Sampling I: Inference Control and Driving of Natural Systems

MSci/MSc

Nick Jones

nick.jones@imperial.ac.uk

Sampling I

What we have covered so far.

In the next two lectures we will be providing an explanation of sampling techniques. These will provide the ground-work for the practical lectures where we will investigate mechanisms for sampling considered in a neural setting.
What is Bayesian Cognitive Science?
Written for a popular audience:
[1] Statistics and the Bayesian Mind (2006) - Significance.
Written for a general scientific audience:
[2] How to Grow a Mind: Statistics, Structure, and Abstraction (2011) - Science.
Or longer: A tutorial introduction to Bayesian models of cognitive development by Perfors et al.
An example paper:
[3] Pure Reasoning in 12-Month-Old Infants as Probabilistic Inference (2011) - Science.

Implementing Bayesian Inference

Posterior is: $P(\theta|D)$. We need to be able to sample from it in order to calculate functions like means.

Why is it hard to sample from $P(\mathbf{x})$? I

It is often hard to normalize $P(\mathbf{x})$.
Consider $P^*(\mathbf{x})$ which is un-normalized.
Suppose that it is easy to evaluate $P^*(\mathbf{x})$ for any $\mathbf{x}$.
Imagine that I evaluate $P^*(\mathbf{x})$ at a particular point $\mathbf{x}$ and it is very large. What does this mean? Not much until I normalize.
If we consider the setting where $\mathbf{x}$ is N-dimensional. Imagine that we discretize and evaluate $P^*(\mathbf{x})$ at S points per dimension. It's clear that in order to normalize this discretized distribution we'd need S^N function evaluations. Bad news if S or N are moderate sized.
See [4]. This is why the crude strategy I introduced in my first lecture isn't such a great idea.

Why is it hard to sample from $P(\mathbf{x})$? II

What about, instead, sampling $P^*(\mathbf{x})$ at a R points by picking $\mathbf{x}$ uniformly? A draw might be $\mathbf{x_r}$.

This helps calculate integrals of the form $F = \int f(\mathbf{x})P(\mathbf{x})d\mathbf{x}$ (normalize by summing the values of $P^*(\mathbf{x})$ at each of the sampled points).

$F \simeq \sum_{r=1}^{R} f(\mathbf{x_r})P^*(\mathbf{x_r})/Z_R$ with $Z_R = \sum_{r=1}^{R} P^*(\mathbf{x_r})$

The issue with this strategy is that, for high dimensional functions $P^*(\mathbf{x})$, it is very likely that we will get very poor sampling of the underlying distribution (just as this strategy would give me a poor representation of the average depth of a 10km by 10km region of ocean including a small, 1m by 1m, but terrifically deep borehole: a random sampling of depths will miss the hole so my approximate mean will thus be very inaccurate).
See [4]

Why is it hard to sample from $P(\mathbf{x})$? Solutions I

There exist a variety of methods to handle the problem of using $P^*(\mathbf{x})$ to make draws from $P(\mathbf{x})$ and then to calculate expectations.

Importance and rejection sampling: guess in advance where your distribution is localized and leverage that. This (implicitly) deals with the issue we identified with uniform sampling from $P^*(\mathbf{x})$. However you still need to be very accurate in high dimensions [4]. Importance sampling works for calculating expectations of functions. Rejection sampling allows you to sample from $P(\mathbf{x})$. See [4] [5]

Why is it hard to sample from $P(\mathbf{x})$? Solutions II

There exist a variety of methods to handle the problem of using $P^*(\mathbf{x})$ to make draws from $P(\mathbf{x})$ and then to calculate expectations.

Markov Chain Monte Carlo: walk around $P^*(\mathbf{x})$ in a manner which seeks out the larger values: a mix of search and sampling. Obviously this has the advantage that you don't need to specify in advance where the distribution is likely to be localized. MCMC exploits relative probabilities - or effectively - a local probability estimation rather than global probability estimation - it looks at the ratio of the unnormalized distribution at two points, $\mathbf{x}$, $P^*(\mathbf{x})$ and, $\mathbf{x}'$, $P^*(\mathbf{x}')$. This is, of course, the same as $P(\mathbf{x})$ and $P(\mathbf{x}')$. So we can make judgements about ratios without worrying about normalization.
See [4] [5].

Sampling and the brain

Authors have investigated how sampling strategies could be implemented by neural populations. The basic logic is that if the brain performs Bayesian inference then in order to deal with the tricky integrals you need to sample. We will discuss this in more detail next time.

Gibbs sampling [6] - Interpreting Neural Response Variability as Monte Carlo Sampling of the Posterior

Importance sampling [7]- Neural Implementation of Hierarchical Bayesian Inference by Importance Sampling

Particle filters and message passing [8] (we won't cover these).

Rejection Sampling

We would like to draw from $P(x)$. I'll draw on the board two distributions $P^*(x)$ and $cQ(x)$ where we know that we have a c such that $cQ(x) > P^*(x) \forall x$.

As well as assuming we know c we also suppose that it is easy to sample from $Q(x)$. $Q(x)$ is called the proposal distribution.
Make two draws

- draw x_0 from $Q(x)$.
- draw a uniform number, u_0, on the range $[0, cQ^*(x_0)]$

If $u_0 > P^*(x_0)$ reject the sample and draw another.
Otherwise accept the same as a true draw from $P(x_0)$.
Research how rejection sampling performs for higher dimensional problems and how the number of rejections depends on Q, P^* and c.

Reminder: Finding Expectations (when it is easy to sample)

Suppose I want to find the expected value of $f(x)$ with X distributed as $Q(x)$ and when it easy to draw from the normalized distribution $Q(x)$: $F = \int f(x)Q(x)dx$. We consider samples $\{x_1, ...x_N\}$ drawn from $Q(x)$ and evaluate $F = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$ (obviously N needs to be chosen carefully).

Importance Sampling

Importance sampling is a sampling tool to approximate expectations: $F = \int f(x)P(x)dx$. It is not a method to sample directly from distributions (like Rejection Sampling).

I'll draw on the board two distributions $Q(x)$ (from which it is easy to sample) and un-normalized $P^*(x)$. The basic importance sampling strategy of re-weighting draws from $Q(x)$ can be explained through a drawing.

We can rewrite

$F = \frac{1}{Z_p}\int f(x)P^*(x)dx = \frac{1}{Z_p}\int f(x)[P^*(x)/Q(x)]Q(x)dx$.

Where Z_p is the normalizer for $P^*(x)$ (hard to calculate and - non-examinable - related to the Partition Function in Statistical physics).

Importance Sampling II

$F = \frac{1}{Z_p} \int f(x)P^*(x)dx = \frac{1}{Z_p} \int f(x)[P^*(x)/Q(x)]Q(x)dx$
We make a set of draws $\{x_1, ...x_N\}$ from $Q(x)$ and consider
$w_i = [P^*(x_i)/Q(x_i)]$.
$Z_p = \int [P^*(x)/Q(x)]Q(x)dx = \frac{1}{N} \sum_{i=1}^{N} w_i$.
It follows that $F = \frac{1}{Z_p} \int f(x)P^*(x)dx =$
$\frac{1}{Z_p} \int f(x)[P^*(x)/Q(x)]Q(x)dx = \frac{1}{NZ_p} \sum_{i=1}^{N} f(x_i)w_i$
or $F = \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} f(x_i)w_i$
Research the shortcomings of Importance Sampling.

What we've covered

The need for sampling

Rejection sampling

Importance sampling

Next time: Markov Chain Monte Carlo

[1] T. Griffiths and J. Tenenbaum, Statistics and the Bayesian Mind, Significance, 2006

[2] J. Tenenbaum et al. How to Grow a Mind: Statistics, Structure, and Abstraction, Science, 2011

[3] E. Teglas et al. Pure Reasoning in 12-Month-Old Infants as Probabilistic Inference, Science, 2011

[4] D. J. C. MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003.

[5] A. Gelman et al. Bayesian Data Analysis 2nd ed. Chapman and Hall, 2003.

[6] P. O. Hoyer and A. Hyvarinen. Interpreting neural response variability as Monte Carlo sampling from the posterior. In Adv. Neur. Inf. Proc. Syst. 16, 293, 2003.

[7] L. Shi and T.L. Griffiths, Neural Implementation of Hierarchical Bayesian Inference by Importance Sampling, NIPS, 2009.

[8] T. S. Lee and D. Mumford. Hierarchical Bayesian inference in the visual cortex. Journal of the Optical Society of America A, 20, 1434, 2003.