Unbiased Contrastive Divergence

If there exists two Markov chains $\{a_t\}$ and $\{b_t\}$ such that

- $\mathbb{E}[f(a_t)] \to \mathbb{E}[f(x)]$ as $t \to \infty$;
- $\mathbb{E}[f(a_t)] = \mathbb{E}[f(b_t)]$ for all $t \geq 0$;
- For same random time τ, $a_t = b_{t-1}$ for all $t \geq \tau$.

Then we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{M}}[f(x)] &= \mathbb{E}_{\mathcal{M}}\left[f(a_1) + \sum_{t=2}^{\infty}(f(a_t) - f(a_{t-1}))\right] \\
&= \mathbb{E}_{\mathcal{M}}\left[f(a_1) + \sum_{t=2}^{\infty}(f(a_t) - f(b_{t-1}))\right] \\
&= \mathbb{E}_{\mathcal{M}}\left[f(a_1) + \sum_{t=2}^{\tau-1}(f(a_t) - f(b_{t-1}))\right].
\end{aligned}$$

However, the construction of chains is a highly non-trivial task.

Adversarially Variational Inference

$$
\begin{aligned}
\mathbb{E}_{p_d(x)}[\log p_\theta(x)] &= \mathbb{E}_{p_d(x)}[\log E_\theta(x)] - \log Z_\theta \\
&= \mathbb{E}_{p_d(x)}[\log E_\theta(x)] - \log \int q_\phi(x) \frac{E_\theta(x)}{q_\phi(x)} dx \\
&\leq \mathbb{E}_{p_d(x)}[\log E_\theta(x)] - \int q_\phi(x) \log \frac{E_\theta(x)}{q_\phi(x)} dx \\
&= \mathbb{E}_{p_d(x)}[\log E_\theta(x)] - \mathbb{E}_{q_\phi(x)}[\log E_\theta(x)] - H(q_\phi(x))
\end{aligned}
$$

Adversarial Training.

$$
\max_\theta \min_\phi \mathbb{E}_{q_\phi(x)}[\log E_\theta(x)] - \mathbb{E}_{p_d(x)}[\log E_\theta(x)] - H(q_\phi(x))
$$

Score Matching

The basic score matching objective minimizes a discrepancy between tow distribution called the Fisher divergence

$$D_F(p_d(x)||p_\theta(x)) := \mathbb{E}_{p_d(x)}\left[\frac{1}{2}||\nabla_x \log p_d(x) - \nabla_x \log p_\theta(x)||^2\right]. \quad (2)$$

- The first-order gradient function of a log-PDF is called the *score* of that distribution. Thus (2) is also known as *score matching*.
- When $D_F(p_d(x)||p_\theta(x)) = 0$, then we have $p_\theta(x) = p_d(x)$.
- Note that $\nabla_x \log p_\theta(x) = \nabla_x \log E_\theta(x) - \nabla_x \log Z_x = \nabla_x \log E_\theta(x)$, thus we can ignore the normalzing term during training.
- However, the second term is generally imparactical to calculate since $\log p_d(x)$ is unknown.

Score Matching

$$
\begin{aligned}
&D_F(p_d(x)||p_\theta(x)) \\
&= \frac{1}{2}\int p_d(x)\left(\nabla_x \log p_d(x) - \nabla_x \log p_\theta(x)\right)^2 dx \\
&= \frac{1}{2}\int p_d(x)(\nabla_x \log p_d(x))^2 dx + \frac{1}{2}\int p_d(x)(\nabla_x \log p_\theta(x))^2 dx \\
&\quad \underbrace{-\int p_d(x)\nabla_x \log p_d(x)\nabla_x \log p_\theta(x) dx}_{\mathcal{I}}
\end{aligned}
$$

Integrate by parts:

$$
\begin{aligned}
\mathcal{I} &= -\int \nabla_x p_d(x)\nabla_x \log p_\theta(x) dx \\
&= -p_d(x)\nabla_x \log p_\theta(x)|_{x=-\infty}^{\infty} + \int p_d(x)\nabla_x^2 \log p_\theta(x) dx \\
&= \int p_d(x)\nabla_x^2 \log p_\theta(x) dx
\end{aligned}
$$

Score Matching

Score Matching.

$$D_F(p_d(x)||p_\theta(x)) := \mathbb{E}_{p_d(x)}\left[\frac{1}{2}||\nabla_x \log p_d(x) - \nabla_x \log p_\theta(x)||^2\right]$$
$$= \mathbb{E}_{p_d(x)}\left[\frac{1}{2}||\nabla_x \log p_\theta(x)||^2 + tr(\nabla_x^2 \log p_\theta(x))\right] + \text{const}$$

An important downside is that the computation of Hessian matrix is expensive, thus does not scale to high dimensionality.

Denoising Score Matching

The main issue that arises in SM is that it only works with continuously differentiable $\log p_d(x)$.

However, these conditions may not hold in practice. To alleviate this difficulty, one can:

- Add a bit of noise to each datpoint: $\tilde{x} = x + \epsilon$, $\epsilon \sim p(\epsilon)$;
- The resulting noisy data distribution is $q(\tilde{x}) = \int q(\tilde{x}|x)p_d(x)dx$ is smooth;
- The Fisher divergence $D_F(q(\tilde{x}|x)||p_\theta(\tilde{x}))$ is a proper objective.

Denoising Score Matching

Denoising Score Matching.

$$D_F(q(\tilde{x})||p_\theta(\tilde{x})) := \mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}||\nabla_x \log q(\tilde{x}) - \nabla_x \log p_\theta(\tilde{x})||^2\right]$$
$$= \mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}||\nabla_x \log q(\tilde{x}|x) - \nabla_x \log p_\theta(\tilde{x})||^2\right] + \text{const.}$$

The main drawback is $D_F(q(\tilde{x})||p_\theta(\tilde{x})) \neq D_F(p_d(x)||p_\theta(x))$. One way to attenuate the inconsistency is to use a small noise perturbation, such that $q(\tilde{x}) \approx p_d(\tilde{x})$.
As an example, suppose $q(\tilde{x}|x) = \mathcal{N}(\tilde{x}|x, \sigma^2 I)$ and $\sigma \approx 0$, we have

$$D_F(q(\tilde{x})||p_\theta(\tilde{x})) = \mathbb{E}_{p_d(x)}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\left[\frac{1}{2}||z/\sigma + \nabla_x \log p_\theta(x + \sigma z)||^2\right]$$

Denoising Score Matching

Denoising Score Matching.

$$D_F(q(\tilde{x})||p_\theta(\tilde{x})) = \mathbb{E}_{p_d(x)}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\left[\frac{1}{2}||z/\sigma + \nabla_x \log p_\theta(x+\sigma z)||^2\right]$$

- Denoising score matching generally suffers from high variance when $\sigma \approx 0$.
- Note that $\mathbb{E}_{x,z}\left[2z^T\nabla_x \log p_\theta(x)/\sigma\right] = 0$ and $\mathbb{E}_z\left[||z||^2/\sigma^2\right] = d/\sigma^2$.
- We can construct a score function to reduce variance

$$c_\theta(x,z) = \mathbb{E}_{p_d(x)}\mathbb{E}_{z\sim\mathcal{N}(0,I)}\left[\frac{2}{\sigma}z^T\nabla_x \log p_\theta(x) + \frac{||z||^2}{\sigma^2} - \frac{d}{\sigma^2}\right].$$

Sliced Score Matching

By adding noise to data, DSM avoids the expensive computation of second-oder derivatives. However, the objective of DSM corresponds to the distribution of noise-perturbed data $q(\tilde{x})$, not the original noise-free data distribution $p_d(x)$.

Sliced Score Matching is one alternative to Denoising Score Matching that is both consistent and computationally efficient.

Sliced Score Matching.

$$D_{SF}(p_d(x)||p_\theta(x)) := \mathbb{E}_{p_d(x)}\mathbb{E}_{p(v)}\left[\frac{1}{2}(v^T\nabla_x \log p_d(x) - v^T\nabla_x \log p_\theta(x))^2\right]$$
$$= \mathbb{E}_{p_d(x)}\mathbb{E}_{p(v)}\left[v^T\nabla_x^2 \log p_\theta(x) v + \frac{1}{2}||\nabla_x \log p_\theta(x)||^2\right]$$

where $p(v)$ is a noise distribution, *e.g.*, the standard Gaussian.

Sliced Score Matching

Sliced Score Matching.

$$D_{SF}(p_d(x)||p_\theta(x)) = \mathbb{E}_{p_d(x)}\mathbb{E}_{p(v)}\left[v^T\nabla_x^2 \log p_\theta(x) v + \frac{1}{2}||\nabla_x \log p_\theta(x)||^2\right]$$

Hessian-vector products:

$$\begin{aligned} v^T\nabla_x^2 \log p_\theta(x) v &= \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2 E_\theta(x)}{\partial x_i \partial x_j} v_i v_j \\ &= \sum_{i=1}^{d} \frac{\partial}{\partial x_i} \underbrace{\left(\sum_{j=1}^{d} \frac{\partial E_\theta(x)}{\partial x_j} v_j\right)}_{:=f(x)} v_i, \end{aligned}$$

where $f(x)$ is the same for different values of i.

Score Matching & Contrastive Divergence

Recall the Langevin MCMC method

$$x^{k+1} \leftarrow x^k - \frac{\epsilon^2}{2}\nabla_x \log E_\theta(x) + \epsilon z^k.$$

Contrastive Divergence with 1 step Langevin MCMC

$$\begin{aligned}
-\mathbb{E}_{p_d}[\nabla_\theta \log p_\theta(x)] &= \mathbb{E}_{p_d}[\nabla_\theta \log E_\theta(x)] - \mathbb{E}_{p_\theta}[\nabla_\theta \log E_\theta(x)] \\
&\approx \mathbb{E}_{p_d}[\nabla_\theta \log E_\theta(x)] - \mathbb{E}_z\left[\nabla_\theta E_\theta\left(x - \frac{\epsilon^2}{2}\nabla_x \log E_{\theta'}(x) + \epsilon z\right)\Big|_{\theta'=\theta}\right] \\
&= \frac{\epsilon^2}{2}\nabla_\theta D_F(p_d(x)||p_\theta(x)) + o(\epsilon^2).
\end{aligned}$$

The last equation holds after Taylor series expansion with respect to ϵ.

Score-Based Generative Models

When θ is optimal in SM, we have

$$p_d(x) \propto E_{\theta^*}(x);$$
$$\nabla_x \log p_d(x) = \nabla_x \log E_{\theta^*}(x).$$

One typical application of SM is creating new samples that are similar to training data, by using Langevin MCMC

$$x^{k+1} \leftarrow x^k + \frac{\epsilon^2}{2} \underbrace{\nabla_x \log p_d(x^k)}_{=\nabla_x \log E_{\theta^*}(x)} + \epsilon z^k, k = 0, 1, \ldots, K-1.$$

Noise Contrastive Estimation

NCE treats the normalized term Z_θ as a learnable paramenter and learns parameters by distinguishing the sample from empirical distribution p_d and noise distribution p_n.

- We first define a mixture distribution

$$p_{n,d} := p(y=0)p_n(x) + p(y=1)p_d(x).$$

- The posterior distribution is given by

$$p_{n,d}(y=0|x) = \frac{p_n(x)}{p_n(x) + vp_d(x)}.$$

- The posterior probability given the noise/model mixture is

$$p_{n,\theta}(y=0|x) = \frac{p_n(x)}{p_n(x) + vp_\theta(x)}.$$

Noise Contrastive Estimation

The training objective of NCE is

$$\begin{aligned}\theta^* &= \operatorname*{argmin}_{\theta} \mathbb{E}_{p_{n,d}(x)} \left[KL(p_{n,d}(y|x)||p_{n,\theta}(y|x)) \right] \\ &= \operatorname*{argmin}_{\theta} \mathbb{E}_{p_{n,d}(x,y)} \left[\log p_{n,\theta}(y|x) \right].\end{aligned}$$

When θ is optimal, we have

$$\begin{aligned} p_{n,\theta^*}(y=0|x) &\equiv p_{n,d}(y=0|x) \\ \Leftrightarrow \frac{p_n(x)}{p_n(x) + v p_{\theta^*}(x)} &\equiv \frac{p_n(x)}{p_n(x) + v p_d(x)} \\ \Leftrightarrow p_{\theta^*}(x) &\equiv p_d(x). \end{aligned}$$

Noise Contrastive Estimation & Score Matching

The flexibility of NCE allows adaptation to special properties with hand-tuned $p_n(x)$ and v.

- We define the noise distribution as $p_n(x) = p_d(x - v)$.
- The posterior distribution is

$$p_{n,\theta} := \frac{p_\theta(x - v)}{p_\theta(x) + p_\theta(x - v)}.$$

- In this case, the NCE objective reduces to

$$\begin{aligned}\theta^* &= \operatorname*{argmin}_\theta \mathbb{E}_{p_d}\left[\log(1 + E_\theta(x)/E_\theta(x - v)) + \log(1 + E_\theta(x)/E_\theta(x + v))\right] \\ &\approx \operatorname*{argmin}_\theta \frac{1}{4}\mathbb{E}_{p_d(x)p(v)}\left[\frac{1}{2}||\nabla_x \log p_\theta(x)||^2 + v^T \nabla_x^2 \log p_\theta(x) v\right] \\ &\qquad + 2\log 2 + o(||v||^2).\end{aligned}$$

Conclusion

- We reviewed some of the modern approaches for the training of unnormalized probabilistic models.
- We focused on maximum likelihood estimation with MCMC sampling (Contrastive Divergence), Score Matching and Noise Contrastive Estimation.
- We introduced the application of generative models, but did not cover another aspects, like latent variables models, some downstream applications.