

$$= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \nabla_{\boldsymbol{\theta}} \log \tilde{p}(\mathbf{x}). \quad (18.13)$$

This derivation made use of summation over discrete \mathbf{x} , but a similar result applies using integration over continuous \mathbf{x} . In the continuous version of the derivation, we use Leibniz's rule for differentiation under the integral sign to obtain the identity

$$\nabla_{\boldsymbol{\theta}} \int \tilde{p}(\mathbf{x}) d\mathbf{x} = \int \nabla_{\boldsymbol{\theta}} \tilde{p}(\mathbf{x}) d\mathbf{x}. \quad (18.14)$$

This identity is applicable only under certain regularity conditions on \tilde{p} and $\nabla_{\boldsymbol{\theta}} \tilde{p}(\mathbf{x})$. In measure theoretic terms, the conditions are: (i) The unnormalized distribution \tilde{p} must be a Lebesgue-integrable function of \mathbf{x} for every value of $\boldsymbol{\theta}$; (ii) The gradient $\nabla_{\boldsymbol{\theta}} \tilde{p}(\mathbf{x})$ must exist for all $\boldsymbol{\theta}$ and almost all \mathbf{x} ; (iii) There must exist an integrable function $R(\mathbf{x})$ that bounds $\nabla_{\boldsymbol{\theta}} \tilde{p}(\mathbf{x})$ in the sense that $\max_i |\frac{\partial}{\partial \theta_i} \tilde{p}(\mathbf{x})| \leq R(\mathbf{x})$ for all $\boldsymbol{\theta}$ and almost all \mathbf{x} . Fortunately, most machine learning models of interest have these properties.

This identity

$$\nabla_{\boldsymbol{\theta}} \log Z = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \nabla_{\boldsymbol{\theta}} \log \tilde{p}(\mathbf{x}) \quad (18.15)$$

is the basis for a variety of Monte Carlo methods for approximately maximizing the likelihood of models with intractable partition functions.

The Monte Carlo approach to learning undirected models provides an intuitive framework in which we can think of both the positive phase and the negative phase. In the positive phase, we increase $\log \tilde{p}(\mathbf{x})$ for \mathbf{x} drawn from the data. In the negative phase, we decrease the partition function by decreasing $\log \tilde{p}(\mathbf{x})$ drawn from the model distribution.

In the deep learning literature, it is common to parametrize $\log \tilde{p}$ in terms of an energy function (equation 16.7). In this case, we can interpret the positive phase as pushing down on the energy of training examples and the negative phase as pushing up on the energy of samples drawn from the model, as illustrated in figure 18.1.

18.2 Stochastic Maximum Likelihood and Contrastive Divergence

The naive way of implementing equation 18.15 is to compute it by burning in a set of Markov chains from a random initialization every time the gradient is needed. When learning is performed using stochastic gradient descent, this means the chains must be burned in once per gradient step. This approach leads to the