Gradient Guided Ratio Matching

Zijing Ou (2021.10.10)

In July this year, I gave a presentation of the energy-based models in our reading groups (slides). In the talk, I proposed a simple method to enhance ratio matching [1] by introducing gradient relaxation [2]. The experimental results on learning Boltzmann Machine seem quite good compared with original ratio matching. Recently, when I skimmed the submission papers on ICLR-2022, I found a paper, named *GRADIENT-GUIDED IMPORTANCE SAMPLING FOR LEARNING DISCRETE ENERGY-BASED MODELS* [3], which applies a similar method to reduce the time and space complexity of the ratio matching. Overall, the idea is simple: instead of matching the ratio between data and all the flipped points, the authors reformulate the objective of ratio matching into the perspective of expectation and reduce its variance by using importance sampling. We briefly discuss this method as follows.

1 Ratio Matching

Hyvärinen [1] and Lyu [4] proposed to learn a discrete energy based model $p_{\boldsymbol{\theta}} = e^{-E_{\boldsymbol{\theta}}(\boldsymbol{x})}/Z$ with $\boldsymbol{x} \in \{0,1\}^d$ by matching the probabilistic ratio by minimizing the objective function

$$\mathcal{J}_{RM}(\boldsymbol{\theta}) = \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})} \sum_{i=1}^{d} \left[\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})} \right]^2 = \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})} \sum_{i=1}^{d} \left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x}) - E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})} \right]^2. \tag{1}$$

The intuition of this objective is pushing down the energy of the training sample $\boldsymbol{x}$ and pushing up the energies of other noisy data points obtained by flipping one dimension of $\boldsymbol{x}$. However, (1) suffers from time complexity with $\mathcal{O}(d)$, which is inefficient in high dimensional data. To address this issue, one can randomly sample several dimensions and reduce their ratios heuristically, which derives the expectation perspective of ratio matching

$$\mathcal{J}_{RM}(\boldsymbol{\theta}, \boldsymbol{x}) = d \sum_{i=1}^{d} \frac{1}{d} \left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x}) - E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})} \right]^2 = d\mathbb{E}_{m(\boldsymbol{x}_{-i})} \left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x}) - E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})} \right]^2, \tag{2}$$

where $m(\boldsymbol{x}_{-i}) = \frac{1}{d}$ for $i = 1, \ldots, d$ is a uniformed category distribution. So, one can estimate (2) via Monte Carlo sampling

$$\widehat{\mathcal{J}_{RM}} \approx d\frac{1}{s} \sum_{t=1}^{s} \left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x}) - E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i}^{(t)})} \right]^2, \quad \boldsymbol{x}_{-i}^{(t)} \sim m(\boldsymbol{x}_{-i}). \tag{3}$$

Such a simple trick can reduce the complexity from $\mathcal{O}(d)$ to $\mathcal{O}(s)$, which is exactly efficient in high dimensional data when $s \ll d$. However, it suffers from high variance due the uniform sampling process. Next, we introduce importance sampling to alleviate this problem.

2 Variance Reduction via Importance Sampling

Importance sampling is a widely used method to reduce variance in MC sampling. Instead of sampling based on the original distribution $m(\boldsymbol{x}_{-i})$, importance sampling proposes to sample from another proposal distribution $n(\boldsymbol{x}_{-i})$.

$$\mathcal{J}_{RM}(\boldsymbol{\theta}, \boldsymbol{x})_n = d\mathbb{E}_{m(\boldsymbol{x}_{-i})}\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2 = d\mathbb{E}_{n(\boldsymbol{x}_{-i})}\frac{m(\boldsymbol{x}_{-i})\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2}{n(\boldsymbol{x}_{-i})}. \tag{4}$$

Thereby, one can estimate (4) by

$$\widehat{\mathcal{J}_{RM}(\boldsymbol{\theta}, \boldsymbol{x})_n} \approx d\frac{1}{s}\sum_{t=1}^{s}\frac{m(\boldsymbol{x}_{-i}^{(t)})\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i}^{(t)})}\right]^2}{n(\boldsymbol{x}_{-i}^{(t)})}, \quad \boldsymbol{x}_{-i}^{(t)} \sim n(\boldsymbol{x}_{-i}). \tag{5}$$

By selecting a proper proposal distribution $n(\boldsymbol{x}_{-i})$, one can reduce the variance of (5) by a large amount. The optimal $n^*(\boldsymbol{x}_{-i})$, with zero variance (see appendix A), is given by

$$n^*(\boldsymbol{x}_{-i}) = \frac{m(\boldsymbol{x}_{-i})\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2}{\sum_{i=1}^{d} m(\boldsymbol{x}_{-i})\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2} = \frac{\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2}{\sum_{i=k}^{d}\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-k})}\right]^2}. \tag{6}$$

This proposal distribution is optimal, but not really usable in practice, because for each sample $\boldsymbol{x}_{-i}^* \sim n^*(\boldsymbol{x}_{-i})$, we have

$$\frac{m(\boldsymbol{x}_{-i}^*)\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i}^*)}\right]^2}{n(\boldsymbol{x}_{-i}^*)} = m(\boldsymbol{x}_{-i}^*)\sum_{i=1}^{d}\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2, \tag{7}$$

which is the sum over all flips with complexity of $\mathcal{O}(d)$. However, this problem can be sidestepped by the application of gradient relaxation trick proposed in [2]. Specifically, it can be shown that the computation complexity of $n^*(\boldsymbol{x}_{-i})$ can be reduced to $\mathcal{O}(1)$ via Taylor expansion.

3 Gradient Guided Importance Sampling

It is observed by Grathwohl et al. [2] that many discrete distributions are implemented as continuous differentiable functions, although they are evaluated only in discrete domains. Under this assumption, we can apply Taylor expansion on $E_{\boldsymbol{\theta}}(\boldsymbol{x})$

$$E_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i}) + (\boldsymbol{x} - \boldsymbol{x}_{-i})^T \nabla_{\boldsymbol{x}} E_{\boldsymbol{\theta}}(\boldsymbol{x}). \tag{8}$$

Note that we have $[\boldsymbol{x}]_i - [\boldsymbol{x}_{-i}]_i = -1$ if $[\boldsymbol{x}]_i = 0$ and $[\boldsymbol{x}]_i - [\boldsymbol{x}_{-i}]_i = 1$ if $[\boldsymbol{x}]_i = 1$, which can be unified as $[\boldsymbol{x}]_i - [\boldsymbol{x}_{-i}]_i = 2[\boldsymbol{x}]_i - 1$. Thus we have

$$E_{\boldsymbol{\theta}}(\boldsymbol{x}) - E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i}) = [(2\boldsymbol{x} - 1) \odot \nabla_{\boldsymbol{x}} E_{\boldsymbol{\theta}}(\boldsymbol{x})]_i, \quad i = 1, \ldots, d, \tag{9}$$

where $\odot$ denotes element-wise multiplication. Therefore, we can approximate $n^*(\boldsymbol{x}_{-i})$ with $\mathcal{O}(1)$ complexity

$$\tilde{n}^*(\boldsymbol{x}_{-i}) = \frac{\left[e^{2(2\boldsymbol{x}-1)\odot\nabla_{\boldsymbol{x}}E_{\boldsymbol{\theta}}(\boldsymbol{x})}\right]_i}{\sum_{k=1}^{d}\left[e^{2(2\boldsymbol{x}-1)\odot\nabla_{\boldsymbol{x}}E_{\boldsymbol{\theta}}(\boldsymbol{x})}\right]_k}. \tag{10}$$

Thus, we can train the discrete energy based model by minimizing the following objective

$$\widehat{\mathcal{J}_{RM}(\boldsymbol{\theta},\boldsymbol{x})_{\tilde{n}^*}} \approx d\frac{1}{s}\sum_{t=1}^{s}\frac{m(\boldsymbol{x}_{-i}^{(t)})\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i}^{(t)})}\right]^2}{\tilde{n}^*(\boldsymbol{x}_{-i}^{(t)})}, \quad \boldsymbol{x}_{-i}^{(t)} \sim \tilde{n}^*(\boldsymbol{x}_{-i}), \tag{11}$$

which complexity is $\mathcal{O}(s)$, and this is the final objective proposed in [3].

4 My Notes

The training process can be concluded as following steps: i) sampling the flipping index $n^*(\boldsymbol{x}_{-i}) \propto \left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2 \approx \left[e^{2(2\boldsymbol{x}-1)\odot\nabla_{\boldsymbol{x}}E_{\boldsymbol{\theta}}(\boldsymbol{x})}\right]_i$, which is with a higher probability of interest on the flipped dimension that enjoys lower energy; and ii) simultaneously pushing down the energy of the data point and pushing up the energy of flipped data by minimizing $\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i}^{(t)})}\right]^2$ *w.r.t.* $\boldsymbol{\theta}$. Overall, the intuition behind this method is quite simple, and similar to the adversarial training: instead of minimizing the energy of all flipped data points, we only need to find the worst case, which has the comparatively lower energy, and push up its energy.

Another quite interesting thing is that, generally, the optimal proposal distribution (6) is not usable in practice, because the partition function of $n^*(\boldsymbol{x}_{-i})$ is intractable, as shown in (7). So this zero-variance importance sampling densities just provides insight into the design of a good importance sampling scheme, but not usable in practice. However, with the guidance of gradient, the optimal proposal distribution is magically practical, since we can estimate the partition function efficiently, as shown in (10). I do believe this trick has a lot of potentials to be used in a wide range of applications.

Besides, I find that we can directly apply the same trick on (1), that is

$$\mathcal{J}_{RM}(\boldsymbol{\theta},\boldsymbol{x}) = \sum_{i=1}^{d}\left[e^{E_{\boldsymbol{\theta}}(\boldsymbol{x})-E_{\boldsymbol{\theta}}(\boldsymbol{x}_{-i})}\right]^2 \approx \sum_{i=1}^{d}\left[e^{2(2\boldsymbol{x}-1)\odot\nabla_{\boldsymbol{x}}E_{\boldsymbol{\theta}}(\boldsymbol{x})}\right]_i = \left\|e^{(2\boldsymbol{x}-1)\odot\nabla_{\boldsymbol{x}}E_{\boldsymbol{\theta}}(\boldsymbol{x})}\right\|_2^2. \tag{12}$$

Recall the objective of score matching

$$\mathcal{J}_{SM}(\boldsymbol{\theta},\boldsymbol{x}) = \frac{1}{2}\|\nabla_{\boldsymbol{x}}E_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2 + tr(\nabla_{\boldsymbol{x}}^2 E_{\boldsymbol{\theta}}(\boldsymbol{x})). \tag{13}$$

It can be seen that (12) is the score function scaled by exponential. There may exit some connections between these two objectives. Although (12) is heuristic, maybe it is useful in practice. I am very curious to know more techniques about a potential shortcut to train discrete energy based models, but it is rarely explored by recent literature. So I am quite excited to see that such a simple method proposed by Anonymous [3] works well in their experiments. I will try to reproduce their experimental results in my spare time[1].

[1]Code is available now: https://github.com/J-zin/RMwGGIS

5 Experimental Reproduction

We reproduce the experiment on synthetic discrete data. The experimental setup follows that in [3], and the results seem quite good.

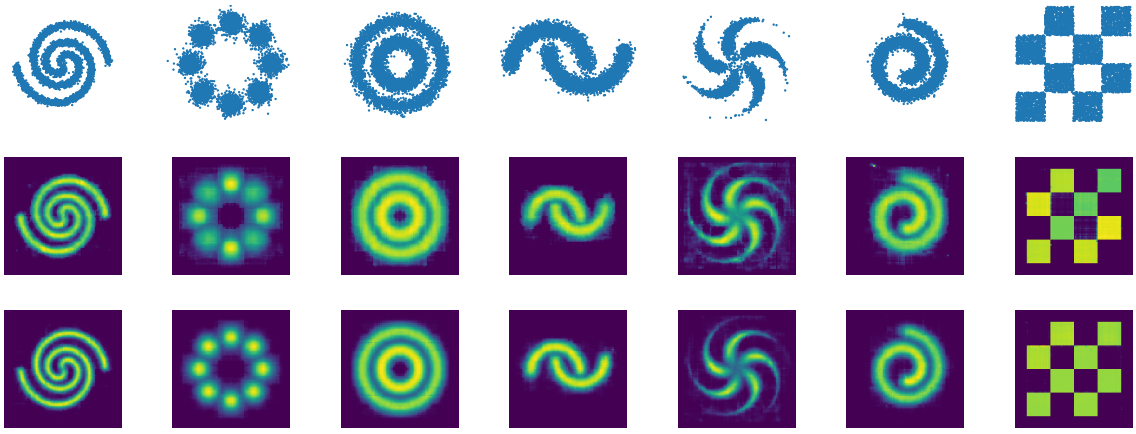

Figure 1: Visualization of learned energy functions on 32-dimensional synthetic discrete datasets. From the first row to the last: training data, RMwGGIS (biased), and RMwGGIS (unbiased).

References

[1] A. Hyvärinen, "Some extensions of score matching," *Computational statistics & data analysis*, vol. 51, no. 5, pp. 2499–2512, 2007.

[2] W. Grathwohl, K. Swersky, M. Hashemi, D. Duvenaud, and C. J. Maddison, "Oops i took a gradient: Scalable sampling for discrete distributions," *arXiv preprint arXiv:2102.04509*, 2021.

[3] Anonymous, "Gradient-guided importance sampling for learning discrete energy-based models," in *Submitted to The Tenth International Conference on Learning Representations*, 2022, under review. [Online]. Available: https://openreview.net/forum?id=IEKL-OihqX0

[4] S. Lyu, "Interpretation and generalization of score matching," *arXiv preprint arXiv:1205.2629*, 2012.

A The Optimal Proposal Distribution

The importance sampling estimate of $\mu = \mathbb{E}_p[f(\boldsymbol{x})]$ is

$$\hat{\mu}_q = \frac{1}{n}\sum_{i=1}^{n} \frac{f(\boldsymbol{x}_i)p(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i)}, \quad \boldsymbol{x}_i \sim q(\boldsymbol{x}).$$

Theorem 1. *When the proposal distribution $q(\boldsymbol{x})$ is given by*

$$q(\boldsymbol{x}) = \frac{p(\boldsymbol{x})f(\boldsymbol{x})}{\int p(\boldsymbol{x})f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}},$$

then we have $Var(\hat{\mu}_q) = 0$.

Proof.

$$\begin{aligned}
Var(\hat{\mu}_q) &= \frac{1}{n} Var\left(\frac{f(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})}\right) \\
&= \frac{1}{n}\left\{\mathbb{E}_{q(\boldsymbol{x})}\left[\frac{f(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]^2 - \left[\mathbb{E}_{q(\boldsymbol{x})}\frac{f(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]^2\right\} \\
&= \frac{1}{n}\left\{\int \frac{f^2(\boldsymbol{x})p^2(\boldsymbol{x})}{q(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} - \left[\mathbb{E}_{p(\boldsymbol{x})}f(\boldsymbol{x})\right]^2\right\} \\
&= \frac{1}{n}\left\{\left[\mathbb{E}_{p(\boldsymbol{x})}f(\boldsymbol{x})\right]^2 - \left[\mathbb{E}_{p(\boldsymbol{x})}f(\boldsymbol{x})\right]^2\right\} \\
&= 0.
\end{aligned}$$

This completes the proof of Theorem 1. □