CS 189/289A Introduction to Machine Learning
Fall 2024 Jennifer Listgarten, Saeed Saremi **DIS10**

1 Energy Function Motivation

Lots of generative models can be represented by probability distributions $p(x \mid w)$ where x is some data/input vector and w is a series of learnable parameters. In order to be a valid probability distribution, we need

$$\int p(x \mid w)\, dx = 1.$$

Consider an arbitrary function $E(x, w)$. In practice, $E(x, w)$ is modeled either through a neural network or another architecture (with parameters w and input x). The exponential $\exp(-E(x, w))$ is a non-negative quantity that can be viewed as an *un-normalized* probability distribution of x; higher energy values correspond to lower probabilities.

We then define

$$p(x \mid w) = \frac{1}{Z(w)} \exp(-E(x, w))$$

where

$$Z(w) = \int \exp(-E(x, w))\, dx.$$

(a) Given a dataset of i.i.d. samples $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, we would like to compute the log-likelihood $\log p(\mathcal{D} \mid w)$, so that we can calculate gradients later. In terms of $E(x_i, w)$ and $Z(w)$, what is the log-likelihood $\log p(\mathcal{D} \mid w)$?

Solution:

$$\log p(\mathcal{D} \mid w) = \log \prod_{i=1}^{n} p(x_i \mid w) = \log \prod_{i=1}^{n} \frac{1}{Z(w)} \exp(-E(x_i, w)) = \sum_{i=1}^{n} \Big[-E(x_i, w) - \log(Z(w))\Big].$$

(b) From the above we can calculate

$$\mathbb{E}_{x \sim p_{\mathcal{D}}}[\nabla_w \log p(x \mid w)] = -\mathbb{E}_{x \sim p_{\mathcal{D}}}[\nabla_w E(x, w)] - \nabla_w \log Z(w),$$

since the samples x_i are i.i.d. from some distribution $p_{\mathcal{D}}$. Note that the second term doesn't depend on $\mathcal{D}$ and only depends on the normalizing function $Z(w)$.

Show that $-\nabla_w \log Z(w) = \int \nabla_w E(x, w) p(x \mid w)\, dx$.

Solution: Note that $-\nabla_w \log Z(w) = \frac{-\nabla_w Z(w)}{Z(w)}$.

The numerator is equal to

$$-\nabla_w \int \exp(-E(x,w))\,dx = -\int \nabla_w \exp(-E(x,w))\,dx = \int \nabla_w E(x,w) \exp(-E(x,w))\,dx.$$

We commute integrals and gradients together.

Combining with the denominator gives

$$-\nabla_w \log Z(w) = \int \nabla_w E(x,w) \frac{\exp(-E(x,w))}{Z(w)}\,dx = \int \nabla_w E(x,w) p(x \mid w)\,dx$$

.

(c) The above shows that $-\nabla_w \log Z(w) = \mathbb{E}_{x \sim p(x|w)}[\nabla_w E(x,w)]$. Use this result and the result from part (a) to derive a simplified expression for $\mathbb{E}_{x \sim p_{\mathcal{D}}}[\nabla_w \log p(x \mid w)]$. The result you get will form a basis for why we use Langevin sampling to help approximate this gradient.

Solution: Combining (b) and (c) gives that

$$\mathbb{E}_{x \sim p_{\mathcal{D}}}[\nabla_w \log p(x \mid w)] = -\mathbb{E}_{x \sim p_{\mathcal{D}}}[\nabla_w E(x,w)] + \mathbb{E}_{x \sim p(x|w)}[\nabla_w E(x,w)].$$

We see that both terms in this expression are over the same expression, $\nabla_w E(x,w)$, but over different distributions. Furthermore, they are opposing in sign.

In regions where the model density—$p(x \mid w)$—exceeds the true data density—$p_{\mathcal{D}}$— the net effect will be to increase the energy function and reduce the probability. Conversely, when the data density exceeds the model density, the net effect will be to decrease the energy function and increase the probability density. The two regions are equal when the data density equals the model density, at which point the gradient is equal to 0.

(d) How can we use Langevin sampling to help approximate the gradient?

Solution: The first part of the gradient comes from our dataset and can be approximated as an average of $\nabla_w E(x_i, w)$ given i.i.d. samples x_i. The second term is something that must be modeled, since we do not have access to the distribution $p(x \mid w)$. Note that given an input x and a sample w calculating $\nabla_w E(x,w)$ is tractable (for example, assume $E(x,w)$ is a neural net, and then we can run back-propagation on it). The issue is getting a representative distribution to sample these gradients from.

Langevin sampling helps us in the case where we want to approximate a target distribution $p(x \mid w) = \frac{-E(x,w)}{Z(w)}$ by starting with some x_0 and an update equation. Running it for a long time converges to a sample from the distribution $p(x \mid w)$. We can repeat this process many times to obtain a list of samples $x \sim p(x \mid w)$ and then evaluate $\nabla_w E(x,w)$ for each of them to then approximate $\mathbb{E}_{x \sim p(x|w)}[\nabla_w E(x,w)]$ through averaging.

In this case, Langevin sampling helps us deal with the distribution $p(x \mid w)$ to help generate samples from it to approximate gradients.

2 (un-adjusted) Langevin Sampling

We'll now go through a concrete example of applying Langevin sampling when the target distribution is Gaussian. We'll analyze the convergence rate of this algorithm as well as the actual distribution it converges to.

Recall that the Langevin update equation is given by

$$x_{t+1} = x_t + \eta \nabla_x \log p(x_t) + \sqrt{2\eta} \cdot \epsilon_t$$

where $\epsilon_t \sim N(0, I)$, $\nabla_x \log p(x_t)$ is a *score function*, and $\eta > 0$ is the step size.

This looks like a "noisy" version of gradient ascent where we add noise in every step. While gradient ascent will always find a local maxima (given appropriate step sizes), this algorithm often converges upon somewhere near a local maxima but due to the stochastic nature sometimes ends up in smaller probability regions.

For the rest of this question, assume

$$f(x) = \frac{1}{2}(x-\mu)^\top H(x-\mu)$$

We're going to apply Langevin sampling to this function to show that we eventually converge to a sample from $p(x) = \frac{\exp(-f(x))}{Z}$ where $Z = \int \exp(-f(x))\, dx$. Note that $p(x)$ is a Gaussian distribution centered at μ with covariance matrix H^{-1}.

(a) Derive $\nabla_x \log(p(x))$.

Solution: The score function can be simplified as follows

$$\begin{aligned}\nabla_x \log(p(x)) &= -\nabla_x f(x) - \nabla_x \log(Z) \\ &= -\nabla_x f(x) - 0 \\ &= -H(x-\mu)\end{aligned}$$

Note that Z is a constant so the gradient is zero w.r.t. x.

(b) Rewrite the score function using the gradient above. What is $x_t - \mu$ in terms of $x_0 - \mu$?

Solution: Let's try unrolling the equation. It's easier to work with the quantity $x_t - \mu$, so let's find an expression for that.

$$x_t - \mu = (x_{t-1} - \mu) - \eta H(x_{t-1} - \mu) + \sqrt{2\eta}\epsilon_{t-1} = (I - \eta H)(x_{t-1} - \mu) + \sqrt{2\eta}\epsilon_{t-1}.$$

Note that this is just a linear transformation of $x_{t-1} - \mu$ by the matrix $I - \eta H$, with some additional noise added.

Expanding for x_t in terms of x_{t-2} gives

$$x_t - \mu = (I - \eta H)\left((I - \eta H)(x_{t-2} - \mu) + \sqrt{2\eta}\epsilon_{t-2}\right) + \sqrt{2\eta}\epsilon_{t-1}$$

$$= (I - \eta H)^2 (x_{t-2} - \mu) + (I - \eta H)\sqrt{2\eta}\epsilon_{t-2} + \sqrt{2\eta}\epsilon_{t-1}.$$

We can see a pattern start to emerge. We will have a $(I - \eta H)^t (x_0 - \mu)$ term combined with noise terms that are weighted by successive powers of the matrix $I - \eta H$. Writing this out in terms of x_0 gives

$$x_t - \mu = (I - \eta H)^t (x_0 - \mu) + \sqrt{2\eta} \sum_{i=0}^{t-1} (I - \eta H)^{t-1-i} \epsilon_i$$

(c) What is the expectation of x_t? What does this approach as $t \to \infty$? Assume that the eigenvalues of $I - \eta H$ are upperbounded in absolute value by 1 (i.e. η is set appropriately small).

Solution:

$$\begin{aligned} \mathbb{E}[x_t - \mu] &= \mathbb{E}[(I - \eta H)^t (x_0 - \mu) + \sqrt{2\eta} \sum_{i=0}^{t-1} (I - \eta H)^{t-1-i} \epsilon_i] \\ &= (I - \eta H)^t (x_0 - \mu) + \sqrt{2\eta} \sum_{i=0}^{t-1} (I - \eta H)^{t-1-i} \mathbb{E}[\epsilon_i] \\ &= (I - \eta H)^t (x_0 - \mu) \end{aligned}$$

since the noise has mean 0. As $t \to \infty$ this approaches 0 since the eigenvalues are bounded by 1 in absolute value. So $\mathbb{E}[x_t - \mu] = 0 \implies \mathbb{E}[x_t] = \mu$ as $t \to \infty$.

(d) To simplify calculations, assume $H = I$ (i.e. the covariance of $p(x)$ is just the identity matrix). What is the covariance of x_t, i.e., $\mathbb{E}[(x_t - \mu)(x_t - \mu)^\top]$ as $t \to \infty$? What do you notice about this? As $\eta \to 0$ what does this approach?

Solution: Using the expresion for $x_t - \mu$ from part (b), we get that

$$\mathbb{E}[(x_t - \mu)(x_t - \mu)^\top] = (I - \eta I)^t (x_0 - \mu)(x_0 - \mu)^\top (I - \eta I)^t + 2\eta \sum_{i=0}^{t-1} (I - \eta I)^{t-1-i} \mathbb{E}[\epsilon_i \epsilon_i^\top] (I - \eta I)^{t-1-i}.$$

The other terms vanish because $E[(x_0 - \mu)\epsilon_i^T] = 0$ and $E[\epsilon_i \epsilon_j^T] = 0$ for $i \neq j$. We also use the fact that $(I - \eta I)$ is a symmetric matrix (i.e. $(I - \eta I)^\top = (I - \eta I)$. Using the fact that $E[\epsilon_i \epsilon_i^T] = I$ we can simplify the expression to

$$(I - \eta I)^t (x_0 - \mu)(x_0 - \mu)^T (I - \eta I)^t + 2\eta \sum_{i=0}^{t-1} (I - \eta I)^{2(t-1-i)}$$

.

The term $(I - \eta I)^t (x_0 - \mu)(x_0 - \mu)^\top (I - \eta I)^t \to 0$ as $t \to \infty$.

To decompose the other summation, note that as $t \to \infty$,

$$\sum_{i=0}^{t-1} (I - \eta I)^{2(t-1-i)} = \sum_{i=0}^{\infty} (I - \eta I)^{2i} = I \sum_{i=0}^{\infty} (1 - \eta)^{2i}.$$

This sum is geometric with ratio $(1-\eta)^2$ and thus converges to $\frac{1}{1-(1-\eta)^2}$.

Combining this with the above, we get that

$$2\eta \sum_{i=0}^{t-1} (I - \eta I)^{2(t-1-i)} \approx \frac{2\eta}{1-(1-\eta)^2} I = \frac{2\eta}{2\eta - \eta^2} = \frac{2}{2-\eta} I$$

as $t \to \infty$.

Thus, the covariance approaches $\frac{2}{2-\eta} I$ (*overshoots it*). Note that this is actually *not* equal to I which is the expected covariance if converging to the distribution of $p(x)$.

However, as $\eta \to 0$, this approaches I as expected. This illustrates that our step size, η should not be too large because then the distribution we converge to is not our approximate distribution. At the same time, we don't want η to be too small because we can view this process as noisy gradient descent and would like to sample less if possible for convergence guarantees.