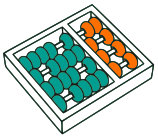

Monte Carlo inference

Lesson No. 11

Patrick de Carvalho Tavares Rezende Ferreira - 175480

1 Abstract

Deterministic algorithms for posterior inference make use of many benefits of the Bayesian approach and still are as fast as optimization-based point-estimation methods. But there is an issue with these methods, they can be rather complicated to derive and limited in their domain of applicability. And although they are fast, their accuracy is limited by the choosen form of approximation.

There is an alternative class of algorithims based on the ideia of Monte Carlo approximation with a simple idea behind: Generate some samples from the posterior, $x^s \sim p(\mathbf{x}|D)$, and then compute any quantity of interest, which may be the posterior marginal, $p(x_1|D)$, posterior of differences, $p(x_1 - x_2|D)$, the posterior predictive, $p(y|D)$, etc. These quantities can be approximated by $E[f|D] \approx \frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x}^s)$, for some "$f$" suitable function.

2 Standard distributions

In this sections we are going to see examples of how to sample 1 or 2 dimension distributions and how to apply these methods in a generalistic way. Inverse probability is the simplest method to base on while sampling an univariate distribution. Given a F cdf where we sample from, F^{-1} is the inverse. Thus, if $U \sim U(0,1)$ is a uniform random varibale (rv), then $F^{-1}(U) \sim F$.

We can sample from any univariate distribution if we can evaluate its inverse cdf [1]. We have an example of this over an exponential distribution in equation 1.

$$Expon(x|\lambda) \stackrel{\Delta}{=} \lambda e^{-\lambda x} \mathbb{I}(x \geq 0) \tag{1}$$

For equation 1, the cdf is shown in equation 2.

$$F(x) = 1 - e^{-\lambda x} \mathbb{I}(x \geq 0) \tag{2}$$

The inverse cdf for exponential cdf, therefore, is shown in equation 3.

$$F^{-1}(p) = -\frac{ln(1-p))}{\lambda} \tag{3}$$

3 Rejection sampling

There are some cases when we cannot use inverse cdf method, having one alternative that is to use rejection sampling. We create a proposal distribution $q(x)$ which satisfies $Mq(x) \geq \tilde{p}(x)$, for some M constant, where $\tilde{p}(x)$ is an unnormalized version of p(x). The function $Mq(x)$ provides an upper envelope for $\tilde{p}(x)$. Then, sample $x \sim q(x)$, which corresponds to picking a random x location and then we sample $u \sim U(0,1)$, which corresponds to picking a random height in y, under the envelope. If $u > \frac{\tilde{p}}{Mq(x)}$, we reject the sample, else we accept it, only this decision.

Let we have the equations 4 and 5 below in order to prof it.

$$S = \{(x,u) : u \leq \tilde{p}(x)/Mq(x)\} \tag{4}$$

$$S_0 = \{(x,u) : x \leq x_0, u \leq \tilde{p}(x)/Mq(x)\} \tag{5}$$

Then the cdf of the accepted points is given by [6,7,8,9], which is the desired cdf of p(x).

$$P(x \leq x_0 | x\ accepted) = \tag{6}$$

$$\frac{P(x \leq x_0, x\ accepted)}{P(x\ accepted)} = \tag{7}$$

$$\frac{\int\int \mathbb{I}((x,u) \in S_0) q(x) du dx}{\int\int \mathbb{I}((x,u) \in S) q(x) du dx} = \tag{8}$$

$$\frac{\int_{-\infty}^{x_0} \tilde{p}(x) dx)}{\int_{-\infty}^{\infty} \tilde{p}(x) dx} \tag{9}$$

The probability of acceptance is 10, what give us that M must be as small as possible while still satisfying $Mq(x) \geq \tilde{p}(x)$.

$$p(accept) = \int \frac{\tilde{p}(x)}{Mq(x)} q(x) dx = \frac{1}{M} \int \tilde{p}(x) dx \tag{10}$$

3.1 Bayesian statistics

If we want to draw unweigthed samples from the posterior - $p(\theta|D) = p(D|\theta)p(\theta)/p(D)$ - we can use rejection sampling with $\tilde{p}(\theta) = p(D|\theta)P(\theta)$ as target distribution, $q(\theta) = p(\theta)$ as our proposal, and $M = p(D|\hat{\theta})$, given $\hat{\theta} = argmax p(D|\theta)$ is the MLE. Points are accepted with probability given in 11.

$$\frac{\tilde{p}(\theta)}{Mq(\theta)} = \frac{p(D|\theta)}{p(D|\tilde{\theta})} \tag{11}$$

Prior samples with high likelihood are more likely to be retained in the posterior. Of course, if there is a big mismatch between prior and posterior, this procedure is very inneficient.

4 Importance Sampling

The description of Monte Carlo method known as importance sampling is shown in 12.

$$I = E[f] = \int f(x) p(x) dx \tag{12}$$

The concept of this tecnique is to draw samples x in regions which have hig probability, p(x), but also where |f(x)| is large. We get to efficient results if compared to when we were sampling from the exact distribution p(x). The reason is that the samples are focussed on the important parts of space. For example, suppose we want to estimate the probability of a rare event. Define $f(x) = \mathbb{I}(\mathbf{x} \in E)$, for some set E. Then it is better to sample from a proposal of the form $q(x) \propto f(x)p(x)$ than to sample from p(x) itself. Importance sampling samples from any proposal, q(x). It uses these samples to estimate the integral as 13.

$$E[f] = \int f(x) \frac{p(x)}{q(x)} q(x) dx \approx \frac{1}{S} \sum_{s=1}^{S} w_s f(x^s) = \hat{I}, \tag{13}$$

given that $w_s \stackrel{\Delta}{=} \frac{p(x^s)}{q(x^s)}$ are the importance weights. Unlike rejection sampling, we use all samples here.

A criterion to choose the proposal is to minimize the variance of the estimate $\hat{I} = \sum_s w_s f(x_s)$. Therefore, we have 14.

$$var_{q(x)}[f(x)w(x)] = E_{q(x)}[f^2(x)w^2(x)] - I^2 \tag{14}$$

We can ignore the last term, as it's independent of q. Then we have 15 as lower bound.

$$E_{q(x)}[f^2(x)w^2(x)] \geq (E_{q(x)}[|f(x)w(x)|]^2) = \left(\int |f(x)| p(x) dx\right)^2 \tag{15}$$

We obtain the lower bound using the optimal importancedistribution 16.

$$q^*(x) = \frac{|f(x)|p(x)}{\int |f(x')|p(x')dx'} \tag{16}$$

Having no particular target function f(x) in mind, we often just try to make q(x) as close as possible to p(x). This is difficult, but it's possible to adapt the proposal distribution to improve the approximation. This is adaptative importance sampling.

4.1 Unnormalized distributions

Sometimes there is a difficulty which is that we frequently can evaluate the unnormalized target distribution, $\tilde{p}(x)$, but not its normalization constant, Z_p. Sometimes we may want to use either an unnormalized proposal, $\tilde{q}(x)$, with possibly unknown normalization constant Z_q. We can do this with same steps. First, we compute 17.

$$E[f] = \frac{Z_q}{Z_p}\int f(x)\frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)dx \approx \frac{Z_q * 1}{Z_p * S}\sum_{s=1}^{S}\bar{w}_s f(x^s), \tag{17}$$

for $\bar{w}_s \triangleq \frac{\tilde{p}^s}{\tilde{q}^s}$ being the unnormalized importance weight. We may use the same set of samples to evaluate the ratio Z_p/Z_q as in equation 18.

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(x) = \int \frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)dx \approx \frac{1}{S}\sum_{s=1}^{S}\bar{w}_s \tag{18}$$

Thus, we have 19.

$$\hat{I} = \frac{\frac{1}{S}\sum_{s=1}^{S}\bar{w}_s f(x^s)}{\frac{1}{S}\sum_{s=1}^{S}\bar{w}_s} = \sum_{s=1}^{S}\bar{w}_s f(x^s), \tag{19}$$

for $w_s \triangleq \frac{\bar{w}_s}{\sum_{s'}\bar{w}_{s'}}$ being the normalized importance weights. Subsequent estimation is a ratio of two estimates, and biased. However, as $S \to \infty$, we have that $\hat{I} \to I$, under some brief assumptions.

5 Particle Filtering

We can apply Monte Carlo algorithm for recursive Bayesian inference, what e call Particle Filtering (PF). Many areas use this method, and here we explain the basic algorithm. The idea behind is to approximate the belief state using a weighted set of particles, as in 20.

$$p(z_{1:t}|y_{1:t}) \approx \sum_{s=1}^{S}\hat{w}_t^s \delta_{z_{1:t}^z}(Z_{1:t}), \tag{20}$$

for $\hat{w}_t^s$ being the normalized weight sample s at time t. With this representation, we can compute the marginal distribution over the most recent state, $p(Z_t|y_{1:t})$, by ignoring the previous parts of the trajectory, $z_{1:t}$.

This belief is updated using through importance sampling. For a proposal with form $q(z_{1:t}^s|y_{1:t})$, the importance weights are given by 21.

$$w_t^s \propto \frac{p(z_{1:t}^s|y_{1:t})}{q(z_{1:t}^s|y_{1:t})}, \tag{21}$$

that, when normalizing, gives 22.

$$\hat{w}_t^s = \frac{w_t^s}{\sum_{s'} w_t^s} \tag{22}$$

Just rewriting numerator gives us 23.

$$p(z_{1:t}|Y_{1:t}) = \frac{p(y_t|z_{1:t}, y_{1:t-1})p(z_{1:t}|y_{1:t-1})}{p(y_t|Y_{1:t-1})} = \frac{p(y_t|z_t)p(z_t|z_{1:t-1}, y_{1:t-1})p(z_{1:t-1}|y_{1:t-1})}{p(y_t|Y_{1:t-1})} \propto p(y_t|z_t)p(z_t|z_{t-1})p(z_{1:t-1}|y_{1:t-1}) \tag{23}$$

We focus on proposal densities as 24.

$$q(z_{1:t}|y_{1:t}) = q(z_t|z_{1:t}, y_{1:t})q(z_{1:t-1}|y_{1:t-1}) \tag{24}$$

Then we can keep the trjectory adding new state z_t to the end. In this case, importance weights gets the form of 25.

$$w_t^s \propto \frac{p(y_t|z_t^s)p(z_t^s|z_{t-1}^s)p(z_{1:t-1}^s|y_{1:t-1})}{q(z_t^s|z_{1:t-1}^s, y_{1:t})q(z_{1:t-1}^s|y_{1:t-1})} = w_{t-1}^s \frac{p(y_t|z_t^s)p(z_t^s|z_{t-1}^s)}{q(z_t^s|z_{1:t-1}^s, y_{1:t})} \tag{25}$$

In cases we can make more assumptions, such as $q(z_t|z_{1:t-1}, y_{1:t}) = q(z_t|z_{t-1}, y_t)$, we only need to keep most recent trajectory part and observation sequence. Now, weight becames 26.

$$w_t^s \propto w_{t-1}^s \frac{p(y_t|z_t^s)p(z_t^s|z_{t-1}^s)}{q(z_t^s|z_{t-1}^s, y_t)} \tag{26}$$

The approximated posterior becomes 27.

$$p(z_t|y_{1:t}) \approx \sum_{s=1}^{S} \hat{w}_t^s \delta_{z_t^s}(z_t) \tag{27}$$

Finally, the whole process is: For each old sample s, we propose an extension by $z_t^s \sim q(z_t|z_{t-1}^s, y_t)$, and calculate new w_t^s from 26.

References

[1] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.