

Noise contrastive estimation is based on the idea that a good generative model should be able to distinguish data from noise. A closely related idea is that a good generative model should be able to generate samples that no classifier can distinguish from data. This idea yields generative adversarial networks (section 20.10.4).

18.7 Estimating the Partition Function

While much of this chapter is dedicated to describing methods that avoid needing to compute the intractable partition function $Z(\boldsymbol{\theta})$ associated with an undirected graphical model, in this section we discuss several methods for directly estimating the partition function.

Estimating the partition function can be important because we require it if we wish to compute the normalized likelihood of data. This is often important in *evaluating* the model, monitoring training performance, and comparing models to each other.

For example, imagine we have two models: model \mathcal{M}_A defining a probability distribution $p_A(\mathbf{x}; \boldsymbol{\theta}_A) = \frac{1}{Z_A} \tilde{p}_A(\mathbf{x}; \boldsymbol{\theta}_A)$ and model \mathcal{M}_B defining a probability distribution $p_B(\mathbf{x}; \boldsymbol{\theta}_B) = \frac{1}{Z_B} \tilde{p}_B(\mathbf{x}; \boldsymbol{\theta}_B)$. A common way to compare the models is to evaluate and compare the likelihood that both models assign to an i.i.d. test dataset. Suppose the test set consists of m examples $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$. If $\prod_i p_A(\mathbf{x}^{(i)}; \boldsymbol{\theta}_A) > \prod_i p_B(\mathbf{x}^{(i)}; \boldsymbol{\theta}_B)$ or equivalently if

$$\sum_i \log p_A(\mathbf{x}^{(i)}; \boldsymbol{\theta}_A) - \sum_i \log p_B(\mathbf{x}^{(i)}; \boldsymbol{\theta}_B) > 0, \quad (18.38)$$

then we say that \mathcal{M}_A is a better model than \mathcal{M}_B (or, at least, it is a better model of the test set), in the sense that it has a better test log-likelihood. Unfortunately, testing whether this condition holds requires knowledge of the partition function. Unfortunately, equation 18.38 seems to require evaluating the log probability that the model assigns to each point, which in turn requires evaluating the partition function. We can simplify the situation slightly by re-arranging equation 18.38 into a form where we need to know only the **ratio** of the two model's partition functions:

$$\sum_i \log p_A(\mathbf{x}^{(i)}; \boldsymbol{\theta}_A) - \sum_i \log p_B(\mathbf{x}^{(i)}; \boldsymbol{\theta}_B) = \sum_i \left(\log \frac{\tilde{p}_A(\mathbf{x}^{(i)}; \boldsymbol{\theta}_A)}{\tilde{p}_B(\mathbf{x}^{(i)}; \boldsymbol{\theta}_B)} \right) - m \log \frac{Z(\boldsymbol{\theta}_A)}{Z(\boldsymbol{\theta}_B)}. \quad (18.39)$$