Sampling Techniques

Krishna Balasubramanian

May 18, 2020

1 Introduction

A large part of statistics depends on computing expectations. As a few examples consider:

- In statistics, you assume the data set (denoted as $\mathcal{D}$ and consisting of n samples) is assumed to come from some probability distribution with parameter $\boldsymbol{\theta}$. In Bayesian statistics, it is further assumed that the parameter $\boldsymbol{\theta}$ is also drawn from another distribution. Often times, we are interested in computing some statistic $(g(\cdot))$ of the random vector $\boldsymbol{\theta}$ or its expected value. In this case, the posterior distribution is given by $\mathsf{E}\left(g(\boldsymbol{\theta})|\mathcal{D}\right) = \int_{\boldsymbol{\theta}} g(\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$.
- Marginalizing over missing data $p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \mathsf{E}_{p(\mathbf{z})}[p(\mathbf{x}|\mathbf{z})]$
- Computing probabilities $p(\mathbf{x} \in C) = E_p(\mathbf{I}_C)$, where $\mathbf{I}_C$ is the indicator set.
- Simulated annealing and related techniques.
- Computing regular deterministic integrals, by writing the integrand as a product of a function and a probability density function.

1.1 Basic Idea

When the expectation (integral or summation) does not have a closed form we need to resort to approximation techniques. One class of approximation methods is numerical integration techniques such as the trapezoid or Simpson's method. A second class of approximation methods, which we will concentrate on, are based on Monte Carlo sampling and the law of large numbers:

$$\frac{1}{m}\sum_{i=1}^{m} g(\mathbf{x}_i) \approx \mathsf{E}_{p(\mathbf{x})}[g(\mathbf{x})] \quad \text{if} \quad \mathbf{x}_1, \ldots, \mathbf{x}_m \sim p(\mathbf{x}).$$

The above estimator is **unbiased** and has **variance**:

$$\frac{\mathsf{Var}\left(g(\mathbf{x})\right)}{m}$$

The convergence above is quite stable and rapid (as indicated by the uniform law of large numbers and large deviation theory) and does not depend on the dimensionality of $\mathbf{x}$. Slow convergence may occur, however, if $g(\cdot)$ is high where $p(\cdot)$ is low and vice verse.
The benefit of sampling methods over numerical integration methods is that they work better in high dimensional cases. But it depends crucially on our ability to sample from $p(\mathbf{x})$. Some high dimensional models are easy to sample from where as in other high dimensional models such as exponential family models or Markov random fields, sampling is not straightforward. We now look at some sampling techniques.

1.2 Histogram Method

In general, we will assume that we can sample from a uniform $U([0,1])$ distribution. Sampling from a uniform distribution has been widely studied and many efficient methods for doing so exist. To sample from a discrete one dimensional RV $\mathbf{x} \in \mathbb{R}$, we can just generate a $r \sim U([0,1])$ random number and compare it with cdf F_X and sample x_i for which $r_i \in [F_X(x_i), F_X(x_{i+1})]$. The above method can be applied to continuous RVs by discretizing them (approximating a continuous RV by its discrete histogram). The method works well for one dimensional RVs but suffers greatly in high dimensional cases.

1.3 Transformation Method

We focus on the case of one dimensional distribution. Extensions to high dimensionality models are straightforward. Assume we can sample from a uniform RV in $[0,1]$ and we wish to sample from a RV $\mathbf{x}$. The transformation $\mathbf{x} \mapsto F_{\mathbf{x}}(\mathbf{x})$ results in a uniform RV

$$P(F_{\mathbf{x}}(\mathbf{x}) \leq r) = P(F_{\mathbf{x}}^{-1}(F_X(\mathbf{x})) \leq F_{\mathbf{x}}^{-1}(r)) = P(\mathbf{x} \leq F_{\mathbf{x}}^{-1}(r)) = F_{\mathbf{x}}(F_{\mathbf{x}}^{-1}(r)) = r.$$

As a result transforming the uniform samples by $r \mapsto F^{-1}(r)$ results in samples from X. Technically, there is a problem with the method as stated above if the pdf or pmf of X is not strictly positive (F_X is not invertible). A more careful method statement should resolve that difficulty. In low dimensional cases, the above transformation works well. The basic problem of computing F_X and inverting it become difficult in high dimensions and other methods are necessary.

1.4 What about Discrete random variables

It turns out that there is a neat trick based on Gumbel distribution that could be used to sample from a discrete random variable $\mathbf{x}$ taking values in $\{1, \dots, K\}$ with probability proportional to $p_1, \dots, p_k$ (note just known upto a normalization constant is enough for p_k.). A random variable $\mathbf{g} \in \mathbb{R}$ is called as standard Gumbel distribution if $\mathbf{g} = -\log(-\log(r))$ where $r \sim U[0,1]$. We now have the following fact.

Fact 1.0.1. Let $\mathbf{x}$ be a discrete random variable taking values in $\{1, \dots, K\}$ with $P(\mathbf{x} =$

$k) \propto p_k$ and let $\mathbf{g}_k$ be a i.i.d. sequence of standard Gumbel random variables. Then

$$\mathbf{x} = \arg\max_k(\log(p_k) + \mathbf{g}_k)$$

Hence, we have the following method for sampling:

1. Draw K i.i.d. standard Gumbel random variables, $\mathbf{g}_1, \ldots, \mathbf{g}_K$.
2. Add $\log(p_k)$ to the Gumbel random variables.
3. Take the value of $k \in \{1, \ldots, K\}$ that produes the maximum.

1.5 Rejection Sampling

Again, we start with a one dimensional formulation. Assume that we want to sample from p, but we can sample from q instead, and we know further that $p(\mathbf{x}) \leq kq(\mathbf{x})$ everywhere for some constant k. Sampling $\mathbf{x}_i \sim q$ and then $r_i \sim U([0, kq(\mathbf{x}_i)])$ would give a pair $(\mathbf{x}_i, r_i)$ which would be uniformly distributed over the graph of the function (area under the function curve) kq. By rejecting the pair if $r_i \geq p(\mathbf{x})$ we ensure that the remaining sample pairs are uniformly distributed over the graph of $p(\mathbf{x})$. We can then discard the r_i and keep the $\mathbf{x}_i$ samples which constitute a sample from p. Rejection sampling can be modified for use if p is known up to a constant $p = c\tilde{p}$ (its normalization term is not easily computable). In this case we find q such that $\tilde{p} \leq kq$ and proceed as before.

Adaptive rejection sampling is way of adaptively computing q and k for distributions p whose logarithm is concave. In this case, we can upper bound the $\log p$ with a piecewise linear function (envelope) computed based on the derivative $\nabla \log p$ at different points. The distribution itself p is then upper bounded by a piecewise exponential function which constitutes the proposal q. As samples get rejected, they are added to the computation of the envelope and the quality of the upper bound improves. The difficulty here is as before in cases of high dimensionality.

1.6 Importance Sampling

Importance sampling directly estimates the expectation $\mathsf{E}_p(g)$ by noticing that

$$\mathsf{E}_p(g) = \int g(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})\,d\mathbf{x} = \mathsf{E}_q(g\,p/q)$$

which is approximated by averaging $g(\mathbf{x})\,p(\mathbf{x})/q(\mathbf{x})$ over samples from $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim q$. A useful trick is performing importance sampling when we can't evaluate the normalization terms of p and q. In this case $p = \tilde{p}/Z_p$ and $q = \tilde{q}/Z_q$ and

$$\mathsf{E}_p(g) = \frac{Z_q}{Z_p} \int g(\mathbf{x}) \frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})} q(\mathbf{x})\, d\mathbf{x} \approx \frac{Z_q}{Z_p} \frac{1}{m} \sum_{i=1}^{m} \frac{\tilde{p}(\mathbf{x}_i)}{\tilde{q}(\mathbf{x}_i)} g(\mathbf{x}_i)$$

where $\mathbf{x}_i$ are samples from q. The factor Z_q/Z_p may be approximated as follows

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q} \int \tilde{p}(\mathbf{x})\, d\mathbf{x} = \int \frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})} q(\mathbf{x})\, d\mathbf{x} \approx \frac{1}{m} \sum_{i=1}^{m} \frac{\tilde{p}(\mathbf{x}_i)}{\tilde{q}(\mathbf{x}_i)}$$

where $\mathbf{x}_i \sim q$. Putting all this together gives

$$\mathsf{E}_p(g) \approx \sum_{i=1}^{m} w_i g(\mathbf{x}_i) \qquad w_i = \frac{\tilde{p}(\mathbf{x}_i)/\tilde{q}(\mathbf{x}_i)}{\sum_{i=1}^{m} \tilde{p}(\mathbf{x}_i)/\tilde{q}(\mathbf{x}_i)} \qquad \mathbf{x}_i \sim q.$$

As before, the main problem is high dimensions. If p, q are high dimensional, weights $p(\mathbf{x}_i)/q(\mathbf{x}_i)$ become smaller rapidly. If q is low where pg is high, the estimator will be highly inaccurate since it may take a long time to obtain a meaningful sample.

2 Markov Chain Monte Carlo

We previously saw how samples can be used to approximate expectations

$$\frac{1}{m} \sum_{i=1}^{m} g(\mathbf{x}_i) \approx \mathsf{E}_p(g(\mathbf{x})) \quad \text{where} \quad \mathbf{x}_1, \ldots, \mathbf{x}_m \sim p.$$

We also saw a number of techniques for producing samples from a distribution p such as the histogram and transformation methods and rejection and importance sampling. Markov chain Monte Carlo (MCMC) is a collection of sampling methods that are based on following random walks on Markov chains.

Brief Introduction to Markov Chains: Homogenous Markov chains $X_0, X_1, X_2, \ldots$ are random processes that are completely characterized by the transition probabilities $P(X_n = y | X_{n-1} = z) = T(z, y)$ and initial probabilities $\pi_0(z) = P(X_0 = z)$. To simplify the notation we will assume that X_i are discrete and finite $X_i \in \{1, \ldots, k\}$ and we will consider π and T as a (row) vector and matrix of probabilities. For homogenous Markov processes conditional distribution of X_n given X_{n-1} is independent of $X_1, \ldots, X_{n-2}$. As a result, we have $P(X_1) = \pi_1 = \pi_0 T$. Similarly, $\pi_k = \pi_0 T^k$ and for large k, π_k tends to a unique stationary distribution π satisfying $\pi T = \pi$ (regardless of π_0) if the Markov chain characterized by T is ergodic. In other words, no matter what is the initial distribution π_0 (or where we start from) the resulting position distribution after k steps tends to the stationary distribution π for large k.

2.1 MCMC

The idea of MCMC is to generate a random sample from p by following a random walk of k steps on a Markov chain T, for which p equals its stationary distribution π. Thus, no matter where we start, if we follow a random walk for a long period (called burn-in time) we will end up with a sample from its stationary distribution. If we want several samples, we can either (i) repeat the process several times (ii) take consecutive samples after the burn in time or (iii) follow a random walk and record every l-step as a sample. Approach (ii) will not produce independent samples and approach (iii) will result in approximately independent samples from π if l is sufficiently large.

To sample from p using MCMC, we need to design a Markov chain T whose stationary distribution π is p. To ensure that, it suffices to show that T satisfies the detailed balance property with respect to p

$$p_i T_{ij} = p_j T_{ji} \quad \forall\, i, j$$

since then

$$[pT]_i = \sum_j p_j T_{ji} = \sum_j p_i T_{ij} = p_i \sum_j T_{ij} = p_i \quad \Rightarrow \quad pT = p.$$

We also need to ensure that the Markov chain described by T is ergodic so there will be a unique stationary distribution. One simple way to ensure ergodicity of T is to have $T_{ij} > 0$ for all i, j. It is useful to know (and easy to verify) that if we have several Markov chains $T_1, \ldots, T_l$ that satisfy the detailed balance property then a linear combination of them $\sum_i \alpha_i T_i$ would also satisfy it.

2.2 The Metropolis-Hastings Algorithm

The Metropolis-Hastings sampling model constructs an ergodic Markov chain that satisfies the detailed balance property with respect to p and therefore produce the appropriate samples. The transition T is based on sampling from a proposal conditional distribution $q(\mathbf{x}|\mathbf{x}^{(t)})$ (which we assume may be easily done). Specifically, given the t-step in the random walk $\mathbf{x}^{(t)}$ we generate the next step $\mathbf{x}^{(t+1)}$ as follows:

$$\mathbf{x}^{(t+1)} = \begin{cases} \mathbf{x}' & \text{with probability } r(\mathbf{x}^{(t)}, \mathbf{x}') = \min\left(1, \frac{p(\mathbf{x}')}{p(\mathbf{x}^{(t)})} \frac{q(\mathbf{x}^{(t)}|\mathbf{x}')}{q(\mathbf{x}'|\mathbf{x}^{(t)})}\right) \\ \mathbf{x}^{(t)} & \text{with probability } 1 - r(\mathbf{x}^{(t)}, \mathbf{x}') \end{cases}$$

where $\mathbf{x}' \sim q(\mathbf{x}|\mathbf{x}^{(t)})$. The two stage process results in the following Markov transition

$$T(\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}) = r(\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}) q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) + \left(1 - \sum_{\mathbf{x}'} r(\mathbf{x}^{(t)}, \mathbf{x}') q(\mathbf{x}'|\mathbf{x}^{(t)})\right) \delta_{\mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}}.$$

T is ergodic if $q(\mathbf{x}|\mathbf{x}^{(t)}) > 0$ and detailed balance w.r.t p holds since T above is written as a sum of two matrices that satisfy the detailed balance property w.r.t p

$$\begin{aligned} p(\mathbf{x})r(\mathbf{x},\mathbf{x}')q(\mathbf{x}'|\mathbf{x}) &= \min\left(p(\mathbf{x})q(\mathbf{x}'|\mathbf{x}), p(\mathbf{x})q(\mathbf{x}'|\mathbf{x})\frac{p(\mathbf{x}')}{p(\mathbf{x})}\frac{q(\mathbf{x}|\mathbf{x}')}{q(\mathbf{x}'|\mathbf{x})}\right) \\ &= \min(p(\mathbf{x})q(\mathbf{x}'|\mathbf{x}), p(\mathbf{x}')q(\mathbf{x}|\mathbf{x}')) \\ &= p(\mathbf{x}')r(\mathbf{x}',\mathbf{x})q(\mathbf{x}|\mathbf{x}') \end{aligned}$$

$$p(\mathbf{x})\left(1 - \sum_{\mathbf{x}'} r(\mathbf{x},\mathbf{x}')q(\mathbf{x}'|\mathbf{x})\right)\delta_{\mathbf{x},\mathbf{x}'} = p(\mathbf{x}')\left(1 - \sum_{z} r(\mathbf{x}',\mathbf{x})q(\mathbf{x}|\mathbf{x}')\right)\delta_{\mathbf{x}',\mathbf{x}}$$

In practice, the proposal is often taken to be a Gaussian $q(\mathbf{x}'|\mathbf{x}) = N(\mathbf{x}'\,;\mathbf{x},\sigma^2 I)$ which is an easy distribution to sample from (for example using the Box-Müller method[1]). In this and related other case $q(\mathbf{x}'|\mathbf{x}) = q(\mathbf{x}|\mathbf{x}')$ and the acceptance probability simplifies to $\min(1, \frac{p(\mathbf{x}')}{p(\mathbf{x}^{(t)})})$ which demonstrates that if the proposed state is more likely than the old one, it is accepted with probability 1. If the proposed state $\mathbf{x}'$ is less likely than the current one $\mathbf{x}^{(t)}$, the probability of accepting depends on the likelihood ratio $p(\mathbf{x}')/p(\mathbf{x}^{(t)})$. Choosing a proposal with small variance (for example $\sigma^2 \to 0$ for the above Gaussian proposal) would result in relatively high acceptance rates but with strongly correlated consecutive samples. Increasing the variance would de-correlate consecutive accepted samples to some extent, but it is also likely to reduce the acceptance rate.

2.3 Gibbs Sampling

Gibbs sampling is a special case of Metropolis-Hastings where the proposal q is based on the following two stage procedure. First, a single dimension i of $\mathbf{x}$ is chosen randomly (say uniformly). The proposed value $\mathbf{x}'$ is identical to $\mathbf{x}$ except for its value along the i-dimension $\mathbf{x}_i$ is sampled from the conditional $p(\mathbf{x}_i|\mathbf{x}_{-i}^{(t)})$ where $\mathbf{x}_{-i}^{(t)} = \{\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_{i-1}^{(t)}, \mathbf{x}_{i+1}^{(t)}, \ldots, \mathbf{x}_m^{(t)}\}$. Since

$$\frac{p(\mathbf{x}')}{p(\mathbf{x}^{(t)})}\frac{q(\mathbf{x}^{(t)}|\mathbf{x}')}{q(\mathbf{x}'|\mathbf{x}^{(t)})} = \frac{p(\mathbf{x}_i'|\mathbf{x}_{-i}')p(\mathbf{x}_{-i}')}{p(\mathbf{x}_i^{(t)}|\mathbf{x}_{-i}^{(t)})p(\mathbf{x}_{-i}^{(t)})}\frac{p(\mathbf{x}_i^{(t)}|\mathbf{x}_{-i}')}{p(\mathbf{x}_i|\mathbf{x}_{-i}^{(t)})} = \frac{p(\mathbf{x}_i'|\mathbf{x}_{-i}')p(\mathbf{x}_{-i}')}{p(\mathbf{x}_i^{(t)}|\mathbf{x}_{-i}')p(\mathbf{x}_{-i}')}\frac{p(\mathbf{x}_i^{(t)}|\mathbf{x}_{-i}')}{p(\mathbf{x}_i|z_{-i}')} = 1$$

the acceptance rate is always 1 and Gibbs sampling performs a random walk where at each iteration the value along a randomly selected dimension is updated according to the conditional distribution (geometrically, this constitutes axis aligned transitions). The detailed balance property holds since Gibbs is a special case of Metropolis and T is ergodic if all dimensions are updated with positive probability.

[1] See for example Numerical Recipes in C, `http://www.nrbook.com/a/bookcpdf/c7-2.pdf`

Gibbs sampling is useful when sampling from $p(\mathbf{x}_i|\mathbf{x}_{-i}^{(t)})$ is easy and quick. In these cases, each random walk iteration is quick and all proposed values are accepted. Examples for such models are Bayesian networks or other models that are specified as a product of conditional distributions.

2.4 Langevin Monte Carlo

Metropolis-Hastings and Gibbs sampling operate only by evaluating $p(\mathbf{x})$ (and $q(x)$). It does not leverage any geometric information (for example, gradient or Hessian) about the density that you are sampling from. As a result of this, the so-called *burn-in* period of these algorithms is long, i.e., you need to take a large number of steps before actually getting samples from the target density. (In most cases, the dependence on dimensionality will be a high-degree polynomial in d). Langevin Monte Carlo is an approach that uses gradient information of the target density that we are sampling from, to obtain a shorter burn-in period (typically, linear in the dimensionality d) for a class of densities. The algorithm also has a similar flavor to the gradient descent algorithm. Before proceeding, we discuss some notation.

Recall that for a function $p(\mathbf{x})$ to be a density, it has to satisfy: (i) $p(\mathbf{x}) \geq 0$ and (ii) $\int p(\mathbf{x})\, d\mathbf{x} = 1$. For a function $f(\mathbf{x})$, if you consider $p(\mathbf{x}) \propto e^{-f(\mathbf{x})}$, then the first condition is automatically satisfied. To make it satisfy the second condition too, we just calculate the normalization constant:

$$p(\mathbf{x}) = \frac{e^{-f(\mathbf{x})}}{Z_p},$$

where $Z_p = \int e^{-f(\mathbf{x})}\, d\mathbf{x}$. Now, we can assume that the function $f(\mathbf{x})$ is convex. This in turn would imply that $-f(\mathbf{x})$ is concave. In terms of $p(\mathbf{x})$, we have $\log p(\mathbf{x}) = -f(\mathbf{x}) + C$ and so is concave. Such class of densities are called as log-concave densities. In fact, a standard example of the above form is the Gaussian density, where $f(\mathbf{x}) = (\mathbf{x} - \mu)^\top \Sigma^{-1}(\mathbf{x} - \mu)$. Langevin Monte Carlo is ideally suited for sampling from such log-concave densities.

The iterates of the Langevin Monte Carlo algorithm is given by

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta_t \nabla f(\mathbf{x}^{(t)}) + \sqrt{2\eta_t}\ \epsilon^{(t+1)}$$

where $\epsilon^{(t)} \sim N(0, \mathbf{I}_d)$ are i.i.d. samples from standard Gaussian random variable, for all t. The main thing to note here is that there is an η_t terms for the gradient and $\sqrt{\eta_t}$ term for the Gaussian vector term. One can show that by selecting η_t appropriately, the burn-in period of this approach could be made linear in dimensionality d. Also, note that we need to calculate and evaluate the gradient of $f(\mathbf{x})$ for this method. In comparison for Metropolis-Hastings and Gibbs sampling, we need only function evaluations and did not require any gradient evaluation. But we know from optimization, a way of estimating the gradients based on function evaluations.

Some comments on sampling: In one-dimensional setting (or low-dimensional (< 30) settings), sampling is practical. For the high-dimensional setting, we need more structure on $p(\mathbf{x})$. One such assumption is $p(\mathbf{x})$ is log-concave in which case, Langevin Monte Carlo works really well. For general cases of $p(\mathbf{x})$, still we can run any of the above methods, but the burn-in period might be a high-degree polynomial in d. In some sense, log-concave class of densities for the sampling problem plays the rule of convex functions for the optimization problems. Both of the cases are what we can call as *success stories* for sampling and optimization, respectively as we can give precise guarantees on several different algorithms in this setting. When deviating away from such assumptions, we just try out several heuristics and hacks to make things work, as much as possible.