DS 102 Data, Inference, and Decisions **Fall 2019**

Lecture 10: Approximate Inference via Sampling

Lecturer: Jacob Steinhardt

10.1 Review and Motivation

In the last few lectures, we introduced the idea of latent variable models, where in addition to observed data, x, and unknown hyperparameters, θ, there is also hidden structure (modeled via latent variables z) which we would like to make inferences about. We saw three examples:

1. Hierarchical Bayesian models, where there is independent latent structure for every observation;
2. Hidden Markov models, where latent structure has a time component; and
3. an election forecasting model which incorporates unknown biases for every pollster.

We also outlined two frameworks for performing inference in these latent variable models. One option was to perform maximum likelihood estimation, and the other was to be "fully Bayesian" by introducing a prior on θ and sampling $\mathbb{P}(\theta, z|x)$.

We focused on the first approach in the previous lecture. Specifically, we discussed how to perform maximum likelihood estimation when we observe data x that is generated with respect to some hidden or latent variables z. We saw that, in order to obtain a posterior probability distribution over z after observing the data x,

$$\mathbb{P}(z|x, \theta) = \frac{\mathbb{P}(x \mid z, \theta) \cdot \mathbb{P}(z|\theta)}{\mathbb{P}(x|\theta)},$$

we must compute $\mathbb{P}(x|\theta) = \sum_{z \in \mathcal{Z}} \mathbb{P}(x \mid z, \theta) \cdot \mathbb{P}(z|\theta)$ (or $\mathbb{P}(x|\theta) = \int_{z \in \mathcal{Z}} \mathbb{P}(x \; |z, \theta) \cdot \mathbb{P}(z|\theta) dz$ for continuous random variables). In general, calculating the normalizing constant $\mathbb{P}(x|\theta)$ exactly can be quite difficult; it may require computing a sum over an exponential number of possibilities (or otherwise computing an integral with no closed-form solution). We saw that the Expectation-Maximization (EM) algorithm addresses precisely this problem.

Today, we will focus on the second, fully Bayesian approach. One potential issue with this approach is that, if the conditional distributions do not turn out to have a nice closed form, the distribution of interest, $\mathbb{P}(\theta, z|x)$, may not be easy to work with algebraically. This lecture will introduce methods for *approximating* distributions via sampling.

In general, sampling is a very powerful idea. Having many samples from a distribution is a human-interpretable way to represent a distribution. Moreover, while sampling is not always easy, the problem of obtaining samples from an arbitrary probability distribution is of lower complexity than that of obtaining closed-form expressions for arbitrary distributions. With samples, we can also approximate any statistic $\mathbb{E}_{x\sim\mathbb{P}}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$, where the x_i are n samples from $\mathbb{P}$.

In the next lecture, our discussion of sampling algorithms will culminate in the Metropolis-Hastings algorithm, which is an example of a Markov Chain Monte Carlo (MCMC) algorithm. Compared to EM, which is useful whenever we are lucky enough that both the E and M steps can be computed efficiently, Metropolis-Hastings is a much more flexible family of algorithms that can be applied to a wide variety of problems. In the current lecture, we will focus on two simpler sampling algorithms, rejection sampling and importance sampling, which introduce the basic tools that will be relevant for studying Metropolis-Hastings.

10.2 Rejection Sampling

Example 10.1. Imagine you want to sample uniformly from the joint distribution $p(x_1, x_2) \propto \mathbb{I}\{x_1^2 + x_2^2 \leq 1\}$, that is, you want to sample uniformly from points (x_1, x_2) that lay within the unit circle (blue region in Figure 10.1). Sampling directly from that distribution is quite difficult, and you might wish that you had been asked instead to sample from the unit square (grey region in Figure 10.1) – if this were the task, you could first sample $x_1 \sim Uniform(-1, 1)$ and then sample $x2 \sim Uniform(-1, 1)$ as independent draws, and the concatenation (x_1, x_2) would be drawn uniformly from the unit square.

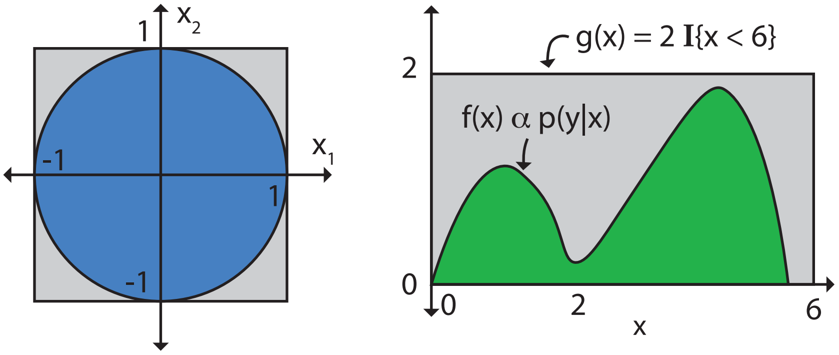

Figure 10.1: Two illustrations of sampling problems. The colored regions denote distributions we'd like to sample from, grey regions denote areas it is easy to sample from.

In fact, we can use the ease of sampling on the unit square to our advantage. In particular, we can sample candidate pairs (z_1, z_2) uniformly from the unit square as above, but then only put them in our sample if $z_1^2 + z_2^2 \leq 1$. This procedure will only give us points within the unit circle, and furthermore, the probability is uniform across the unit circle.

Example 10.1 illustrates the key idea behind rejection sampling, which is to generate many samples according to a distribution that is easy to sample from, and then discard samples at a rate (called the *selection rate*) to match the second distribution. In the unit circle example, the selection rate is an indicator function (0 if candidate point (z_1, z_2) falls outside the circle, 1 otherwise). In general, rejection sampling allows a selection rate that is any number in the continuous range $[0, 1]$.

More generally, **rejection sampling** is an algorithm for sampling from a target distribution $p(x) \propto f(x)$, where $f(x)$ is a known function that does not have to be a probability distribution (i.e. it does not need to integrate to 1), by making use of a proposal distribution $q(x)$. To use rejection sampling, we must have $q(x) \geq f(x) \, \forall f(x) > 0$. Algorithmically, rejection sampling takes the following steps:

rejection sampling

```
1. repeat:
    (a) Generate sample $x$ from $q(\cdot)$.
    (b) Calculate the acceptance probability: $\alpha = \frac{f(x)}{q(x)}$
    (c) With probability $\alpha$, put $x$ in the sample, otherwise reject it.
```

In our unit circle example, $f(x) = \mathbb{I}\{x_2^2 + y_1^2\} \leq 1$ (the indicator of being within the unit circle), and $q(x) = \mathbb{I}\{\max(|x_1|, |x_2|) < 1\}$ (the indicator of being within the unit square). Again, in this example, α is either 1 or 0, but in general this need not be the case, as demonstrated in Example 10.2.

Example 10.2. Suppose we want to sample from a univariate posterior probability $\mathbb{P}(z|x) =\propto \mathbb{P}(x|z)\mathbb{P}(z)$, where we have already specified a likelihood $\mathbb{P}(x|z)$ and a prior $\mathbb{P}(z)$, but the prior is not a conjugate prior to the likelihood. The posterior probability, itself, is unknown, but we do know $f(z) = \mathbb{P}(x|z)\mathbb{P}(z)$. We can thus use the rejection sampling algorithm above to get a sample which we can use to approximate the posterior, as shown in Figure 10.1. Note that in Figure 10.1, the fraction of time we accept a candidate point drawn from the grey distribution is equal to the area of the colored distribution over the area of the grey distribution.

Despite being a simple and powerful idea, rejection sampling alone does not get us very far except in very simple cases. The utility of rejection sampling is limited primarily by the following two issues:

1. Rejection sampling is often very wasteful. For instance, in Example 10.2 the selection rate (given by the ratio of the green area to the total grey area in the right-hand panel of Figure 10.1) is roughly $\frac{1}{2}$. This means that we reject roughly half of all candidate samples. This phenomenon may not seem too bad in one dimension, but in high-dimensional sampling settings, rejection sampling often rejects the vast majority of proposals. We will discuss ways to address this issue in more detail in the next lecture, but it turns out that we need to be careful about the way we design our proposal distribution.

2. Recall that, in order to use rejection sampling, we require that $q(x) \geq f(x)$ and $p(x) \propto f(x)$, where $f(x)$ is *known*. This means that we must have $p(x) \leq C \cdot q(x)$ where the constant of

proportionality C is known. It turns out that the problem posed by not knowing C can be solved quite easily by using importance sampling, which we discuss in the next section.

10.3 Importance Sampling

Taking a closer look at Figure 10.1 (right), we see that most of the samples we get from the grey distribution defined by $q(x)$ will actually tell us *something* about the distribution $p(x)$. Some points (e.g. in $x = 2$) are less likely, so we would want to somehow down-weight those samples according to their probability of being sampled from a density proportional to $p(\cdot)$. This is exactly what importance sampling does.

Importance sampling approximates a target distribution distribution $p(x)$ by sampling points from a proposal distribution $q(x)$, and weighting the sampled points by their likelihood ratio $\frac{p(x)}{q(x)}$:

```
importance sampling

1. repeat:
    (a) Generate sample $x^{(i)}$ from $q(\cdot)$.
    (b) Calculate the importance weight $w(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})}$.
    (c) Put $x^{(i)}$ in the sample, and store weight $w(x^{(i)})$ as well.
```

Importance sampling looks very similar to rejection sampling, except now every point we sample from $q(\cdot)$ is getting used in our sample. Then, to estimate some function, say $\phi(x)$, in expectation over the population, we would weight each example in our sample average:

$$\mathbb{E}_{x\sim p}\left[\phi(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n} w(x^{(i)})\phi(x^{(i)}).$$

In fact, such an estimate is unbiased:

$$\begin{aligned}
\mathbb{E}_{x^{(0)},\ldots,x^{(i)} \overset{i.i.d.}{\sim} q(x)}\left[\frac{1}{n}\sum_{i=1}^{n} w^{(i)}\phi(x^{(i)})\right] &= \mathbb{E}_{x^{(0)},\ldots,x^{(i)} \overset{i.i.d.}{\sim} q(x)}\left[\frac{p(x)}{q(x)}\cdot\phi(x)\right] \\
&= \int_{x\in\mathcal{X}} \frac{p(x)}{q(x)}\phi(x)q(x)dx \\
&= \int_{x\in\mathcal{X}} \phi(x)p(x)dx \\
&= \mathbb{E}_{x\sim p(x)}\left[\phi(x)\right].
\end{aligned}$$

In practice we'd almost always prefer importance sampling to rejection sampling. For one, we do not need to know the constant of proportionality C and do not need the normalization constant for our target distribution. Moreover, importance sampling has lower variance, and is especially helpful for sampling low probability events, where the probability of rejection in rejection samplings would be high. In general, importance sampling ideas are very powerful and show up in many areas of statistics and probability, as well as across scientific domains.