This chapter focuses on how we solve the partition function which normalizes potential functions of undirected graph. We can see that solution for this kind can be divided into three trends: ① Approximation (Sampling, Monte Carlo Methods) on the gradient.

② Avoid calculating partition function using ratio.

③ Directly approximate partition function using sampling methods.

The log-likelihood Gradient and Stochastic Maximum Likelihood and Contrastive Divergence.

- When we calculating $\nabla_\theta \log P(x|\theta)$ (likelihood), we can write

$$\nabla_\theta \log P(x|\theta) = \nabla_\theta \log \tilde{P}(x|\theta) - \nabla_\theta \log Z(\theta)$$ ← Partition function

$$\nabla_\theta \log Z = \frac{1}{Z} \nabla_\theta Z = \frac{1}{Z} \nabla_\theta \sum_x \tilde{P}(x)$$

$$= \frac{1}{Z} \sum_x \nabla_\theta \tilde{P}(x)$$

Here we can get expectation form $\Rightarrow \sum_x P(x) \nabla_\theta \log \tilde{P}(x)$

- We can write the gradient of partition function to expectation form and then sampling approximation, when we using SGD to maximum likelihood.
- Three algorithms of sampling when calculating gradient.

  Sampling k samples from distribution ($P(x)$, $P(x|\theta)$, $P(x|\theta+\epsilon g)$) to calculate partition function.
- We can see stochastic maximum likelihood as optimizing θ with two forces:

  positive force: The likelihood of data / training data / observed data, which puts our model towards the distribution of training data.

  negative force: The belief shown by partition function, tends to normalize the distribution.
- All these methods based on using MCMC to draw samples from the model can in principle be used with almost any variant of MCMC.
- Here since we calculate Z independently, we can use lower bound methods on $\tilde{P}(x|\theta)$.

Pseudo likelihood

Consider this method of we don't calculate partition function, we approximate it by

$$\log P(x) = \log P(x_1) + \log P(x_2|x_1) + \cdots + \log P(x_n|x_{1:n-1})$$

Here each conditional probability can be computed by $\frac{\tilde{P}(x)}{\tilde{P}(y)}$, with canceling both partition function on top and down.

It can be used for tasks that require only the conditional distributions used during training, such as filling missed value, not good at estimating full joint distribution.

Score Matching and Ratio Matching

Here score matching means that we want $\nabla_x P_{data}$ as similar as $\nabla_x P_{model}$, namely P_{data} is as close as P_{model} under consideration of this metric. The idea behind this is that $\nabla_x Z = 0$

- score matching in practice: $\hat{L}(x,\theta) = \sum_j \left(\frac{\partial^2}{\partial x_j^2} \log P_{model}(x|\theta) + \frac{1}{2}\left(\frac{\partial}{\partial x_j} P_{model}(x|\theta)\right)^2\right)$
- ratio matching in practice: $L^{(RM)}(x|\theta) = \sum_j \left(1 + \frac{P_{model}(x|\theta)}{P_{model}(f(x,j)|\theta)}\right)^{-2}$

Denoising Score Matching

$$P_{smoothed}(x) = \int P_{data}(y) \cdot q_\sigma(x|y)\, dy$$

x is got by adding noise on y

Denoise Auto Encoder using Gaussian Noise and MSE to train can be seen as score matching, where the P_{model} is true distribution and P_{data} is noisy version.

Noise-Contrastive Estimation

The idea behind NCE is that we write $\log P_{model}(x) = \log \tilde{p}(x) + c$. we learn θ and c (representative to partition function) together by leaning a supervised, two-class problem, to judge if data is generated by P_{model} or not:

$P_{noise}(x)$ vs. $P_{model}(x)$ (Initial ideas of GAN)

One significant drawback of this is that we may face learning only sub-modes of the data due to diversity of data.

Estimating the Partition Function

Unlike methods mentioned before (calculating the gradient or avoid calculating partition), here we directly face estimating partition function.

- Why we need that: Comparing performance of model A and model B, because the partition function of these two models will never be the same, thus we cannot cancel them.

- Sampling methods based on importance sampling / metropolis-hasting sampling engage a distribution P_0 which we need for approximating P_1. Then how we build the gap between P_0 and P_1 is important:
  - Annealed Importance Sampling

    We construct a series of $P_{\eta_1} \cdots P_{\eta_n}$ from P_0 to P_1, and going from P_0 to P_1 by steping P_{η_1} to P_{η_n} can be seen as a markov chain.

    Samples from P_{η_1} to P_{η_n} corresponds to different states of P_0 to P_1.
  - Bridge Sampling

    Use a direct P^*, to bridge the gap between P_0 and P_1, this can be used with AIS as an internal step from $P_{\eta_{n-1}}$ to P_{η_n}. (Also can combine with tempering)