




# Advanced Deep Learning

## Confronting the Partition Function-2

**U Kang**  
**Seoul National University**



# Outline

- The Log-Likelihood Gradient
- Stochastic Maximum Likelihood and Contrastive Divergence
-   **Pseudolikelihood**
- Noise-Contrastive Estimation
- Estimating the Partition Function



# Pseudolikelihood

- Another approach to solve intractability of partition function is to train the model without computing the partition function
- Observation: conditional distribution can be computed without partition function

$$\frac{p(\mathbf{x})}{p(\mathbf{y})} = \frac{\frac{1}{Z}\tilde{p}(\mathbf{x})}{\frac{1}{Z}\tilde{p}(\mathbf{y})} = \frac{\tilde{p}(\mathbf{x})}{\tilde{p}(\mathbf{y})}$$



# Pseudolikelihood

## ■ Conditional probabilities

$$p(\mathbf{a} | \mathbf{b}) = \frac{p(\mathbf{a}, \mathbf{b})}{p(\mathbf{b})} = \frac{p(\mathbf{a}, \mathbf{b})}{\sum_{\mathbf{a}, \mathbf{c}} p(\mathbf{a}, \mathbf{b}, \mathbf{c})} = \frac{\tilde{p}(\mathbf{a}, \mathbf{b})}{\sum_{\mathbf{a}, \mathbf{c}} \tilde{p}(\mathbf{a}, \mathbf{b}, \mathbf{c})}$$

- If  $\mathbf{a}$  and  $\mathbf{c}$  are large, it may require large number of variables to be marginalized



# Main Idea of Pseudolikelihood

$$p(\mathbf{a} | \mathbf{b}) = \frac{p(\mathbf{a}, \mathbf{b})}{p(\mathbf{b})} = \frac{p(\mathbf{a}, \mathbf{b})}{\sum_{\mathbf{a}, \mathbf{c}} p(\mathbf{a}, \mathbf{b}, \mathbf{c})} = \frac{\tilde{p}(\mathbf{a}, \mathbf{b})}{\sum_{\mathbf{a}, \mathbf{c}} \tilde{p}(\mathbf{a}, \mathbf{b}, \mathbf{c})}$$

- If we have  $n$  variables, with the chain of probabilities:

$$\log p(\mathbf{x}) = \log p(x_1) + \log p(x_2 | x_1) + \cdots + \log p(x_n | \mathbf{x}_{1:n-1})$$

- We can make  $\mathbf{a}$  maximally small, but in the worst case, we must marginalize a set of variables of size  $n-1$
- What if we move  $\mathbf{c}$  into  $\mathbf{b}$  to ignore computational cost?  $\Rightarrow$  pseudolikelihood

$$\sum_{i=1}^n \log p(x_i | \mathbf{x}_{-i})$$



# Pseudolikelihood

## ■ Generalized Pseudolikelihood Estimator

- We trade computational complexity for deviation from maximum likelihood behavior
- Uses  $m$  sets of indices of variable that appear together on the left side of the conditioning bar

- Objective function: 
$$\sum_{i=1}^m \log p(\mathbf{x}_{\mathcal{S}(i)} \mid \mathbf{x}_{-\mathcal{S}(i)})$$



# Pseudolikelihood

- Performance of pseudolikelihood-based approaches
  - Depends largely on how the model is used
  - Performs poorly when a good model of the full joint  $p(x)$  is needed (e.g. sampling)
  - Performs well when only the conditional distributions used during training are needed (e.g. filling in small amounts of missing values)



# Pseudolikelihood

- Performance of pseudolikelihood-based approaches

- Generalized pseudolikelihood:

Powerful if the data has regular structure that allows the index sets to be designed to capture the most important correlations while leaving out negligible correlations (e.g. pixels in natural images: each GPL's set is a small, spatially localized window)





# Pseudolikelihood

- Pseudolikelihood estimator's weaknesses
  - Cannot be used with other approximations that only provide a lower bound on  $\tilde{p}(x)$ , such as variational inference
    - ⇒ Hard to apply on deep models (e.g. Deep Boltzmann machine) because variational methods are used to approximately marginalizing out the many layers of hidden variables that interact with each other
  - But, useful to train single-layer models or deep models with methods not based on lower bounds



# Outline

- The Log-Likelihood Gradient
- Stochastic Maximum Likelihood and Contrastive Divergence
- Pseudolikelihood
- Noise-Contrastive Estimation**
- Estimating the Partition Function



# Noise-Contrastive Estimation (NCE)

- Consider a language model which predicts a word  $w$  in a vocabulary  $V$  based on a context  $c$ 
  - $p_{\theta}(w|c) = \frac{u_{\theta}(w,c)}{\sum_{w' \in V} u_{\theta}(w',c)} = \frac{u_{\theta}(w,c)}{Z_{\theta}(c)}$
- The goal is to learn parameter  $\theta$  using maximum likelihood.
  - I.e., find parameters of the model  $p_{\theta}(w|c)$  that best approximates the empirical distribution  $\tilde{p}(w|c)$
- However, evaluating the partition function  $Z_{\theta}(c)$  is expensive
- How can we learn  $\theta$  without evaluating the partition function?

[Chris Dyer, Notes on Noise Contrastive Estimation and Negative Sampling]



# Noise-Contrastive Estimation

- NCE reduces the language model estimation problem to the problem of estimating the parameters of a probabilistic classifier that uses the same parameters to distinguish samples from the empirical distribution from samples generated by the noise distribution
  - Intuition: a good theta can perform well for classifying whether a word is from the noise distribution or empirical distribution:  $p(D=0 \text{ or } 1 \mid c, w)$
- The two-class training data is generated as follows: sample a  $c$  from  $\tilde{p}(c)$ , then sample one “true” sample from  $\tilde{p}(w|c)$ , with auxiliary label  $D=1$  indicating the datapoint is drawn from the true distribution, and  $k$  “noise” samples from a noise distribution  $q(w)$  with auxiliary label  $D=0$  indicating these data points are noise
- Thus, the joint probability of  $d, w$  given  $c$  is given by

$$\square \quad p(d, w|c) = \begin{cases} \frac{k}{1+k} q(w) & \text{if } d = 0 \\ \frac{1}{1+k} \tilde{p}(w|c) & \text{if } d = 1 \end{cases}$$



# Noise-Contrastive Estimation

- Then, a conditional probability of  $d$  having observed  $w$  and  $c$  is given by

- $$p(D = 0 | c, w) = \frac{\frac{k}{1+k}q(w)}{\frac{1}{1+k}\tilde{p}(w|c) + \frac{k}{1+k}q(w)} = \frac{k \times q(w)}{\tilde{p}(w|c) + k \times q(w)}$$

- $$p(D = 1 | c, w) = \frac{\tilde{p}(w|c)}{\tilde{p}(w|c) + k \times q(w)}$$

- NCE replaces the empirical distribution  $\tilde{p}(w|c)$  with the model distribution  $p_{\theta}(w|c)$

- NCE makes two further assumptions

- It proposes partition function value  $Z(c)$  be estimated as parameter  $z_c$
- Fixing  $z_c = 1$  for all  $c$  is effective

- With the two assumptions, then the above probabilities are

- $$p(D = 0 | c, w) = \frac{k \times q(w)}{u_{\theta}(w,c) + k \times q(w)}$$

- $$p(D = 1 | c, w) = \frac{u_{\theta}(w,c)}{u_{\theta}(w,c) + k \times q(w)}$$



# Noise-Contrastive Estimation

- We now have a binary classification problem with parameters  $\theta$  that can be trained to maximize conditional log-likelihood of  $D$ , with  $k$  negative samples chosen:
  - $L_{NCE_k} = \sum_{(w,c) \in D} (\log p(D = 1|c, w) + k E_{\bar{w} \sim q} \log p(D = 0|c, \bar{w}))$
- The above  $L_{NCE_k}$  can be approximated with its Monte Carlo approximation
  - $L_{NCE_k}^{MC} = \sum_{(w,c) \in D} (\log p(D = 1|c, w) + k \sum_{i=1}^k \frac{1}{k} \log p(D = 0|c, \bar{w}_i)) = \sum_{(w,c) \in D} (\log p(D = 1|c, w) + \sum_{i=1}^k \log p(D = 0|c, \bar{w}_i))$



# Negative Sampling


- Negative sampling is a special case of Noise-Contrastive Estimation (NCE) where  $k=|V|$  and  $q$  is uniform. Thus,

$$\square p(D = 0 | c, w) = \frac{k \times q(w)}{u_{\theta}(w, c) + k \times q(w)} = \frac{1}{u_{\theta}(w, c) + 1}$$

$$\square p(D = 1 | c, w) = \frac{u_{\theta}(w, c)}{u_{\theta}(w, c) + k \times q(w)} = \frac{u_{\theta}(w, c)}{u_{\theta}(w, c) + 1}$$



# Outline

- The Log-Likelihood Gradient
- Stochastic Maximum Likelihood and Contrastive Divergence
- Pseudolikelihood
- Noise-Contrastive Estimation
-   **Estimating the Partition Function**





# Estimating Partition Function

- Reminder of the comparing two models
  - Assume we have two models:
    - $M_A$  with a probability distribution  $p_A(\mathbf{x}; \boldsymbol{\theta}_A) = \frac{1}{Z_A} \tilde{p}_A(\mathbf{x}; \boldsymbol{\theta}_A)$
    - $M_B$  with a probability distribution  $p_B(\mathbf{x}; \boldsymbol{\theta}_B) = \frac{1}{Z_B} \tilde{p}_B(\mathbf{x}; \boldsymbol{\theta}_B)$
  - If  $\sum_i \log p_A(x^{(i)}; \boldsymbol{\theta}_A) - \sum_i \log p_B(x^{(i)}; \boldsymbol{\theta}_B) > 0$ , we say that  $M_A$  is a better model than  $M_B$ .
  - It seems that we need to know partition functions to evaluate the above equation
  - Since  $\sum_i \log p_A(x^{(i)}; \boldsymbol{\theta}_A) - \sum_i \log p_B(x^{(i)}; \boldsymbol{\theta}_B) = \sum_i \left( \log \frac{\tilde{p}_A(x^{(i)}, \boldsymbol{\theta}_A)}{\tilde{p}_B(x^{(i)}, \boldsymbol{\theta}_B)} \right) - m \log \frac{Z(\boldsymbol{\theta}_A)}{Z(\boldsymbol{\theta}_B)}$ , we don't need to know the partition functions but only their ratio.



# Estimating Partition Function

- If we know the ratio of two partition functions  $r = \frac{Z(\boldsymbol{\theta}_B)}{Z(\boldsymbol{\theta}_A)}$  and the actual value of one of the two functions  $Z(\boldsymbol{\theta}_A)$ , we could compute the value of the other:
  - $Z(\boldsymbol{\theta}_B) = \frac{Z(\boldsymbol{\theta}_B)}{Z(\boldsymbol{\theta}_A)} \cdot Z(\boldsymbol{\theta}_A) = rZ(\boldsymbol{\theta}_A)$



# Monte Carlo Method

- MC provides an approximation of the partition function.
- Example
  - Consider a desired partition function  $Z_1 = \int \tilde{p}_1(\mathbf{x}) d\mathbf{x}$  with a proposal distribution  $p_0(\mathbf{x}) = \frac{1}{Z_0} \tilde{p}_0(\mathbf{x})$  which supports tractable sampling and tractable evaluation of both  $Z_0$  and  $\tilde{p}_0(\mathbf{x})$ .
  - Then,  $Z_1 = \int \tilde{p}_1(\mathbf{x}) d\mathbf{x} = \int \frac{p_0(\mathbf{x})}{p_0(\mathbf{x})} \tilde{p}_1(\mathbf{x}) d\mathbf{x}$ 
$$= Z_0 \int p_0(\mathbf{x}) \frac{\tilde{p}_1(\mathbf{x})}{\tilde{p}_0(\mathbf{x})} d\mathbf{x}$$
  - $\hat{Z}_1 = \frac{Z_0}{K} \sum_{k=1}^K \frac{\tilde{p}_1(\mathbf{x}^{(k)})}{\tilde{p}_0(\mathbf{x}^{(k)})}$  such that  $\mathbf{x}^{(k)} \sim p_0$
  - This also allows to estimate the ratio  $\frac{\hat{Z}_1}{Z_0} = \frac{1}{K} \sum_{k=1}^K \frac{\tilde{p}_1(\mathbf{x}^{(k)})}{\tilde{p}_0(\mathbf{x}^{(k)})}$



# Monte Carlo Method

- Now, we have an approximation  $\hat{Z}_1$  of the desired partition function.
- However, this method is not practical due to difficulty of finding a tractable  $p_0$  s.t.
  - It is simple enough to evaluate
  - Close to  $p_1$  to result in a high quality approximation
    - If it is not, then most samples from  $p_0$  will have low probability under  $p_1$  and therefore make negligible contribution to the sum



# Monte Carlo Method

- Problem: the proposal  $p_0$  is too far from  $p_1$
- Two strategies to solve the problem
  - Annealed importance sampling, and bridge sampling
  - Idea: introduce intermediate distributions that attempt to bridge the gap between  $p_0$  and  $p_1$



# Annealed Importance Sampling

- Annealed Importance Sampling (AIS) overcomes this problem by using intermediate distributions.
- We have a sequence of distributions  $p_0, p_1, \dots, p_n$  with a sequence of partition functions  $Z_0, Z_1, \dots, Z_n$ .

■ Then,

$$\begin{aligned} \frac{Z_n}{Z_0} &= \frac{Z_n}{Z_0} \cdot \frac{Z_1}{Z_1} \cdots \frac{Z_{n-1}}{Z_{n-1}} \\ &= \frac{Z_1}{Z_0} \cdot \frac{Z_2}{Z_1} \cdots \frac{Z_n}{Z_{n-1}} = \prod_{j=0}^{n-1} \frac{Z_{j+1}}{Z_j} \end{aligned}$$



# Annealed Importance Sampling

- How can we obtain the intermediate distributions?
  - One popular choice is to use the weighted geometric average of target distribution  $p_1$  and proposal distribution  $p_0$ 
    - $p_{\eta_j} \propto p_1^{\eta_j} p_0^{1-\eta_j}$



# Bridge Sampling

- Bridge sampling is another method similar to AIS.
- The only difference between the two methods is that bridge sampling uses only one distribution  $p_*$ , known as the bridge, while AIS uses a series of distributions.





# What you need to know

- The Log-Likelihood Gradient
- Stochastic Maximum Likelihood and Contrastive Divergence
- Pseudolikelihood
- Noise-Contrastive Estimation
- Estimating the Partition Function



# Questions?