Bayesian Networks in Educational Assessment

Estimating Parameters with MCMC

Roy Levy
Arizona State University
Roy.Levy@asu.edu

Bayesian Inference: Expanding Our Context

Posterior Distribution

Posterior distribution for ***unknowns*** given ***knowns*** is

$$p(unknowns \mid knowns) \propto p(knowns \mid unknowns)\, p(unknowns)$$

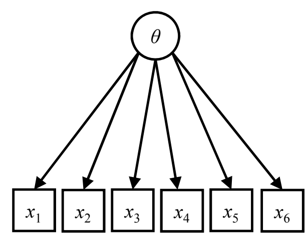
Inference about examinee latent variables (θ) given observables ($\mathbf{x}$)

$$p(\theta \mid \mathbf{x}) \propto p(\mathbf{x} \mid \theta)\, p(\theta)$$

Example: ACED Bayes Net Fragment for *Common Ratio*

- θ = *Common Ratio*
- $\mathbf{x}$ = Observables from tasks that measure *Common Ratio*

Bayes Net Fragment



θ = *Common Ratio*

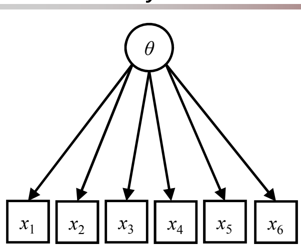
$\boxed{p(\theta)}$

xs = Observables from tasks that measure *Common Ratio*

$$\boxed{p(\mathbf{x} \mid \theta) = \prod_{j=1}^{J} p(x_j \mid \theta)}$$

$$p(\theta \mid x) \propto p(x \mid \theta)\, p(\theta)$$

Probability Distribution for the Latent Variable



θ = *Common Ratio*

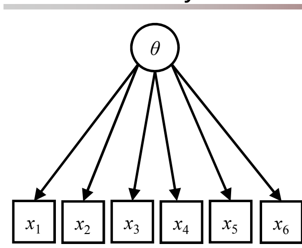
$\theta \sim$ Categorical($\boldsymbol{\lambda}$)

ACED Example

- 2 Levels of θ (Low, High)
- $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ contains probabilities for Low and High

	θ (Common Ratio)	
	1	2
Prob.	λ_1	λ_2

Probability Distribution for the Observables



xs = Observables from tasks that measure *Common Ratio*

$(x_j \mid \theta = c) \sim \text{Bernoulli}(\pi_{cj})$

ACED Example

- π_{cj} is the probability of correct response on task j given $\theta = c$

	$p(x_j \mid \theta)$	
θ	0	1
1	$1-\pi_{1j}$	π_{1j}
2	$1-\pi_{2j}$	π_{2j}

Bayesian Inference

$p(\theta \mid \mathbf{x}) \propto p(\mathbf{x} \mid \theta)\, p(\theta)$

	θ (Common Ratio)	
	1	2
Prob.	λ_1	λ_2

If the λs and πs are unknown, they become subject to posterior inference too

	$p(x_j \mid \theta)$	
θ	0	1
1	$1 - \pi_{1j}$	π_{1j}
2	$1 - \pi_{2j}$	π_{2j}

Bayesian Inference

$p(\theta \mid \mathbf{x}) \propto p(\mathbf{x} \mid \theta)\, p(\theta)$

	$p(x_j \mid \theta)$	
θ	0	1
1	$1 - \pi_{1j}$	π_{1j}
2	$1 - \pi_{2j}$	π_{2j}

A convenient choice for prior distribution is the beta distribution

$$\pi_{cj} \sim \text{Beta}(\alpha_{\pi_c}, \beta_{\pi_c})$$

ACED Example: $\pi_{1j} \sim \text{Beta}(1, 1)$ $\pi_{2j} \sim \text{Beta}(1, 1)$

For first task, constrain ($\pi_{21} > \pi_{11}$) to resolve indeterminacy in the latent variable and avoid label switching

Bayesian Inference

$p(\theta \mid \mathbf{x}) \propto p(\mathbf{x} \mid \theta)\, p(\theta)$

	θ (Common Ratio)	
	1 (Low)	2 (High)
Prob.	λ_1	λ_2

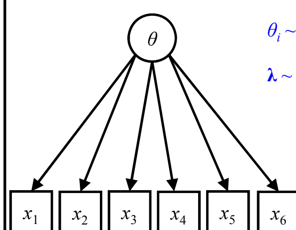
A convenient choice for the prior distribution is the Dirichlet distribution

$$\boldsymbol{\lambda} \sim \text{Dirichlet}(\boldsymbol{\alpha}_\lambda) \qquad \boldsymbol{\alpha}_\lambda = (\alpha_{\lambda_1}, \alpha_{\lambda_2})$$

which generalizes the Beta distribution to the case of multiple categories

ACED Example: $\boldsymbol{\lambda} = (\lambda_1, \lambda_2) \sim \text{Dirichlet}(1, 1)$

Model Summary



$\theta_i \sim \text{Categorical}(\boldsymbol{\lambda})$

$\boldsymbol{\lambda} \sim \text{Dirichlet}(1, 1)$

$(x_{ij} \mid \theta_i = c) \sim \text{Bernoulli}(\pi_{cj})$

$\pi_{11} \sim \text{Beta}(1, 1)$

$\pi_{21} \sim \text{Beta}(1, 1)\ I(\pi_{21} > \pi_{11})$

$\pi_{cj} \sim \text{Beta}(1, 1)$ for others obs.

JAGS Code

JAGS Code

```
for (i in 1:n){
 for(j in 1:J){
  x[i,j] ~ dbern(pi[theta[i],j])
 }
}
```

$(x_{ij} \mid \theta_i = c) \sim \text{Bernoulli}(\pi_{cj})$

Referencing the table for π_js in terms of $\theta = 1$ or 2

	$p(x_j \mid \theta)$	
θ	0	1
1	$1 - \pi_{1j}$	π_{1j}
2	$1 - \pi_{2j}$	π_{2j}

JAGS Code

```
pi[1,1] ~ dbeta(1,1)
```
$\pi_{11} \sim \text{Beta}(1, 1)$

```
pi[2,1] ~ dbeta(1,1) T(pi[1,1], )
```
$\pi_{21} \sim \text{Beta}(1, 1)\ I(\pi_{21} > \pi_{11})$

```
for(c in 1:C){
  for(j in 2:J){
    pi[c,j] ~ dbeta(1,1)
  }
}
```
$\pi_{cj} \sim \text{Beta}(1, 1)$ for remaining observables

JAGS Code

```
for (i in 1:n){
  theta[i] ~ dcat(lambda[])
}
```
$\theta_i \sim \text{Categorical}(\boldsymbol{\lambda})$

```
lambda[1:C] ~ ddirch(alpha_lambda[])
for(c in 1:C){
  alpha_lambda[c] <- 1
}
```
$\boldsymbol{\lambda} \sim \text{Dirichlet}(1, 1)$

Markov Chain Monte Carlo

Estimation in Bayesian Modeling

- Our "answer" is a posterior distribution
  - All parameters treated as random, not fixed
- Contrasts with frequentist approaches to inference, estimation
  - Parameters are fixed, so estimation comes to finding the single best value
  - "Best" here in terms of a criterion (ML, LS, etc.)
- Peak of a mountain vs. mapping the entire terrain of peaks, valleys, and plateaus (of a landscape)

What's In a Name?

Markov chain ***Monte Carlo***

- Construct a sampling algorithm to ***simulate*** or ***draw from*** the posterior.
- Collect many such draws, which serve to empirically approximate the posterior distribution, and can be used to empirical approximate summary statistics.

Monte Carlo Principle:

> Anything we want to know about a random variable θ can be learned by sampling many times from $f(\theta)$, the density of θ.
>
> -- Jackman (2009)

What's In a Name?

Markov ***chain*** Monte Carlo

- Values really generated as a sequence or chain
- t denotes the step in the chain
- $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(t)}, \ldots, \theta^{(T)}$
- Also thought of as a time indicator

Markov chain Monte Carlo

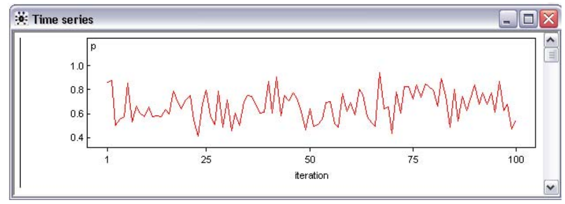
- Follows the Markov property…

The Markov Property

- Current state depends on previous position
  - Examples: weather, checkers, baseball counts & scoring
- Next state conditionally independent of past, given the present
  - Akin to a full mediation model
- $p(\theta^{(t+1)} \mid \theta^{(t)}, \theta^{(t-1)}, \theta^{(t-2)}, \ldots) = p(\theta^{(t+1)} \mid \theta^{(t)})$

$$\boxed{\theta^{(0)}} \rightarrow \boxed{\theta^{(1)}} \rightarrow \boxed{\theta^{(2)}} \rightarrow \boxed{\theta^{(3)}}$$

Visualizing the Chain: Trace Plot



Markov Chain Monte Carlo

- Markov chains are ***sequences of numbers*** that have the Markov property
  - Draws in cycle t+1 depend on values from cycle t, but given those not on previous cycles (Markov property)
- Under certain assumptions Markov chains reach ***stationarity***
- The collection of values converges to a distribution, referred to as a stationary distribution
  - Memoryless: It will “forget” where it starts
  - Start anywhere, will reach stationarity if regularity conditions hold
  - For Bayes, set it up so that this is the posterior distribution
- Upon convergence, samples from the chain approximate the stationary (posterior) distribution

Assessing Convergence

Diagnosing Convergence

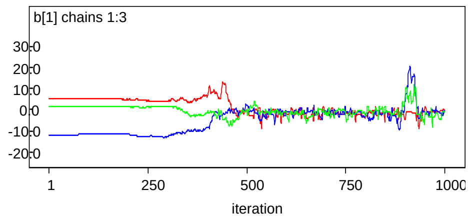
- With MCMC, convergence to a ***distribution***, not a point
- ML:
  - Convergence is when we’ve reached the highest point in the likelihood,
  - The highest peak of the mountain
- MCMC:
  - Convergence when we’re sampling values from the correct distribution,
  - We are mapping the entire terrain accurately

Diagnosing Convergence

- A properly constructed Markov chain is guaranteed to converge to the stationary (posterior) distribution…eventually
- Upon convergence, it will sample over the full support of the stationary (posterior) distribution…over an ∞ number of draws
- In a finite chain, no guarantee that the chain has converged or is sampling through the full support of the stationary (posterior) distribution
- Many ways to diagnose convergence
- Whole software packages dedicated to just assessing convergence of chains (e.g., R packages ‘coda’ and ‘boa’)

Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF)

- Run ***multiple*** chains from dispersed starting points
- Suggest convergence when the chains come together
- If they all go to the same place, it's probably the stationary distribution

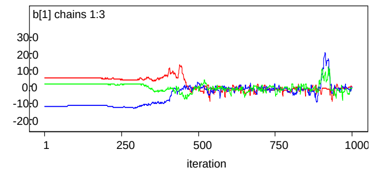


Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF)

- An analysis of variance type argument
- *PSRF* or R =

$$\frac{\text{Total Variance}}{\text{Within Chain Variance}} = \frac{\text{Between Chain Variance} + \text{Within Chain Variance}}{\text{Within Chain Variance}}$$

- If there is substantial between-chain variance, will be >> 1



Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF)

- Run ***multiple*** chains from dispersed starting points
- Suggest convergence when the chains come together
- Operationalized in terms of partitioning variability
- Run multiple chains for $2T$ iterations, discard first half
- Examine between and within chain variability
- Various versions, modifications suggested over time

Potential Scale Reduction Factor (PSRF)

- For any θ, for any chain c the within-chain variance is

$$W_c = \frac{1}{T-1}\sum_{t=1}^{T}(\theta_{(c)}^{(t)} - \bar{\theta}_{(c)})^2$$

- For all chains, the pooled within-chain variance is

$$W = \frac{1}{C}\sum_{c=1}^{C} W_c = \frac{1}{C(T-1)}\sum_{c=1}^{C}\sum_{t=1}^{T}(\theta_{(c)}^{(t)} - \bar{\theta}_{(c)})^2$$

Potential Scale Reduction Factor (PSRF)

- The between-chain variance is

$$B = \frac{T}{C-1}\sum_{c=1}^{C}(\bar{\theta}_{(c)} - \bar{\theta})^2$$

- The estimated variance is

$$\hat{V}ar(\theta) = (T - 1/T)W + (1/T)B$$

Potential Scale Reduction Factor (PSRF)

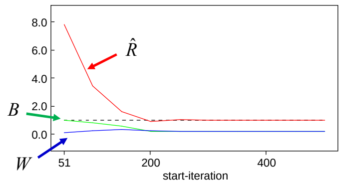
- The potential scale reduction factor is

$$\hat{R} = \sqrt{\frac{\hat{V}ar(\theta)}{W}}$$

- If close to 1 (e.g., < 1.1) for all parameters, can conclude convergence

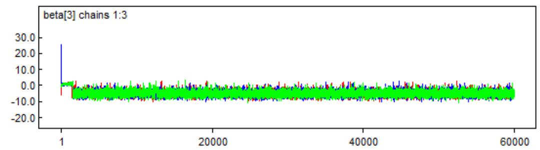
Potential Scale Reduction Factor (PSRF)

- Examine it over "time", look for $\hat{R} \to 1$, stability of B and W
- If close to 1 (e.g., < 1.2, or < 1.1) can conclude convergence



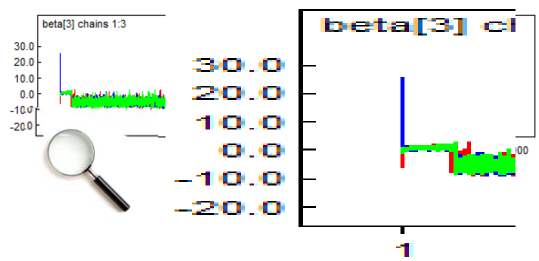
Assessing Convergence: No Guarantees

Multiple chains coming together does not guarantee they have converged



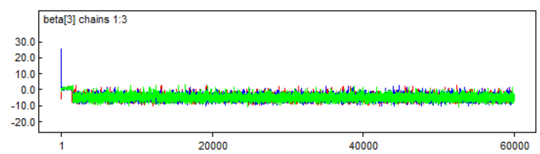
Assessing Convergence: No Guarantees

multiple chains come together does not guarantee they have converged



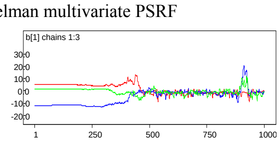
Assessing Convergence: No Guarantees

Multiple chains coming together does not guarantee they have converged



Assessing Convergence

- Recommend running multiple chains far apart and determine when they reach the same "place"
  - PSRF criterion an approximation to this
  - Akin to starting ML from different start values and seeing if they reach the same maximum
  - Here, convergence to a distribution, not a point
- A chain hasn't converged until ***all*** parameters converged
  - Brooks & Gelman multivariate PSRF



Serial Dependence

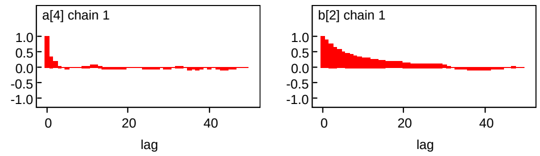
Serial Dependence

- Serial dependence between draws due to the dependent nature of the draws (i.e., the Markov structure)
- $p(\theta^{(t+1)} \mid \theta^{(t)}, \theta^{(t-1)}, \theta^{(t-2)}, \ldots) = p(\theta^{(t+1)} \mid \theta^{(t)})$

$$\boxed{\theta^{(0)}} \rightarrow \boxed{\theta^{(1)}} \rightarrow \boxed{\theta^{(2)}} \rightarrow \boxed{\theta^{(3)}}$$

- However there is a ***marginal*** dependence across multiple lags
- Can examine the autocorrelation across different lags

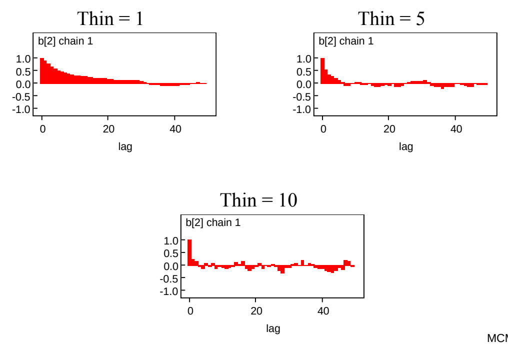
Autocorrelation



Thinning

- Can "thin" the chain by dropping certain iterations
  Thin = 1 → keep every iteration
  Thin = 2 → keep every other iteration (1, 3, 5,…)
  Thin = 5 → keep every 5th iteration (1, 6, 11,…)
  Thin = 10 → keep every 10th iteration (1, 11, 21,…)
  Thin = 100 → keep every 100th iteration (1, 101, 201,…)

Thinning



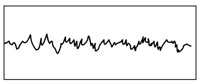
Thinning

- Can "thin" the chain by dropping certain iterations
  Thin = 1 → keep every iteration
  Thin = 2 → keep every other iteration (1, 3, 5,…)
  Thin = 5 → keep every 5th iteration (1, 6, 11,…)
  Thin = 10 → keep every 10th iteration (1, 11, 21,…)
  Thin = 100 → keep every 100th iteration (1, 101, 201,…)
- Thinning ***does not*** provide a better portrait of the posterior
  - A loss of information
- May want to keep, and account for time-series dependence
- Useful when data storage, other computations an issue
  - *I want 1000 iterations, rather have 1000 approximately independent iterations*
- Dependence ***within*** chains, but none ***between*** chains

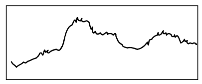
Mixing

Mixing

- We don't want the sampler to get "stuck" in some region of the posterior , or ignore a certain area of the posterior
- Mixing refers to the chain "moving" throughout the support of the distribution in a reasonable way



relatively good mixing



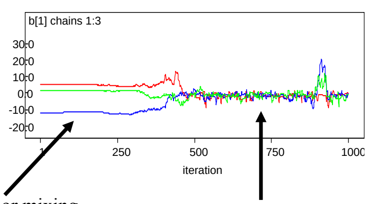
relatively poor mixing

Mixing

- Mixing ≠ convergence, but better mixing usually leads to faster convergence
- Mixing ≠ autocorrelation, but better mixing usually goes with lower autocorrelation (and cross-correlations between parameters)
- With better mixing, then for a given number of MCMC iterations, get more information about the posterior
  - Ideal scenario is independent draws from the posterior
- With worse mixing, need more iterations to (a) achieve convergence and (b) achieve a desired level of precision for the summary statistics of the posterior

Mixing

- Chains may mix differently at different times
- Often indicative of an adaptive MCMC algorithm



relatively poor mixing

relatively good mixing

Mixing

- Slow mixing can also be caused by high dependence between parameters
  - Example: multicollinearity
- Reparameterizing the model can improve mixing
  - Example: centering predictors in regression

Stopping the Chain(s)

When to Stop The Chain(s)

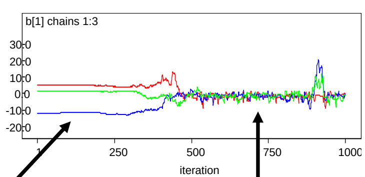
- Discard the iterations prior to convergence as ***burn-in***
- How many more iterations to run?
  - As many as you want ☺
  - As many as time provides
- Autocorrelaion complicates things
- Software may provide the "MC error"
  - Estimate of the sampling variability of the sample mean
  - Sample here is the sample of iterations
  - Accounts for the dependence between iterations
  - Guideline is to go at least until MC error is less than 5% of the posterior standard deviation
- Effective sample size
  - Approximation of how many independent samples we have

Steps in MCMC in Practice

Steps in MCMC (1)

- Setup MCMC using any of a number of algorithms
  - Program yourself (have fun ☺)
  - Use existing software (BUGS, JAGS)
- Diagnose convergence
  - Monitor trace plots, PSRF criteria
- Discard iterations prior to convergence as ***burn-in***
  - Software may indicate a minimum number of iterations needed
  - A lower bound

Adapting MCMC → Automatic Discard



relatively poor mixing
during adaptive phase

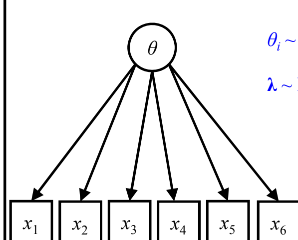
relatively good mixing
after adaptive phase

Steps in MCMC (2)

- Run the chain for a desired number of iterations
  - Understanding serial dependence/autocorrelation
  - Understanding mixing
- Summarize results
  - Monte Carlo principle
  - Densities
  - Summary statistics

ACED Example

Model Summary



$\theta_i \sim \text{Categorical}(\boldsymbol{\lambda})$

$\boldsymbol{\lambda} \sim \text{Dirichlet}(1, 1)$

$(x_{ij} \mid \theta_i = c) \sim \text{Bernoulli}(\pi_{cj})$

$\pi_{11} \sim \text{Beta}(1, 1)$

$\pi_{21} \sim \text{Beta}(1, 1)\ \ I(\pi_{21} > \pi_{11})$

$\pi_{cj} \sim \text{Beta}(1, 1)$ for others obs.

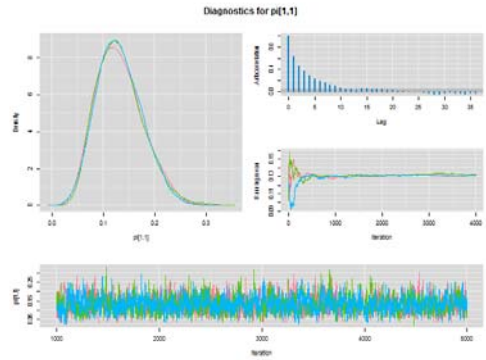
ACED Example

See 'ACED Analysis.R' for Running the analysis in R

See Following Slides for Select Results

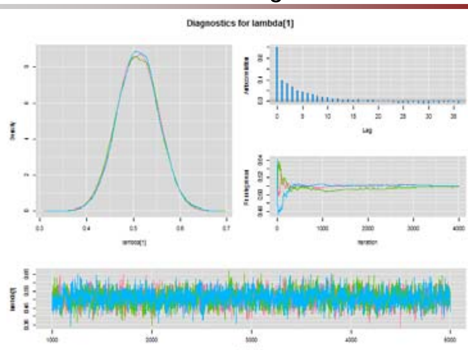
MCMC 55

Convergence Assessment (1)



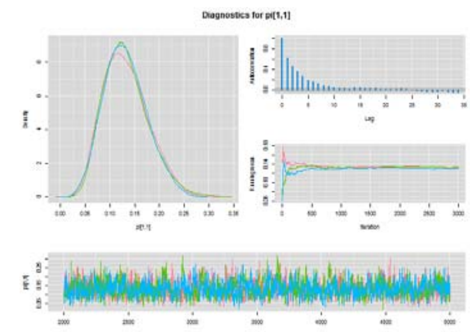
:MC 56

Convergence Assessment (2)



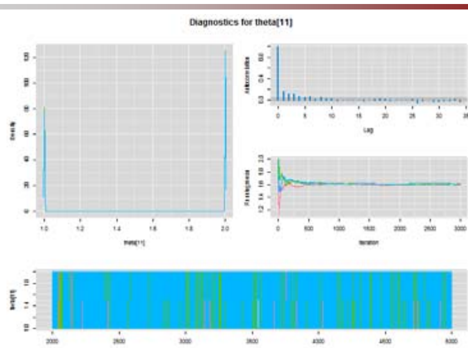
CMC 57

Posterior Summary (1)



MCMC 58

Posterior Summary (2)



CMC 59

Posterior Summary (3)

	Mean	SD	Naive SE	Time-series SE	0.025	0.25	0.5	0.75	0.975	Median	95% HPD lower	95% HPD Upper
lambda[1]	0.51	0.04	0	0	0.42	0.48	0.51	0.54	0.6	0.51	0.43	0.6
lambda[2]	0.49	0.04	0	0	0.4	0.46	0.49	0.52	0.58	0.49	0.4	0.57
pi[1,1]	0.13	0.04	0	0	0.06	0.1	0.13	0.16	0.23	0.13	0.05	0.22
pi[2,1]	0.84	0.04	0	0	0.75	0.81	0.84	0.87	0.91	0.84	0.75	0.92
pi[1,2]	0.22	0.05	0	0	0.12	0.18	0.22	0.26	0.33	0.22	0.12	0.33
pi[2,2]	0.98	0.02	0	0	0.93	0.97	0.99	0.99	1	0.99	0.94	1
pi[1,3]	0.02	0.01	0	0	0	0.01	0.02	0.03	0.06	0.02	0	0.05
pi[2,3]	0.19	0.04	0	0	0.12	0.17	0.19	0.22	0.28	0.19	0.12	0.27
pi[1,4]	0.03	0.02	0	0	0.01	0.02	0.03	0.04	0.07	0.03	0	0.06
pi[2,4]	0.23	0.05	0	0	0.15	0.2	0.23	0.26	0.33	0.23	0.15	0.33
pi[1,5]	0.15	0.04	0	0	0.08	0.12	0.15	0.17	0.22	0.15	0.08	0.22
pi[2,5]	0.64	0.05	0	0	0.53	0.6	0.64	0.67	0.74	0.64	0.53	0.74
pi[1,6]	0.17	0.04	0	0	0.1	0.14	0.17	0.2	0.25	0.17	0.1	0.25
pi[2,6]	0.82	0.05	0	0	0.72	0.79	0.82	0.86	0.92	0.82	0.73	0.92
theta[1]	2	0.06	0	0	2	2	2	2	2	2	2	2
theta[2]	1	0.02	0	0	1	1	1	1	1	1	1	1
theta[3]	1	0.01	0	0	1	1	1	1	1	1	1	1
theta[4]	1.97	0.17	0	0	1	2	2	2	2	2	2	2
theta[5]	1.17	0.38	0	0.01	1	1	1	1	2	1	1	2
theta[6]	1	0.01	0	0	1	1	1	1	1	1	1	1
theta[7]	1.01	0.07	0	0	1	1	1	1	1	1	1	1

MCMC 60

Summary and Conclusion

Summary

- Dependence on initial values is "forgotten" after a sufficiently long run of the chain (memoryless)
- Convergence to a ***distribution***
  - Recommend monitoring multiple chains
  - PSRF as approximation
- Let the chain "burn-in"
  - Discard draws prior to convergence
  - Retain the remaining draws as draws from the posterior
- Dependence across draws induce autocorrelations
  - Can thin if desired
- Dependence across draws within and between parameters can slow mixing
  - Reparameterizing may help

Wise Words of Caution

Beware: MCMC sampling can be dangerous!

-- Spiegelhalter, Thomas, Best, & Lunn (2007)
(WinBUGS User Manual)