Statistical Computing (36-350)

Lecture 16: Simulation II: Markov Chains, Monte Carlo, MCMC

Cosma Shalizi

24 October 2012

Agenda

- Chaining together random variables
  - Natural orderings
  - Markov chains
- Monte Carlo approximation of integrals and expectations
- Markov Chain Monte Carlo

READING: Handouts on the class webpage

Multiple Random Variables

`rnorm`, `runif`, etc., give independent and identically distributed (IID) random variables

Most stochastic models don't call for IID random variables

Varying distributions, dependence

How do we generate such things?

Putting the Variables in Order

Try to arrange the variables in order of priority and/or time

Who someone votes for might change with their age or their race, but not vice versa

Putting the Variables in Order

Try to arrange the variables in order of priority and/or time

Who someone votes for might change with their age or their race, but not vice versa

Generate the **exogenous** variables first

Putting the Variables in Order

Try to arrange the variables in order of priority and/or time

Who someone votes for might change with their age or their race, but not vice versa

Generate the **exogenous** variables first

Then all the **endogenous** variables which only depend on exogenous ones

Putting the Variables in Order

Try to arrange the variables in order of priority and/or time

Who someone votes for might change with their age or their race, but not vice versa

Generate the **exogenous** variables first

Then all the **endogenous** variables which only depend on exogenous ones

Then all the variables which depend only on first-generation endogenous ones, etc.

Putting the Variables in Order

Try to arrange the variables in order of priority and/or time

Who someone votes for might change with their age or their race, but not vice versa

Generate the **exogenous** variables first

Then all the **endogenous** variables which only depend on exogenous ones

Then all the variables which depend only on first-generation endogenous ones, etc.

You'll see more of this with graphical models in 36-402

Time Series

Can have a sequence of variables going on in time, $X_1, X_2, \ldots X_n$
Earlier ones can cause later but not other way

$$p(X_1, X_2, \ldots X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2, X_1)\ldots p(X_n|X_{n-1}, X_{n-2}, \ldots X_1)$$

Markov Processes

The **Markov property:** Given the current **state** X_t, earlier states $X_{t-1}, X_{t-2}, \ldots$ are irrelevant to the future states $X_{t+1}, X_{t+2}, \ldots$

Markov Processes

The **Markov property:** Given the current **state** X_t, earlier states $X_{t-1}, X_{t-2}, \ldots$ are irrelevant to the future states $X_{t+1}, X_{t+2}, \ldots$
$\Leftrightarrow$

$$p(X_1, X_2, \ldots X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2)\ldots p(X_n|X_{n-1})$$

Markov Processes

The **Markov property:** Given the current **state** X_t, earlier states $X_{t-1}, X_{t-2}, \ldots$ are irrelevant to the future states $X_{t+1}, X_{t+2}, \ldots$
$\Leftrightarrow$

$$p(X_1, X_2, \ldots X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2)\ldots p(X_n|X_{n-1})$$

This is an *assumption*, not a law of nature

Markov Processes

The **Markov property:** Given the current **state** X_t, earlier states $X_{t-1}, X_{t-2}, \ldots$ are irrelevant to the future states $X_{t+1}, X_{t+2}, \ldots$
$\Leftrightarrow$

$$p(X_1, X_2, \ldots X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2)\ldots p(X_n|X_{n-1})$$

This is an *assumption*, not a law of nature
To simulate a Markov chain, we need to

- Draw the initial state X_1 from $p(X_1)$
- Draw X_t from $p(X_t|X_{t-1})$ — inherently sequential

Inputs: number of steps, drawing function for initial distribution, drawing function for transition distribution

```
rmarkov <- function(n,rinitial,rtransition) {
  x <- vector(length=n)
  x[1] <- rinitial()
  for (t in 2:n) {
    x[t] <- rtransition(x[t-1])
  }
  return(x)
}
```

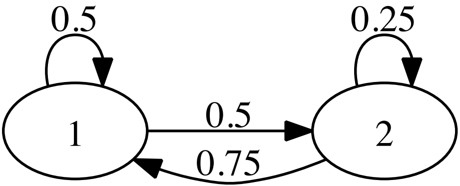
Markov Chains

Each X_t is discrete, not continuous
Represent $p(X_t|X_{t-1})$ in a **transition matrix**,
$\mathbf{q}_{ij} = \Pr(X_t = j | X_{t-1} = i)$
Each row sums to 1 (**stochastic matrix**)

Markov Chains

Each X_t is discrete, not continuous
Represent $p(X_t|X_{t-1})$ in a **transition matrix**,
$\mathbf{q}_{ij} = \Pr(X_t = j|X_{t-1} = i)$
Each row sums to 1 (**stochastic matrix**)
Represent $p(X_1)$ as a vector p_0, summing to 1

Graph vs. matrix



$\Leftrightarrow$

$$q = \left[\begin{array}{cc} 0.5 & 0.5 \\ 0.75 & 0.25 \end{array} \right]$$

Your Basic Markov Chain

```
rmarkovchain <- function(n,p0,q) {
  k <- length(p0)
  stopifnot(k==nrow(q),k==ncol(q),all.equal(rowSums(q),rep(1,time=k)))
  rinitial <- function() { sample(1:k,size=1,prob=p0) }
  rtransition <- function(x) { sample(1:k,size=1,prob=q[x,]) }
  return(rmarkov(n,rinitial,rtransition))
}
```

It runs:

```
> x <- rmarkovchain(1e4,c(0.5,0.5),q)
> head(x)
[1] 1 1 2 1 2 2
```

How do we know it works?

```
> ones <- which(x[-1e4]==1)
> twos <- which(x[-1e4]==2)
> signif(table(x[ones+1])/length(ones),3)
    1     2
0.489 0.511
> signif(table(x[twos+1])/length(twos),3)
    1     2
0.752 0.248
```

vs. $(0.5, 0.5)$ and $(0.75, 0.25)$ ideally
Uses law of large numbers + conditional independence

Hidden Markov Model (HMM)

X_t is Markov, but we see $Y_t = h(X_t) + \text{noise}$, not Markov
e.g.

```
> means <- c(10,-10)
> sds <- c(1,5)
> y <- rnorm(n=length(x),mean=means[x],sd=sds[x])
> signif(head(y),3)
[1]  11.00  10.00 -10.60  11.80 -16.30  -2.41
```

(noise and distortion might be much more complicated)

Variations

Interacting/coupled Markov chains: transition probability for chain 1 depends on its state and chain 2's state

Variations

Interacting/coupled Markov chains: transition probability for chain 1 depends on its state and chain 2's state

Continuous-time Markov chain: stay in the state for a random time, with exponential distribution, then take a chain step

Variations

Interacting/coupled Markov chains: transition probability for chain 1 depends on its state and chain 2's state

Continuous-time Markov chain: stay in the state for a random time, with exponential distribution, then take a chain step

Semi-Markov chain: like CTMC, but non-exponential holding times

Variations

Interacting/coupled Markov chains: transition probability for chain 1 depends on its state and chain 2's state

Continuous-time Markov chain: stay in the state for a random time, with exponential distribution, then take a chain step

Semi-Markov chain: like CTMC, but non-exponential holding times

Chain with complete connections: as in HMM, $Y_t = h(X_t) + \text{noise}$, but then $X_{t+1} = r(X_t, Y_t)$ (with no noise)

Invariant Distributions

$$\begin{aligned} p_1 &= p_0\mathbf{q} \\ p_2 &= p_1\mathbf{q} = p_0\mathbf{q}^2 \\ p_t &= p_{t-1}\mathbf{q} = p_0\mathbf{q}^t \end{aligned}$$

Fact: If the chain can go from any state to any other and back, and there are no fixed periods, then

$$p_t \rightarrow p_\infty = p_\infty\mathbf{q}$$

p_∞ = left eigenvector of $\mathbf{q}$ of eigenvalue 1
This is the **invariant distribution**

```
> table(rmarkovchain(1e4,c(0.5,0.5),q))
   1    2
5999 4001
> table(rmarkovchain(1e4,c(0.5,0.5),q))
   1    2
5996 4004
> table(rmarkovchain(1e4,c(0,1),q))
   1    2
5989 4011
> table(rmarkovchain(1e4,c(1,0),q))
   1    2
5996 4004
```

```
> eigen(t(q))
$values
[1]  1.00 -0.25

$vectors
          [,1]       [,2]
[1,] 0.8320503 -0.7071068
[2,] 0.5547002  0.7071068

> eigen(t(q))$vectors[,1]/sum(eigen(t(q))$vectors[,1])
[1] 0.6 0.4
```

The Long Run of a Markov Chain

In the long run, all the X_t come close to having the same distribution, the invariant distribution
They're still dependent, though
Ergodic theorem:

$$\frac{1}{n}\sum_{t=1}^{n} f(X_t) \rightarrow \sum_{x} p_\infty(x) f(x) = \mathbb{E}_{p_\infty}[f(X)]$$

time averages converge on expected values

Random Samples and Integrals

Law of large numbers: if $X_1, X_2, \ldots X_n$ all IID with p.d.f. p,

$$\frac{1}{n}\sum_{i=1}^{n} f(X_i) \rightarrow \mathbb{E}_p[f(X)] = \int f(x)p(x)dx$$

The **Monte Carlo principle**: to find $\int g(x)dx$, draw from p and take the sample mean of $f(x) = g(x)/p(x)$

Examples

Buffon's needle (homework!)

Examples

Buffon's needle (homework!)
Area of a complicated shape C: draw X uniformly from box around C, take average of $\mathbf{1}_C(X)$

Examples

Buffon's needle (homework!)
Area of a complicated shape C: draw X uniformly from box around C, take average of $\mathbf{1}_C(X)$
Any expectation value, variance, . . .

Examples

Buffon's needle (homework!)
Area of a complicated shape C: draw X uniformly from box around C, take average of $\mathbf{1}_C(X)$
Any expectation value, variance, ...
Anything your other classes teach you as integrals or expectations: significance levels, risk of portfolios, revenue of ads, thresholds for epidemics, ...

Bayes's Rule and Integrals

Bayes's rule:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x')p(x')dx'}$$

Seems like we need to know the integral

$$p(y) = \int p(y|x')p(x')dx'$$

Monte Carlo can be very accurate

Central limit theorem:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{g(x_i)}{p(x_i)} \rightsquigarrow \mathcal{N}\left(\int g(x)dx, \frac{\sigma^2_{g/p}}{n}\right)$$

Monte Carlo approximation to the integral is unbiased
RMS error $\propto n^{-1/2}$
$\therefore$ Just keep taking Monte Carlo draws, and the error gets as small as you like, even if g or x are very complicated

Importance Sampling

$$\int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx$$

$\therefore$ draw $X_1, X_2, \ldots X_n$ IID from q and

$$\frac{1}{n}\sum_{i=1}^{n} f(x_i)\frac{p(x_i)}{q(x_i)} \approx \int f(x)p(x)dx$$

$p(x)/q(x) =$ **importance weights** (ideally close to 1)

How Do We Do Monte Carlo?

Lots of Monte Carlo needs us to sample from an ugly distribution p
None of the methods from last time might work well for p
Markov chain Monte Carlo, MCMC: build a Markov chain whose invariant distribution is p
Run the chain, take its values

The Metropolis Algorithm

We know $p(x) = f(x)/c$ but we don't know c
Suppose

$$p(x)q(y|x) = p(y)q(x|y)$$

then p would be invariant ("detailed balance")

$$\frac{q(y|x)}{q(x|y)} = \frac{p(y)}{p(x)} = \frac{f(y)}{f(x)}$$

We don't need to know c!

Metropolis Algorithm (cont'd)

1. Set X_1 however, $t \leftarrow 1$
2. Proposal: Draw Z_t from some $r(\cdot|X_t)$
3. Draw $U_t \sim \mathrm{Unif}(0,1)$
4. If $U_t < f(Z_t)/f(X_t)$, then $X_{t+1} \leftarrow Z_t$, else $X_{t+1} \leftarrow X_t$
5. Increase t, go back to 2

Close to, but not quite, rejection method

```
rmetropolis <- function(n,rinitial,rproposal,f) {
  metrostep <- function(x) {
    z <- rproposal(x)
    u <- runif(1)
    return(if(u < f(z)/f(x)) { z } else { x } )
  }
  return(rmarkov(n,rinitial,metrostep))
}
```

Typically, discard first k values (**burn-in**), then only use every m^{th} value (low correlation), or average blocks of length m

Sampling from Bayes's Rule

$$p(x|y) \propto p(y|x)p(x)$$

so we can use Metropolis to draw a sample from $p(x|y)$ without really knowing it!
Key to modern Bayesian statistics

Gibbs Sampling

If X has many dimensions s, even writing $f(x) \propto p(x)$ can be hard
Could try to turn $X_1, X_2, \ldots X_s$ into a Markov chain but that might not work
Might be able to get $p(X_i | X_1, \ldots X_{i-1}, X_{i+1}, X_s) = p(X_i | X_{-i})$
The **Gibbs sampler**:

1. Set $X_1, X_2, \ldots X_s$ somehow
2. Pick a random i
3. Update X_i by drawing from $p(X_i | X_{-i})$
4. Go back to (2)

The sampler is a Markov chain on X
The invariant distribution is p

Summary

1. Break complicated simulations into many draws from basic distributions
   - Make later draws depend on earlier ones
   - Use the Markov property when you can
2. Monte Carlo is a stochastic way of evaluating integrals
   - Or expectation values or probabilities or...
   - Extra useful when the integrand is complicated or the space is high-dimensional
3. Markov chain Monte Carlo approximates integrals as averages over a Markov process with the right invariant distribution