Inference via Sampling

Ron Parr
CPS 271

Why Sampling?

- Very often we want to compute an expectation of some kind:

$$E[f] = \int f(z)p(z)dz$$

- Problem: This integral may not be easy to solve, in general.
- Sampling gives us an "anytime" method of evaluating this integral

How to do sampling

- Basic idea: Replace integral over domain range of z:

$$E[f] = \int f(z)p(z)dz$$

- With a finite sum over samples drawn from p(z), $z^{(1)} \ldots z^{(L)}$:

$$\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(z^{(l)})$$

- Should match your intuitive solution of running a lot of simulations and "taking the average"

Challenges with Sampling

- How do we generate samples from p(z) efficiently?
- How many samples do we need?
- Are all samples equally good?
- Examples:
  - Estimating the expected value of a die roll
  - Estimating the expected payoff from a lottery

Sampling from the Right Distribution

- Most programming environments can provide samples that are "uniform" and pseudo-random on [0,1]
- For most purposes pseudo-random is good enough
- Problem: Uniform is not good enough!
- If all we have is samples from the uniform distribution, how do we generate samples from the desired distribution?

Tricks for Sampling

- Use inverse of the cumulative distribution
- Rejection
- Importance weights
- Markov chain Monte Carlo

Inverse Cumulative Distribution

- Suppose that z is uniformly distributed on [0,1]
- Goal: Generate samples from some target distribution p(y)
- Compute:

$$h(y) = \int_{-\infty}^{y} p(x)dx$$

- Return:

$$y = h^{-1}(z)$$

Why this works – in English

- h(y) tells us how much probability mass is accumulated from up to value y
- 0<=h(y)<=1
- h-1(z) tells us the exact y for which the probability of sampling something smaller is z, and the probability of sampling something larger is 1-z… when sampling from p

Examples

- A random selection from a deck of cards
- Sampling from a biased coin
- Sampling from a loaded die
- Sampling form uniform on [a,b]
- Sampling from p(y)=ae^{-ay}

Complications

- The integral may not be easy (e.g. Gaussian):

$$h(y) = \int_{-\infty}^{y} p(x)dx$$

- The inverse may not be easy:

$$y = h^{-1}(z)$$

Rejection

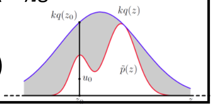
- Idea: Sometimes it is easy to sample from a distribution over a larger domain than the target distribution
  - Sample from the “easier” distribution
  - Reject samples that fall outside of the desired range
- Example: Suppose you want to sample uniformly from the unit circle?
- Solution:
  - Sample uniformly from the unit square
  - Test if the sample is in the unit circle, and reject if not

General Rejection Issues

- Q: Why is rejection valid?
- A: Uniform over a large area implies uniform over a sub-area
- Q: How does rejection scale?
- A: It can scale poorly (many rejected samples for each kept sample) if the target area is smaller than the sampled area
- How does the unit hypersphere scale with the unit hypercube?

Fancier Rejection Sampling

- Suppose we can compute p(z), but it is hard to sample from p(z)
- Suppose it is easy to sample from q(z), and k*q(z)>p(z) (always possible)
- Generate two samples:
  - Sample z0 from q(z)
  - Sample u0 uniformly from [0,k*q(z0)]g
- Accept if u0<p(z0)
- (fancier version of sphere/cube)



©Chris Bishop

Thoughts about Rejection Sampling

- Rejection sampling often viewed as something to avoid b/c it can lead to a lot of wasted calls to the random number generator
- Efficiency of rejection sampling depends critically on how close q is to p
- Adaptive rejection sampling tries to improve on q during execution to make it a better match to p

Importance Sampling

- Recall that we wish to estimate:

$$E[f] = \int f(z)p(z)dz$$

- Using samples $z^{(1)} \ldots z^{(L)}$ drawn from p(z):

$$\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(z^{(l)})$$

- We again consider what to do if it is hard to sample from p(z), but easy to evaluate p(z)

Proposal Distributions

- Suppose it is easier to sample from $z^{(1)} \ldots z^{(L)}$ from q(z) instead of p(z)
- Trivial substitution:

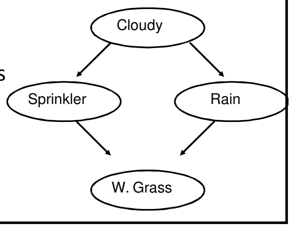
$$E[f] = \int f(z)\frac{p(z)}{q(z)}q(z)dz$$

- Approximation as a sum (remembering that our samples are drawn from q this time):

$$\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(z^{(l)})\frac{p(z^{(l)})}{q(z^{(l)})}$$

Examples of Importance Sampling

- Computing conditional probabilities in Bayesian networks
- What happens when reality doesn't match our samples?
  - Reject
  - Use importance weights
- Example: P(c|w)



Thoughts on Importance Sampling

- In theory, any distribution q that assigns positive probability to all events will work
- When p(z)/q(z) is low, our estimates will be unreliable and we will need may samples to compensate
- As with rejection sampling, when p(z) is close to q(z), we will be happy
- Importance weights can sometimes have the counterintuitive feel of using data that contradict reality

Markov Chains

- Consider a time varying sequence of states
- Let S_t be a random variable for the state at time t
- $P(S_t|S_{t-1},\ldots,S_0) = P(S_t|S_{t-1})$
- (Use subscripts for time; S0 is different from S_0)
- Markov is special kind of conditional independence
- Future is independent of past given current state

Properties of Markov Chains

- Markov chains are widely used to study the properties of dynamic systems
- Will be covered in more detail later in the course
- Some Markov chains are ergodic:
  - Converge to a stationary (equilibrium distribution)
  - Stationary distribution does not depend upon initial state
- A non-ergodic system could be periodic (oscillated between different modes)

Markov Chain Monte Carlo (MCMC)

- Suppose we want to sample from some distribution p(z), but that it’s hard to do this
- As usual, assume p(z) is easy to evaluate
- Also, assume we have another distribution q(z*|z) that is easy to compute
- Idea: Use q to create a Markov chain with a stationary distribution of p

Implementing MCMC

- Suppose we have currently sampled state z
- Generate new state z* from q(z*|z)
- Accept z* with probability:

$$\min(1, \frac{p(z^*)q(z \mid z^*)}{p(z)q(z^* \mid z)})$$

- Otherwise, we assume we stay in state z for one step, and then try again with a new z*

Properties of MCMC

- Not obvious, but can show that for any reasonable q, the resulting Markov chain has p as a stationary distribution of p
- Called the Metropolis-Hastings algorithms
- OK, but how do you use this?
  - Run MCMC for “a while”
  - Pick z as sample from p
  - Repeat
- Why do we need to run for “a while”
  - Want independent samples from p
  - If we draw the next sample immediately, then the samples aren’t independent any more
- How long is “a while”?
  - Depends upon the “mixing rate” of the Markov chain
  - Mixing rate depends upon the eigenvalues of the transition matrix (will make more sense after we cover Markov chains explicitly)
  - Often difficult to determine this precisely

Thoughts on MCMC

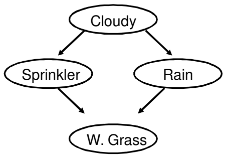
- Often used in a way that is not theoretically sound
  - No effort may to estimate the mixing rate
  - Samples drawn sequentially (perhaps OK if you draw enough of them)
  - Seem to work reasonably well in some cases despite this
- As with other techniques, performance depends heavily upon how well q matches p

Gibbs Sampling

- Suppose we have a distribution where it is (relatively) easy to compute conditional probabilities of a subset of the variables given the rest
- Gibbs sampling for variables $x_1 \ldots x_n$:
    - For i=1 to n
        - Sample x_i given $x_1 \ldots x_{i-1} x_{i+1} \ldots x_n$
- Can be interpreted as special case of MCMC

Gibbs Sampling for Graphical Models

- Gibbs sampling is almost perfectly suited for graphical models
- Why: Easy to compute conditional distribution of one variable given its Markov blanket
- Esp. easy for MRFs



Gibbs Sampling Comments

- Gibbs sampling is a very simple and intuitive algorithm for distributions that factor (i.e. ones for which there are nice graphical models)
- Gibbs sampling can be very slow in practice (requires many, many samples to produce reasonable answers)
- Why: There is no q (no user input), which means that there is no way for the user to push the solution in the right direction

Sampling Conclusions

- Sampling methods are a powerful bridge between simulation and inference
- Effective use of sampling often requires clever choice of a proposal distribution q that matches the target distribution, p, but is easier to deal with
- Some challenges in the effective use of sampling:
    - Picking a good q
    - Understanding/managing the convergence rate