Lecture 17: some mcmc practicalities

STAT 545: Intro. to Computational Statistics

Vinayak Rao

Purdue University

November 17, 2016

Summary so far...

Independent samples from prob. distrib. p is often difficult.

MCMC addresses this by producing dependent samples.

- Begin with an arbitrary initialization X_0.
- Sequentially produce samples $X_1 \to X_2 \to \ldots \to X_N$.

If the chain is stationary w.r.t. $p(x)$, irreducible and aperiodic:

$$\frac{1}{S}\sum_{i=1}^{S} h(X_i) \to \mathbb{E}_p[h]$$

Summary so far...

Independent samples from prob. distrib. p is often difficult.

MCMC addresses this by producing dependent samples.

- Begin with an arbitrary initialization X_0.
- Sequentially produce samples $X_1 \to X_2 \to \ldots \to X_N$.

If the chain is stationary w.r.t. $p(x)$, irreducible and aperiodic:

$$\frac{1}{S}\sum_{i=1}^{S} h(X_i) \to \mathbb{E}_p[h]$$

In practice, S is finite.
Assessing error is much harder

Convergence issues with MCMC

How much burn-in is required to forget initial state?

Convergence issues with MCMC

How much burn-in is required to forget initial state?

How well does your chain mix?

- Are our MCMC samples representative of the overall posterior? Difficult with multimodal distributions.
- Do we have enough samples to estimate expectations accurately? Tricky because of correlation between samples.

Recall

Burn-in time: time to 'forget' the arbitrary initialization.

Typically deal with burn-in by discarding the first B samples (e.g. $B = 1000$)

Recall

Burn-in time: time to 'forget' the arbitrary initialization.

Typically deal with burn-in by discarding the first B samples (e.g. $B = 1000$)

Sometimes people deal with sample dependence by 'thinning' the Markov chain: E.g. Use every mth sample (e.g. $m = 10$)

Thinning is usually unnecessary and increases variance of estimates (unless you want to save memory/computation).

Recall

Burn-in time: time to 'forget' the arbitrary initialization.

Typically deal with burn-in by discarding the first B samples (e.g. $B = 1000$)

Sometimes people deal with sample dependence by 'thinning' the Markov chain: E.g. Use every mth sample (e.g. $m = 10$)

Thinning is usually unnecessary and increases variance of estimates (unless you want to save memory/computation).

However, it's worthwhile remembering that N MCMC samples correspond to a smaller number of independent samples.

Effective sample size

A good diagnostic is the effective sample size (ESS):

$$N_{ESS} = \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho_k}$$

ρ_k is the auto-correlation between X_i and X_{i+k}:

$$\rho_k = \frac{\mathbb{E}[(X_{i+k} - \mu)(X_i - \mu)]}{\sigma^2}$$

(μ, σ^2) are mean and variance of the stationary distribution.

Effective sample size

CLT for Markov chains:

$$\left(\frac{1}{N}\sum_{i=1}^{N} f(X_i) - \mathbb{E}[f(X)]\right) \to \mathcal{N}(0, \sigma^2/N_{ESS})$$

$$N_{ESS} = \frac{N}{1 + 2\sum_{k=1}^{\infty}\rho_k},$$

$$\rho_k = \frac{\mathbb{E}[(f(X_{i+k}) - \mu)(f(X_i) - \mu)]}{\sigma^2}$$

(μ, σ^2) are mean,variance of $f(X)$ under stationary distribution.

Effective sample size

CLT for Markov chains:

$$\left(\frac{1}{N}\sum_{i=1}^{N} f(X_i) - \mathbb{E}[f(X)]\right) \to \mathcal{N}(0, \sigma^2/N_{ESS})$$

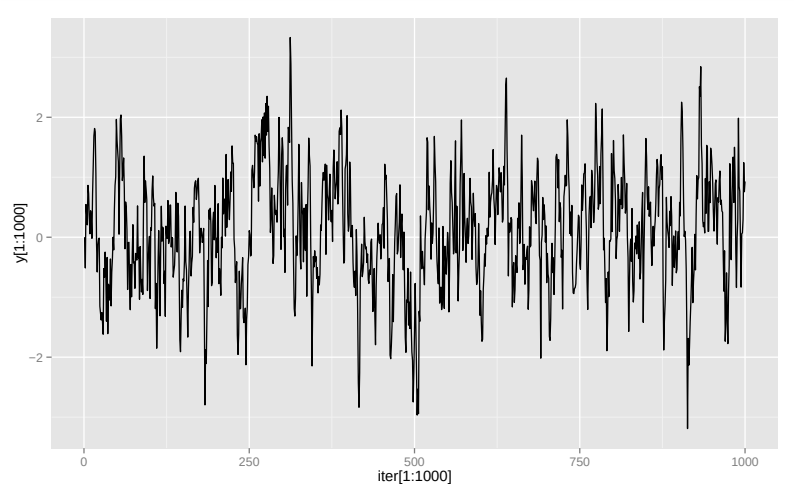
$$N_{ESS} = \frac{N}{1 + 2\sum_{k=1}^{\infty}\rho_k},$$

$$\rho_k = \frac{\mathbb{E}[(f(X_{i+k}) - \mu)(f(X_i) - \mu)]}{\sigma^2}$$

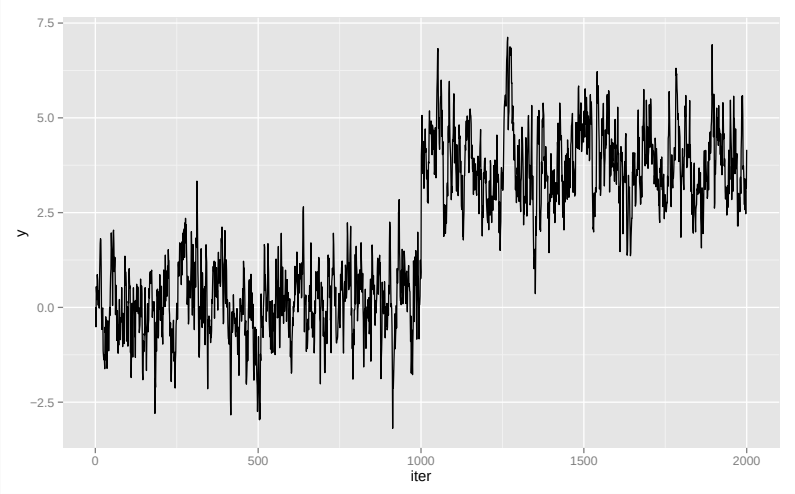
(μ, σ^2) are mean,variance of $f(X)$ under stationary distribution. Different variables/functions have different ESS. Often take the minimum of a few.

Effective sample size

The `Coda` package in `R`calculates this and other diagnostics.



ESS: 130.4

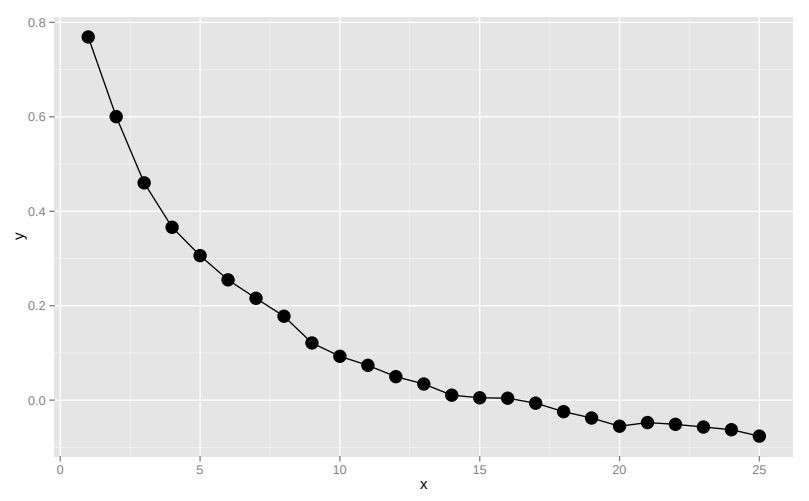


ESS: 9.21

```
> effectiveSize(data.frame(half=z[1:1000],full=z))
      half       full
260.997261   9.216991
```

Note: always useful to visualize traceplots.

Other diagnostics

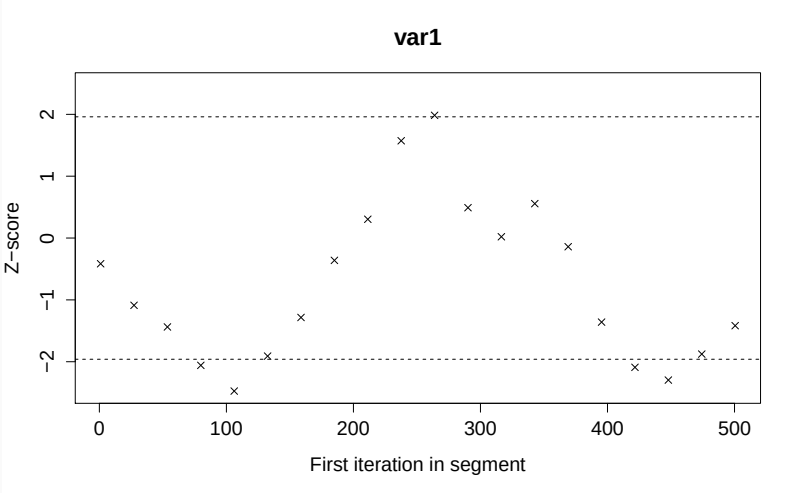


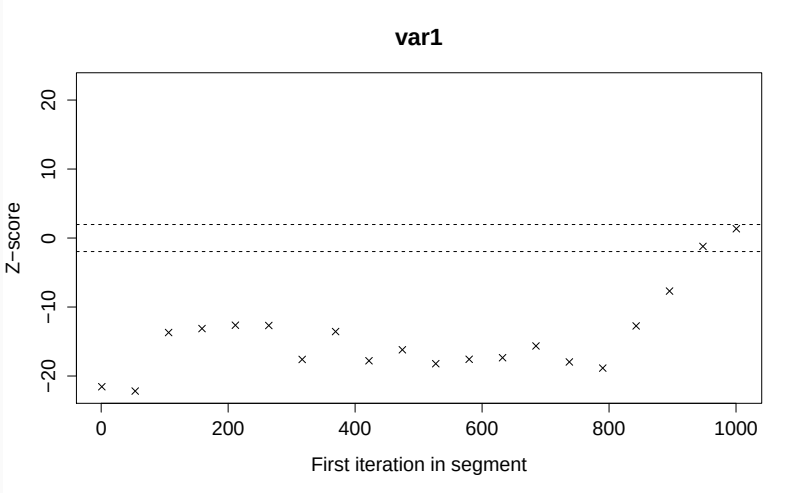
Correlation vs lag

```
> acf <- autocorr(mcmc(z[1:1000]),c(1:25))
```

Other diagnostics

Geweke diagnostic:





Compare 2 non-overlapping parts of the chain (in `R CODA` is the first 10% and last 50%, and test if their means come from the same distribution.

Can repeat, successively discarding initial parts.

```
> geweke.plot(mcmc(z[1:1000]))
> geweke.plot(mcmc(z))
```

Other diagnostics

Gelman-Rubin diagnostic: Run $m \geq 2$ independent chains with overdispersed starting points (e.g. sampled from the prior)

- Calculate within-chain variance and between-chain variance.

Other diagnostics

Gelman-Rubin diagnostic: Run $m \geq 2$ independent chains with overdispersed starting points (e.g. sampled from the prior)

- Calculate within-chain variance and between-chain variance.
- Former typically underestimates variance (bad mixing), and latter overestimates it (overdispersed initialization).
- If latter is much larger than former, run chain longer

Other diagnostics

Gelman-Rubin diagnostic: Run $m \geq 2$ independent chains with overdispersed starting points (e.g. sampled from the prior)

- Calculate within-chain variance and between-chain variance.
- Former typically underestimates variance (bad mixing), and latter overestimates it (overdispersed initialization).
- If latter is much larger than former, run chain longer

```
> gelman.diag(mcmc.list(mcmc(z[1:1000]),
  mcmc(z[1001:2000])))
Potential scale reduction factors:

     Point est. Upper C.I.
[1,]       4.87       10.6
```

Potential scale reduction factor much larger than 1 is trouble

One long chain vs many shorter chains?

M short chain of length N vs 1 chain of length MN:

One long chain vs many shorter chains?

M short chain of length N vs 1 chain of length MN:

Pros:

- Diverse initialization likely means better exploration of different modes.
- Allows easy parallelization

One long chain vs many shorter chains?

M short chain of length N vs 1 chain of length MN:

Pros:

- Diverse initialization likely means better exploration of different modes.
- Allows easy parallelization

Cons:

- Each chain still has a burn-in period B. Must discard MB samples vs B for a single chain.

Debugging MCMC

Never mind mixing, how do we know our sampler is correct?!

After changing something, how do we know it's still correct?

Debugging MCMC

Never mind mixing, how do we know our sampler is correct?!

After changing something, how do we know it's still correct?
Can never be sure, but useful to run a few standard tests.

Debugging MCMC

Never mind mixing, how do we know our sampler is correct?!

After changing something, how do we know it's still correct?
Can never be sure, but useful to run a few standard tests.

Do your results make sense for special cases?

Debugging MCMC

Never mind mixing, how do we know our sampler is correct?!

After changing something, how do we know it's still correct?
Can never be sure, but useful to run a few standard tests.

Do your results make sense for special cases?

Compare different samplers: a Gibbs and MH sampler should give similar results, but unlikely to have same errors.

Debugging MCMC

Never mind mixing, how do we know our sampler is correct?!

After changing something, how do we know it's still correct? Can never be sure, but useful to run a few standard tests.

Do your results make sense for special cases?

Compare different samplers: a Gibbs and MH sampler should give similar results, but unlikely to have same errors.

On scaled down datasets, compare with simple Monte Carlo methods like rejection/importance sampling.

Debugging MCMC

Never mind mixing, how do we know our sampler is correct?!

After changing something, how do we know it's still correct? Can never be sure, but useful to run a few standard tests.

Do your results make sense for special cases?

Compare different samplers: a Gibbs and MH sampler should give similar results, but unlikely to have same errors.

On scaled down datasets, compare with simple Monte Carlo methods like rejection/importance sampling.

Can you analytically calculate the posterior for 1 observation or 2 states or 2 time-periods?

Debugging MCMC (contd)

Recall, in a Bayesian setting we want samples from $p(Y|X)$.

Generate a new dataset every MCMC iteration.

Debugging MCMC (contd)

Recall, in a Bayesian setting we want samples from $p(Y|X)$.

Generate a new dataset every MCMC iteration.

Every iteration, MCMC samples $p(Y_{n+1}|X, Y_n)$.

Debugging MCMC (contd)

Recall, in a Bayesian setting we want samples from $p(Y|X)$.

Generate a new dataset every MCMC iteration.

Every iteration, MCMC samples $p(Y_{n+1}|X, Y_n)$.

After this sample a new dataset $p(X_{n+1}|Y_{n+1})$.

Debugging MCMC (contd)

Recall, in a Bayesian setting we want samples from $p(Y|X)$.

Generate a new dataset every MCMC iteration.

Every iteration, MCMC samples $p(Y_{n+1}|X, Y_n)$.

After this sample a new dataset $p(X_{n+1}|Y_{n+1})$.

Overall, a Gibbs sampler on (X, Y)

Debugging MCMC (contd)

Recall, in a Bayesian setting we want samples from $p(Y|X)$.

Generate a new dataset every MCMC iteration.

Every iteration, MCMC samples $p(Y_{n+1}|X, Y_n)$.

After this sample a new dataset $p(X_{n+1}|Y_{n+1})$.

Overall, a Gibbs sampler on (X, Y)

What is the joint distribution?

What is the marginal distribution of Y?

Debugging MCMC (contd)

Recall, in a Bayesian setting we want samples from $p(Y|X)$.

Generate a new dataset every MCMC iteration.

Every iteration, MCMC samples $p(Y_{n+1}|X, Y_n)$.

After this sample a new dataset $p(X_{n+1}|Y_{n+1})$.

Overall, a Gibbs sampler on (X, Y)

What is the joint distribution?

What is the marginal distribution of Y?

James P. Hobert and Galin L. Jones, *Honest Exploration of Intractable Probability Distributions via Markov Chain Monte Carlo* Statist. Sci. Volume 16, Number 4 (2001), 312-334.

Using MCMC samples:

Consider a Markov chain on (x, y, z) with stationary distrib. $P()$.
We obtain samples $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \ldots$

Using MCMC samples:

Consider a Markov chain on (x, y, z) with stationary distrib. $P()$.
We obtain samples $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \ldots$

What is $\mathbb{E}[f(x, y, z)]$?

$$\mathbb{E}[f(x, y, z)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i, y_i, z_i)$$

Using MCMC samples:

Consider a Markov chain on (x, y, z) with stationary distrib. $P()$.
We obtain samples $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \ldots$

What is $\mathbb{E}[f(x, y, z)]$?

$$\mathbb{E}[f(x, y, z)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i, y_i, z_i)$$

What is $P(x = 1)$?

Using MCMC samples:

Consider a Markov chain on (x, y, z) with stationary distrib. $P()$.
We obtain samples $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \ldots$

What is $\mathbb{E}[f(x, y, z)]$?

$$\mathbb{E}[f(x, y, z)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i, y_i, z_i)$$

What is $P(x = 1)$?

$$P(x = 1) = \mathbb{E}[\delta(x = 1)] \approx \frac{1}{N} \sum_{i=1}^{N} \delta(x_i = 1)$$

Using MCMC samples:

Consider a Markov chain on (x, y, z) with stationary distrib. $P()$.
We obtain samples $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \ldots$

What is $\mathbb{E}[f(x, y, z)]$?

$$\mathbb{E}[f(x, y, z)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i, y_i, z_i)$$

What is $P(x = 1)$?

$$P(x = 1) = \mathbb{E}[\delta(x = 1)] \approx \frac{1}{N} \sum_{i=1}^{N} \delta(x_i = 1)$$

Can we do better? E.g. what if x is continuous?

Rao-Blackwellization:

Suppose we can calculate $P(x|y,z)$.

This is the case if our Markov chain is a Gibbs sampler.

Rao-Blackwellization:

Suppose we can calculate $P(x|y,z)$.
This is the case if our Markov chain is a Gibbs sampler.
Then:

$$P(x=1) = \int\int P(x=1,y,z)\mathrm{d}y\mathrm{d}z$$

Rao-Blackwellization:

Suppose we can calculate $P(x|y,z)$.
This is the case if our Markov chain is a Gibbs sampler.
Then:

$$\begin{aligned} P(x=1) &= \int\int P(x=1,y,z)\mathrm{d}y\mathrm{d}z \\ &= \int\int P(x=1|y,z)p(y,z)\mathrm{d}y\mathrm{d}z \end{aligned}$$

Rao-Blackwellization:

Suppose we can calculate $P(x|y,z)$.
This is the case if our Markov chain is a Gibbs sampler.
Then:

$$\begin{aligned} P(x=1) &= \int\int P(x=1,y,z)\mathrm{d}y\mathrm{d}z \\ &= \int\int P(x=1|y,z)p(y,z)\mathrm{d}y\mathrm{d}z \\ &\approx \frac{1}{N}\sum_{i=1}^{N} P(x=1|y_i,z_i) \end{aligned}$$

Rao-Blackwellization:

Suppose we can calculate $P(x|y,z)$.
This is the case if our Markov chain is a Gibbs sampler.
Then:

$$\begin{aligned} P(x=1) &= \int\int P(x=1,y,z)\mathrm{d}y\mathrm{d}z \\ &= \int\int P(x=1|y,z)p(y,z)\mathrm{d}y\mathrm{d}z \\ &\approx \frac{1}{N}\sum_{i=1}^{N} P(x=1|y_i,z_i) \end{aligned}$$

Typically, this estimate will have lower variance.