CSE6243: Advanced Machine Learning **Fall 2024**

Lecture 10: Energy Based Models

Lecturer: Bo Dai *Scribes: Jonathan Y. Zhou, James Zou*

Note: *LaTeX template courtesy of UC Berkeley EECS Department.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

10.1 Review

- A neural network is simply a function approximator, separate from the loss function you aim to optimize (parameterization of the model).
- Introduce the history of neural networks: Linear Model → Single-layer MLP (Kernel methods) → Multi-layer perceptron → Various specialized neural networks, including:
  - Convolutional Neural Networks (CNNs)
  - Residual Networks (ResNets)
  - Recurrent Neural Networks (RNNs)
  - Transformers
- Each of the above models is motivated by practical problems and tailored for different types of structured data.

10.2 Energy Based Model (EBM)

Although EBMs are not currently popular in the machine learning community, this model serves as the foundation for various generative models that we shall discuss in the future, such as GANs, diffusion models, and normalizing flows.

10.2.1 What is EBM?

An EBM is a model of the form

$$\mathbb{P}[\mathbf{x}] = \frac{\exp(f_{\boldsymbol{\Theta}}(\mathbf{x}))}{Z(\boldsymbol{\Theta})} \tag{EBM}$$

where the probability of the vector $\mathbf{x}$ is given in terms of a *energy (potential) function* $f_{\boldsymbol{\Theta}}(\mathbf{x})$ parameterized by $\boldsymbol{\Theta}$. In order the to make the probability measure integrate to one, we normalize $\mathbb{P}$ by the *partition function*

$$Z(\boldsymbol{\Theta}) = \int \exp(f_{\boldsymbol{\Theta}}(\mathbf{x}))\mathrm{d}\mathbf{x}. \tag{PF}$$

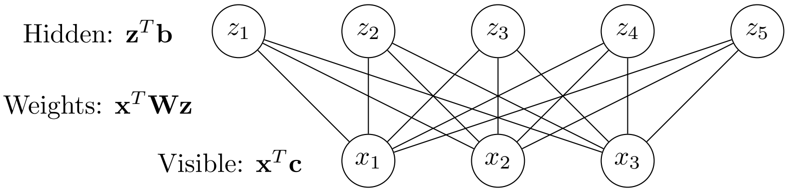
Hidden: $\mathbf{z}^T\mathbf{b}$
Weights: $\mathbf{x}^T\mathbf{W}\mathbf{z}$
Visible: $\mathbf{x}^T\mathbf{c}$
z_1
z_2
z_3
z_4
z_5
x_1
x_2
x_3

Figure 10.1: Restricted Boltzmann Machine (RBM)

An EBM may also be defined for a conditional distribution

$$\mathbb{P}[\mathbf{y} \mid \mathbf{x}] = \frac{\exp(f_{\boldsymbol{\Theta}}(\mathbf{x},\mathbf{y}))}{Z(\mathbf{x},\boldsymbol{\Theta})} \tag{CEBM}$$

$$Z(\mathbf{x},\boldsymbol{\Theta}) = \int \exp(f_{\boldsymbol{\Theta}}(\mathbf{x},\mathbf{y}))\mathrm{d}\mathbf{y} \tag{CPF}$$

We note (EBM) is similar to the exponential family

$$\mathbb{P}[\mathbf{x}] = \frac{\exp(\boldsymbol{\eta}^T T(\mathbf{x}))}{Z(\boldsymbol{\eta})} \tag{EF}$$

except that there are no sufficient statistics $T(\cdot)$, and you replace $\boldsymbol{\eta}^T T(\mathbf{x})$ with $f_{\boldsymbol{\Theta}}(\mathbf{x})$.

10.2.2 Examples of Energy Based Models

10.2.2.1 Markov Random Field (MRF)

Consider a collection of variables $[\mathbf{x}_i]_{i=1}^n$, and the following collection of factors

$$\mathbb{P}[\mathbf{x}] \propto \exp(f_{\boldsymbol{\Theta}}(\mathbf{x})) := \exp\left(\log\left(\prod_{i=1}^d \Psi_i(\mathbf{x}_i)\prod_{i\neq j}\Phi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\right)\right) \tag{MRF}$$

which is same as to say

$$f_{\boldsymbol{\Theta}}(\mathbf{x}) := \sum_{i=1}^d \log\Psi_i(\mathbf{x}_i) + \sum_{i\neq j}\log\Psi_{i,j}(\mathbf{x}_i,\mathbf{x}_j) \tag{10.1}$$

10.2.2.2 Restricted Boltzmann Machine (RBM)

Consider the bi-partite graphical model given in Figure 10.1. where $\mathbf{x}$ are the focal variables and $\mathbf{z}$ is latent. The two are related by the joint distribution

$$f_{\boldsymbol{\Theta}}(\mathbf{x},\mathbf{z}) = \mathbf{x}^T\mathbf{W}\mathbf{z} + \mathbf{z}^T\mathbf{b} + \mathbf{x}^T\mathbf{c} \tag{RBM}$$

which may be marginalized as

$$\mathbb{P}[\mathbf{x}] = \frac{1}{Z}\sum_{\mathbf{z}}\exp f_{\boldsymbol{\Theta}}(\mathbf{x},\mathbf{z}) \tag{10.2}$$

Both of these models are not used widely, but RBM can be used to build a connection to standard regression model. Indeed if

$$\mathbb{P}[\mathbf{y} \mid \mathbf{x}] \propto \exp(f_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{y})) \tag{10.3}$$

$$f_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{y}) = (\mathbf{W}\mathbf{y})^T \Psi(\mathbf{x}) \tag{10.4}$$

Then with appropriate Ψ you have different kinds of regression (linear, logistic $\mathbf{y} \in \{0, 1\}$, softmax $y \in [k]$).

10.2.2.3 Conditional Random Field (CRF)

RBM and MRF were for were for $\mathbb{P}[\mathbf{x}]$. However, we can also construct similar model for conditional case $\mathbb{P}[\mathbf{y} \mid \mathbf{x}]$ for example the following collection of potential functions

$$f_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{y}) = \sum_{i,j} \log \Psi_{i,j}(\mathbf{x}_i, \mathbf{y}_j) \tag{CRF}$$

Then you have

$$\mathbb{P}[\mathbf{y} \mid \mathbf{x}] = \frac{\exp(f_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{y}))}{Z(\mathbf{x}, \boldsymbol{\Theta})} = \frac{\exp(\sum_{i,j} \log \Psi_{i,j}(\mathbf{x}_i, \mathbf{y}_j))}{Z(\mathbf{x}, \boldsymbol{\Theta})} = \frac{\prod_{i,j} \Psi_{i,j}(\mathbf{x}_i, \mathbf{y}_j)}{Z(\mathbf{x}, \boldsymbol{\Theta})} \tag{10.5}$$

$$Z(\mathbf{x}, \boldsymbol{\Theta}) = \int \exp(f_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{y}))\mathrm{d}\mathbf{y} = \int \prod_{i,j} \Psi_{i,j}(\mathbf{x}_i, \mathbf{y}_j)\mathrm{d}\mathbf{y} \tag{10.6}$$

where your partition function is similar to (MRF), but the partition function is only partially marginalized (equal to one conditioned on one object).

From a historical perspective, these were popular around the start of the century, but these forms still manifest themselves implicitly today.

10.2.3 Motivation of EBM

We ought to discuss the origin of (EBM), which follows from *principle of maximum entropy.*

Consider the following variational problem across a space of distributions Δ

$$\begin{aligned} \max_{\mathbb{P} \in \Delta} H(\mathbb{P}) &:= -\int \mathbb{P}[\mathbf{x}] \log \mathbb{P}[\mathbf{x}] \mathrm{d}\mathbf{x} \\ \text{s.t. } & \mathbb{E}_{\mathbb{P}[\mathbf{x}]}[f(\mathbf{x})] = \boldsymbol{\mu} \end{aligned} \tag{MaxEnt}$$

where is f is some vector of statistics on $\mathbf{x}$. The principle of maximum entropy is to choose, from among the distributions $\mathbb{P}$ consistent with the data, the distribution $\mathbb{P}^*$ whose Shannon entropy is maximal.

We may solve (MaxEnt) by the calculus of variations, associate it with the functional

$$\mathcal{L}(\mathbb{P}, \alpha, \boldsymbol{\eta}) = \int \mathbb{P}[\mathbf{x}] \log \mathbb{P}[\mathbf{x}] \mathrm{d}\mathbf{x} - \alpha \left(\left(\int \mathbb{P}[\mathbf{x}] \mathrm{d}\mathbf{x} \right) - 1 \right) - \left(\langle \boldsymbol{\eta}, \int f(\mathbf{x}) \mathbb{P}[\mathbf{x}] \mathrm{d}\mathbf{x} \rangle - \boldsymbol{\mu} \right) \tag{10.7}$$

where we have Lagrange multipliers $\alpha, \boldsymbol{\eta}$. The entropy shall attain its maximum when the functional derivative is equal to zero.

$$\frac{\delta \mathcal{L}}{\delta \mathbb{P}}[\mathbb{P}] = \log \mathbb{P}[\mathbf{x}] + 1 - \alpha - \langle \boldsymbol{\eta}, f(\mathbf{x}) \rangle = 0 \tag{10.8}$$

It remains to solve for $\mathbb{P}$,

$$\begin{aligned} 0 &= \log \mathbb{P}[\mathbf{x}] + 1 - \alpha - \langle \boldsymbol{\eta}, f(\mathbf{x}) \rangle & (10.9) \\ \implies \log \mathbb{P}[\mathbf{x}] &= \langle \boldsymbol{\eta}, f(\mathbf{x}) \rangle + \alpha - 1 & (10.10) \\ \implies \mathbb{P}[\mathbf{x}] &= \exp(\langle \boldsymbol{\eta}, f(\mathbf{x}) \rangle + \alpha - 1) & (10.11) \\ \implies \mathbb{P}[\mathbf{x}] &= \exp(\langle \boldsymbol{\eta}, f(\mathbf{x}) \rangle)/Z & (10.12) \\ \implies \mathbb{P}[\mathbf{x}) &\propto \exp(\langle \boldsymbol{\eta}, f(\mathbf{x}) \rangle). & (10.13) \end{aligned}$$

To prove that this is indeed the maximum, you may take the second variation. Note that this is exactly the form of the exponential family, and the underlying motivation for why EBM takes a similar form.

10.2.4 Pros and Cons of EBM

Pros:

Easy to Analyze Relationships between Different Coordinates: If you have $f_{\boldsymbol{\Theta}}(\mathbf{x})$, you can analyze the relationships between different coordinates of the data. This approach is already widely used in *representation learning.* For instance, consider an EBM of form

$$\mathbb{P}[\mathbf{x}, \mathbf{y}] \propto \exp(\Psi(\mathbf{x})^T \phi(\mathbf{y})) \tag{10.14}$$

here $\mathbf{x}$ and $\mathbf{y}$ might represent images or text, and you want to select the $\mathbf{y}$ that when embedded maximally aligns the embeddings $\mathbf{x}$. By using the potential functions in (10.14), it becomes easy to extract relationships. This is what models like CLIP and SimCLR do (even though they do not explicitly state they are EBMs).

Model Compositionality You can easily combine different EBMs. For example, given two models

$$\begin{aligned} \mathbb{P}_1[\mathbf{x}] &\propto \exp f_{\boldsymbol{\Theta}_1}(\mathbf{x}) & (10.15) \\ \mathbb{P}_2[\mathbf{x}] &\propto \exp f_{\boldsymbol{\Theta}_2}(\mathbf{x}) & (10.16) \end{aligned}$$

you can combine them as

$$\mathbb{P}[\mathbf{x}] \propto \mathbb{P}_1[\mathbf{x}]\mathbb{P}_2[\mathbf{x}] = \exp(f_{\boldsymbol{\Theta}_1}(\mathbf{x}) + f_{\boldsymbol{\Theta}_2}(\mathbf{x})), \tag{10.17}$$

allowing you to capture the combined effect of both models by multiplication.

Cons:

Hard to Sample: When $\mathbf{x}$ is high-dimensional and $f(\mathbf{x})$ is complex, sampling the random variable X becomes computationally expensive (have to resort to Markov Chain Monte Carlo (MCMC), see Section 10.2.5).

Hard to Evaluate: To compute $\mathbb{P}[\mathbf{x}] = \exp(f_{\boldsymbol{\Theta}}(\mathbf{x}))/Z(\boldsymbol{\Theta})$, even if $f_{\boldsymbol{\Theta}}(\mathbf{x})$ is easy to evaluate, calculating the partition function $Z(\boldsymbol{\Theta})$ requires solving a high-dimensional integral, which is difficult.

Hard to Learn: The MLE for an EBM is given by as:

$$\max_{\boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^{n} f_{\boldsymbol{\Theta}}(\mathbf{x}_i) - \log Z(\boldsymbol{\Theta}) \tag{10.18}$$

Solving this exactly requires evaluating the partition function $Z(\boldsymbol{\Theta})$ for each $\boldsymbol{\Theta}$, which is computationally intractable. Indeed, a number of NP-Hard decision problems can be reduced to optimal EBM parameter learning.

10.2.5 Sampling from EBM

Recall the MCMC procedure, which allows us to sample from $\mathbb{P}$ as follows without computing the partition function:

1. $x \sim \mathbb{P}_0(\mathbf{x})$ $\mathbb{P}_0$ is the starting distribution

2. for $t := 1 \dots \infty$

   - $\mathbf{x}_{t+1} \sim T(\cdot \mid \mathbf{x}_t)$ Transition Operator

     Metropolis-Hastings Scheme:

     (a) $\mathbf{y} \sim \mathbb{Q}[\cdot \mid \mathbf{x}_t]$ $\mathbb{Q}$ is some proposal distribution (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$), Straightforward

     (b) $\mathbf{u} \sim \mathcal{U}[0,1]$ Straightforward

     (c) $A(\mathbf{x}_t, \mathbf{y}) := \min\left\{1, \frac{\mathbb{P}[\mathbf{x}_t]\mathbb{Q}[\mathbf{x}_t|\mathbf{y}]}{\mathbb{P}[\mathbf{x}_t]\mathbb{Q}[\mathbf{y}|\mathbf{x}_t]}\right\}$ Calculate Acceptance Probability, Straightforward

       – Note that $A(\mathbf{x}_t, \mathbf{y}) := \min\left\{1, \frac{\mathbb{P}[\mathbf{x}_t]\mathbb{Q}[\mathbf{x}_t|\mathbf{y}]}{\mathbb{P}[\mathbf{x}_t]\mathbb{Q}[\mathbf{y}|\mathbf{x}_t]}\right\} = \min\left\{1, \frac{\exp(f_{\mathbf{\Theta}}(\mathbf{y}))\mathbb{Q}[\mathbf{x}_t|\mathbf{y}]}{\exp(f_{\mathbf{\Theta}}(\mathbf{x}_t))\mathbb{Q}[\mathbf{y}|\mathbf{x}_t]}\right\}$

       – So you *do not need to compute the partition function* in (EBM).

       – If you choose a nice $\mathbb{Q}$ is it also easy to construct marginals.

     (d) $\mathbf{x}_{t+1} := \begin{cases} \mathbf{y} & u \leq A(\mathbf{x}_t, \mathbf{y}) \\ \mathbf{x}_t & u > A(\mathbf{x}_t, \mathbf{y}) \end{cases}$

     Langevin Scheme: Consider the overdamped Langevin Itô diffusion

     $$\dot{x_{\mathbf{t}}} = \mathbf{x}_t + \nabla \log \mathbb{P}[\mathbf{x}] + \sqrt{2}\dot{\boldsymbol{\epsilon}}$$

     for which we approximate via the Euler–Maruyama method

     (a) $\mathbf{y} = \mathbf{x}_t + \tau \nabla \log \mathbb{P}[\mathbf{x}_t] + \boldsymbol{\epsilon}$

     (b) Apply Metropolis-Hastings to accept or reject the proposal

     Note that although this requires a gradient evaluation, the gradient is with respect to $\mathbf{x}$ and so $\mathbf{\Theta}$ does not play a role.

10.2.6 EBM Learning with MCMC

Problem: (MLE) is very difficult to compute, since the computation of the $\log Z(\mathbf{\Theta})$ terms requires a high dimensional integral

$$\max_{\mathbf{\Theta}} \hat{\ell}(\mathbf{\Theta}) := \frac{1}{n}\sum_{i=1}^{n} f_{\mathbf{\Theta}}(\mathbf{x}_i) - \log Z(\mathbf{\Theta}) \tag{MLE}$$

but to perform first order optimization of (MLE), we require only the gradient

$$\frac{\partial \hat{\ell}(\mathbf{\Theta})}{\partial \mathbf{\Theta}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \mathbf{\Theta}} f_{\mathbf{\Theta}}(\mathbf{x}_i) - \frac{\partial}{\partial \mathbf{\Theta}} \log Z(\mathbf{\Theta}) \tag{10.19}$$

the terms $\frac{\partial}{\partial \boldsymbol{\Theta}} f_{\boldsymbol{\Theta}}(\mathbf{x}_i)$ are readily determined from back-propagation of $f_{\boldsymbol{\Theta}}$. The trouble is in approximating the second gradient term $\frac{\partial}{\partial \boldsymbol{\Theta}} \log Z(\boldsymbol{\Theta})$, which we may expand as follows

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\Theta}} \log Z(\boldsymbol{\Theta}) &= \frac{\partial}{\partial \boldsymbol{\Theta}} \log \int \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} && \text{Expansion of (PF)} \quad (10.20)\\
&= \left(\int \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} \right)^{-1} \frac{\partial}{\partial \boldsymbol{\Theta}} \int \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} && \text{Chain Rule} \quad (10.21)\\
&= \left(\int \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} \right)^{-1} \int \frac{\partial}{\partial \boldsymbol{\Theta}} \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} && \text{Leibniz Integral Rule} \quad (10.22)\\
&= \left(\int \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} \right)^{-1} \int \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \frac{\partial}{\partial \boldsymbol{\Theta}} f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} && \text{Chain Rule} \quad (10.23)\\
&= \int \left(\int \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} \right)^{-1} \exp f_{\boldsymbol{\Theta}}(\mathbf{x}) \frac{\partial}{\partial \boldsymbol{\Theta}} f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} && \text{Linearity} \quad (10.24)\\
&= \int \mathbb{P}[\mathbf{x}] \frac{\partial}{\partial \boldsymbol{\Theta}} f_{\boldsymbol{\Theta}}(\mathbf{x}) \mathrm{d}\mathbf{x} && \text{Definition of (EBM)} \quad (10.25)\\
&= \mathbb{E}_{\mathbb{P}[\mathbf{x}]} \left[\frac{\partial}{\partial \boldsymbol{\Theta}} f_{\boldsymbol{\Theta}}(\mathbf{x}) \right] && \text{Definition of Expectation} \quad (10.26)
\end{aligned}$$

Which is still a difficult to compute integral. However, this integral given in (10.26) admits readily a sample average approximation

$$\mathbb{E}_{\mathbb{P}[\mathbf{x}]} \left[\frac{\partial}{\partial \boldsymbol{\Theta}} f_{\boldsymbol{\Theta}}(\mathbf{x}) \right] \approx \frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \boldsymbol{\Theta}} f(\mathbf{x}_i), \quad \mathbf{x}_i \sim \mathbb{P}. \tag{10.27}$$

having terms which are readily determined from back-propagation. Thus, as long as we can draw random samples from the model, we have access to an unbiased Monte Carlo estimate of the log-likelihood gradient, which allows us to optimize the parameters with stochastic gradients.

Running MCMC until convergence to obtain samples can be computationally expensive. To make MCMC-based learning of EBMs practical, we need to make some approximations. One popular method is **Contrastive Divergence**, where the MCMC chain is initialized from some data point, and only a fixed number of MCMC steps are performed — typically fewer than what would be required for full convergence.

Note that in exponential family case, we do note need to perform these steps — it is sufficient to calculate the sufficient statistics themselves to fully characterize the distribution.

10.2.7 Another way to learn EBM — Score Matching

As discussed in Section 10.2.6, sampling is one approach to handle the partition function in EBMs. However, is there a way to avoid dealing with the partition function directly?

Maximum Likelihood Estimation (MLE) works well when the partition function can be easily computed, such as in logistic regression where there are only two possible outcomes. But when the integral involved is difficult to compute, MCMC is often required—though its performance can be suboptimal due to computational cost.

This motivates the search for MLE-free approaches to learn model parameters. There are several representative algorithms, one of which is **Score Matching**, and another is *Noise Contrastive Estimation (NCE)*, which we shall discuss in the representation learning section.

10.2.7.1 MLE as KL-Divergence

Consider the empirical distribution given by

$$\hat{\mathbb{P}}[\mathbf{x}] = \frac{1}{n}\sum_{i=1}^{n} \delta[\mathbf{x}_i] \tag{ED}$$

we note that then the liklihood estimation problem given in (MLE) be equivalently expressed as

$$\min_{\mathbf{\Theta}} \mathcal{KL}(\hat{\mathbb{P}} \| \mathbb{P}_{\mathbf{\Theta}}) = \min_{\mathbf{\Theta}} \left[\int \hat{\mathbb{P}}[\mathbf{x}] \log \frac{\hat{\mathbb{P}}[\mathbf{x}]}{\mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}]} \mathrm{d}\mathbf{x} \right] \tag{10.28}$$

$$= \min_{\mathbf{\Theta}} \left[\int \hat{\mathbb{P}}[\mathbf{x}] \log \hat{\mathbb{P}}[\mathbf{x}] \mathrm{d}\mathbf{x} - \int \hat{\mathbb{P}}[\mathbf{x}] \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \mathrm{d}\mathbf{x} \right] \tag{10.29}$$

Note that the $\int \hat{\mathbb{P}}[\mathbf{x}] \log \hat{\mathbb{P}}[\mathbf{x}] \mathrm{d}\mathbf{x}$ component is not a function of $\mathbf{\Theta}$ and so plays no part in the optimization. At which point we arrive at

$$\min_{\mathbf{\Theta}} \mathcal{KL}(\hat{\mathbb{P}} \| \mathbb{P}_{\mathbf{\Theta}}) = \max_{\mathbf{\Theta}} \int \hat{\mathbb{P}}[\mathbf{x}] \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \mathrm{d}\mathbf{x} \tag{10.30}$$

Note that this still involves a high dimensional integral involving the partition function, as it is essentially a reformulation of MLE. However, if you can find another divergence that does not involve the partition function, then you can avoid this issue.

Indeed, there is such a divergence — the **Fisher Divergence**.

$$\begin{aligned} \mathcal{D}_F(\hat{\mathbb{P}} \| \mathbb{P}_{\mathbf{\Theta}}) :=& \mathbb{E}_{\hat{\mathbb{P}}} \left[\frac{1}{2} \| \nabla_{\mathbf{x}} \log \hat{\mathbb{P}}[\mathbf{x}] - \nabla_{\mathbf{x}} \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \|_2^2 \right] \\ =& \frac{1}{2} \int \hat{\mathbb{P}}[\mathbf{x}] \| \nabla_{\mathbf{x}} \log \hat{\mathbb{P}}[\mathbf{x}] - \nabla_{\mathbf{x}} \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \|_2^2 \mathrm{d}\mathbf{x} \end{aligned} \tag{FD}$$

the *score matching estimator*

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \mathcal{D}_F(\mathbf{\Theta}) \tag{SME}$$

However, this may seem to be difficult to solve, because we have no estimator to the score function (gradient) to the empirical distribution $\hat{\mathbb{P}}$. We may tackle (FD) as follows

$$\mathcal{D}_F(\hat{\mathbb{P}} \| \mathbb{P}_{\mathbf{\Theta}}) := \frac{1}{2} \int \hat{\mathbb{P}}[\mathbf{x}] \| \nabla_{\mathbf{x}} \log \hat{\mathbb{P}}[\mathbf{x}] - \nabla_{\mathbf{x}} \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \|_2^2 \mathrm{d}\mathbf{x} \tag{10.31}$$

$$= \int \hat{\mathbb{P}}[\mathbf{x}] \left[\underbrace{\frac{1}{2} \| \nabla_{\mathbf{x}} \log \hat{\mathbb{P}}[\mathbf{x}] \|_2^2}_{\text{(A)}} + \underbrace{\frac{1}{2} \| \nabla_{\mathbf{x}} \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \|_2^2}_{\text{(B)}} - \underbrace{\langle \nabla_{\mathbf{x}} \log \hat{\mathbb{P}}[\mathbf{x}], \nabla_{\mathbf{x}} \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \rangle}_{\text{(C)}} \right] \mathrm{d}\mathbf{x} \tag{10.32}$$

In the program (SME), (A) is not a function of $\mathbf{\Theta}$ and so plays no part in the optimization, (B) is easy to approximate by sampling. (C) is only the problematic term, so consider

$$\int -\hat{\mathbb{P}}[\mathbf{x}] \langle \nabla_{\mathbf{x}} \log \hat{\mathbb{P}}[\mathbf{x}], \nabla_{\mathbf{x}} \log \mathbb{P}_{\mathbf{\Theta}}[\mathbf{x}] \rangle \mathrm{d}\mathbf{x} \tag{C}$$

To tackle the integral in (C), we recall the following *integration by parts* formula

$$\int_a^b u(\mathbf{x}) v'(\mathbf{x}) \mathrm{d}\mathbf{x} = [u(\mathbf{x}) v(\mathbf{x})] \Big|_a^b - \int_a^b u'(\mathbf{x}) v(\mathbf{x}) \mathrm{d}\mathbf{x} u = u(b)v(b) - u(a)v(a) - \int_a^b u'(\mathbf{x}) v(\mathbf{x}) \mathrm{d}\mathbf{x} u \tag{IP}$$

Under mild regularity conditions, and the multivariate generalization of (IP), allow $u(\mathbf{x}) = \nabla_{\mathbf{x}} \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{x}]$ and $v' = \nabla_{\mathbf{x}} \log \hat{\mathbb{P}}[\mathbf{x}]$, then (C) becomes

$$\underbrace{-\lim_{\mathbf{b}\to\infty} \nabla_{\mathbf{b}} \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{b}]\hat{\mathbb{P}}[\mathbf{b}]}_{=0} + \underbrace{\lim_{\mathbf{a}\to-\infty} \nabla_{\mathbf{a}} \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{a}]\hat{\mathbb{P}}[\mathbf{a}]}_{=0} + \int_{\infty}^{\infty} \nabla_{\mathbf{x}}^2 \mathbb{P}_{\boldsymbol{\Theta}} \hat{\mathbb{P}}[\mathbf{x}] \mathrm{d}\mathbf{x} \tag{10.33}$$

It follows that

$$\mathcal{D}_F(\hat{\mathbb{P}} \| \mathbb{P}_{\boldsymbol{\Theta}}) \propto \mathcal{L}(\boldsymbol{\Theta}) := \mathbb{E}_{\hat{\mathbb{P}}}\left[\mathrm{Tr}\left[\nabla_{\mathbf{x}}^2 \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{x}]\right] + \frac{1}{2}\|\nabla_{\mathbf{x}} \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{x}]\|_2^2\right] \tag{SML}$$

If $\hat{P}$ is given by the form in (ED), (SML) may be approximated as

$$\mathcal{L}(\boldsymbol{\Theta}) \approx \frac{1}{n}\sum_{i=1}^{n} \mathrm{Tr}\left[\nabla_{\mathbf{x}}^2 \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{x}_i]\right] + \frac{1}{2}\|\nabla_{\mathbf{x}} \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{x}_i]\|_2^2 \tag{10.34}$$

In this way, you avoid the partition function, at the cost requiring second order hessian information $\nabla_{\mathbf{x}}^2 \log \mathbb{P}_{\boldsymbol{\Theta}}[\mathbf{x}]$, which is not very memory efficient and expensive to compute. Thus the implicit score matching formulation given in (SML) is only applied to relatively simple energy functions, where the hessian is tractable.