Computational Sampling: Why, God, Why?

Friday, September 07, 2012
2:13 PM

- How to generate samples from an arbitrary probability distribution is a significant concern for those who want to use Bayesian methods
- If we want to summarize the posterior density of some relevant quantity, we have to be able to show and work with it
- This is easy if the posterior is an analytically closed-form density… but many Bayesian posteriors are not
    - Significant advantage of Bayesian methods: very complex models can be specified
    - ...but not as a simple likelihood with a conjugate prior
- We need to find a way to get information about the shape of posterior densities for which we can't write down a direct function

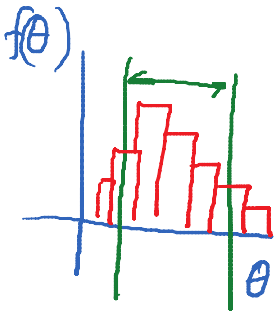
Monte Carlo Integration

Friday, September 07, 2012
5:02 PM

- Start with a more basic idea: how do we determine:

  $$\int_{\theta_0}^{\theta_1} f(\theta) d\theta$$

  the area under a density inside of an interval $[\theta_0, \theta_1]$ if we can't analytically solve the integral (i.e., the cumulative density)?



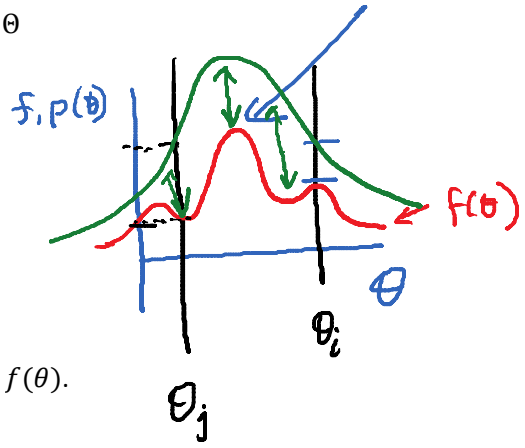
- One idea: Monte Carlo integration
  - Draw a lot of samples of θ from $f(\theta)$
  - Add up the number of samples that are inside of $[\theta_0, \theta_1]$ and divide by the total number of samples
- So… how do we sample θ from $f(\theta)$?
- This is the subject of a massive amount of study. We will cover just a few simple methods.

The Accept-Reject Sampling Algorithm

Friday, September 07, 2012
5:06 PM

$\alpha \geq 1$

1) Create a proposal density, $p(\theta)$; this should be analytically closed (and therefore $\neq f(\theta)$ in most cases).
1) Create an envelope function, $e(\theta) = \alpha p(\theta)$, such that $e(\theta) > f(\theta) \forall \theta \in \Theta$

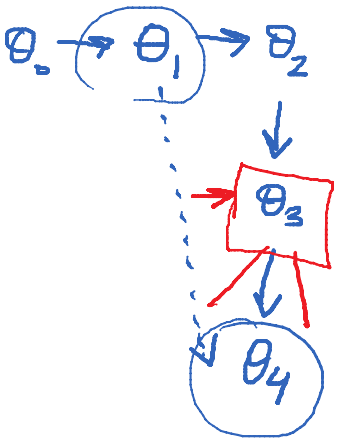


2) For $i = 1 \ldots n$:
   a. Draw a candidate $\theta_i \sim p(\theta)$.
   b. Draw u_i from $U[0,1]$.
   c. If $u_i < \frac{f(\theta_i)}{e(\theta_i)}$, save θ_i. If not, reject θ_i.
3) As $n \to \infty$, the collection of accepted θ_i values will follow the distribution $f(\theta)$.

- In brief, this works because the probability that θ_i is accepted is proportional to $f(\theta_i)$: the higher the density, the more likely that this draw will be accepted.
- If θ is multidimensional, choose a multidimensional $p(\theta)$ and draw candidates from every dimension at the same time

Metropolis-Hastings Algorithm

Friday, September 07, 2012
5:36 PM



- The Metropolis-Hastings algorithm is an extension of the principles behind accept-reject sampling
- Construct a chain of samples of θ_t for $t = 1 \ldots n$, where the value of θ_t is dependent on θ_{t-1}
- This is called a *Markov chain*, a sequence of values where the probability of the present value's appearance is (solely) a function of the previous value
- The Metropolis-Hastings algorithm applies ideas from Accept-Reject sampling to generate a Markov chain
- Under appropriate conditions (TBD), this Markov chain will have a distribution equal to the target distribution $f(\theta)$ as the number of samples gets large

The Metropolis-Hastings Algorithm:

1. Create a proposal density, $g(\theta_t)$, for the next value in the chain; this should be analytically closed (and therefore $\neq f(\theta)$ in most cases).
    a. The proposal density for θ_t will generally depend on θ_{t-1}, $g(\theta_t|\theta_{t-1})$.
2. Select an initial value for θ; set θ_1 equal to this value.
3. For n iterations of the algorithm t:
    a. Propose a candidate, $\theta_t^c \sim g(\theta_t|\theta_{t-1})$.
    b. Compute the Metropolis-Hastings ratio:

$$R = \frac{f(\theta_t^c)g(\theta_{t-1}|\theta_t^c)}{f(\theta_{t-1})g(\theta_t^c|\theta_{t-1})}$$

    c. Draw $u \sim U[0,1]$.
    d. If $u < R$, set $\theta_t = \theta_t^c$. Otherwise, set $\theta_t = \theta_{t-1}$.

- The logic of the M-H algorithm: get the target/proposal density ratio of the proposed member of the chain, and accept with greater likelihood as this ratio increases relative to the ratio of the previous member

Markov Chain: Background

Friday, September 07, 2012
5:54 PM

- Big question: is the Markov chain that results from any algorithm a good approximation of the target density?
- We need a bit of Markov chain theory and definitions to describe what's going on here
- Reminder: a Markov chain is a series of random variables unfolding over time, where the realization of the variable at time *t* depends solely on the state of the variable at time *t-1*
- When the state space (the set of possible realizations of the variable) is discrete and has k states, the chance of changing from state *i* to state *j* is given by a **transition matrix**
  - If the (k x k) transition matrix is called *P* and the marginal probability of the state at time *t* is a (k x 1) vector π, the distribution of the state at time *t+1* is $\pi' P$
  - If $\pi' P = \pi'$, then π is a stationary distribution of the Markov chain
- When the state space is continuous…
  - If the marginal distribution of the state at time *t* is $f(x_t)$ and the transition probability is $f(x_{t+1}|x_t)$, the distribution of x_{t+1} is $f(x_{t+1}) = \int f(x_t) f(x_{t+1}|x_t) dx_t$
  - If $f(x_{t+1}) = f(x_t)$, then $f(x_t)$ is a stationary distribution of the markov chain
- Ergodic theorem: A sequence of realizations from an *irreducible* and *periodic* Markov chain with stationary distribution π will itself have distribution π as the number of realizations goes to ∞

$\theta \in \{1, 2\}$

$f(\theta)$

$\Pr(\theta=1) \quad \Pr(\theta=2)$

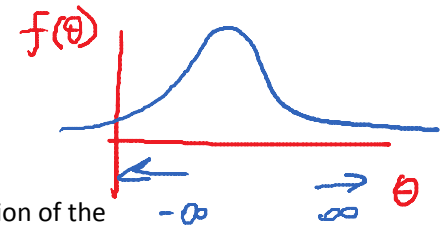
$$P = \begin{matrix} & i & j \\ i & 0.8 & 0.2 \\ j & 0.3 & 0.7 \end{matrix}$$

$$\pi = \begin{matrix} & \\ i & 0.2 \\ j & 0.8 \end{matrix}$$

$$\int f(x_t) \cdot f(x_{t+1}|x_t)\, dx_t \Rightarrow f(x_{t+1})$$

Requirements for our Markov Chain

Sunday, September 09, 2012
1:06 PM

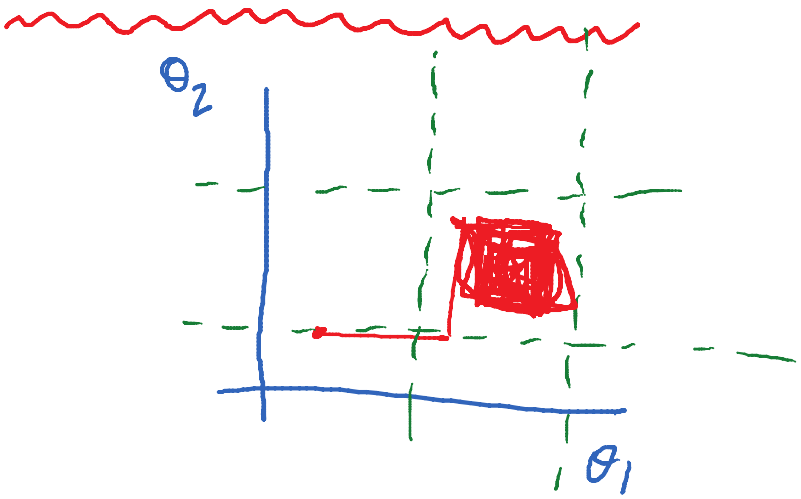


- The consequence: we want our Markov chain to be an *ergodic* representation of the target density

- It needs to be...

1) Recurrent (for all $\theta \in \Theta$ under the target density): the probability of returning to state $\theta_i \in \Theta = 1$
2) Non-null (for all $\theta \in \Theta$ under the target density): the expected time to recurrence is finite
3) Irreducible: the Markov chain can reach any value θ_i from any other state θ_j in a finite number of transitions
4) Aperiodic:
    a. A Markov chain has period k if the probability of going from state j to state j in n steps is 0 for all n not divisible by k
    b. A Markov chain is aperiodic if k=1
5) Have a stationary distribution equal to the target density

ergodic MC ... of target density.



The Validity of Metropolis-Hastings

Sunday, September 09, 2012
1:39 PM

- Consider the joint distribution of the draw at time *t-1* and at time *t*:

$$f(\theta_{t-1}, \theta_t) = f(\theta_{t-1})g(\theta_t|\theta_{t-1})\frac{f(\theta_t)g(\theta_{t-1}|\theta_t)}{f(\theta_{t-1})g(\theta_t|\theta_{t-1})} = f(\theta_{t-1})g(\theta_t|\theta_{t-1})$$

- Now, set $\theta_{t-1} = x_1$ and $\theta_t = x_2$

$$f(x_1, x_2) = f(x_1)g(x_2|x_1)\frac{f(x_2)g(x_1|x_2)}{f(x_1)g(x_2|x_1)} = f(x_2)g(x_1|x_2)$$

- Obviously: $f(\theta_t, \theta_{t+1})$ takes the same form
- Therefore it must be the case that the density of $f(\theta_t)$ is the same as $f(\theta_{t+1})$ (they're the same given any previous value of the chain)
- Integrate out the proposal density:
  $\int f(\theta_{t-1})g(\theta_t|\theta_{t-1})d\theta_{t-1} = f(\theta_{t-1}) = f(\theta_t)$
- So… the stationary distribution of the Metropolis-Hastings algorithm is the target density!

- Still need to check that its resulting density is irreducible and aperiodic
  - Typically true as the number of samples → ∞, but as a practical matter…
  - We have diagnostics for that