Spring Lecture Series Workshop April 18-20, 2019

Dr. Jyotishka Datta

University of Arkansas, Fayetteville

April 14, 2019

Outline for Today's talk

AR(1) Models

Computing Tools
- Nice Markov Chains

MCMC

Convergence

Effective Sample Size

Stan: Building Blocks

Stan: Demonstration

AR(1) models

- Time series: Stochastic process in (discrete) time, sometimes equally spaced.
- Simplest non-trivial model: AR(1): Auto-regressive of order 1.

$$x_t = \phi_1 x_{t-1} + \epsilon_t, \quad \overbrace{\epsilon_t \sim \mathcal{N}(0, \nu)}^{\text{Innovation}}, \; \phi_1 = \text{AR parameter.}$$

- ϵ_t: random shock, added to predicted value for x_t given by $\phi_1 x_{t-1}$. 'Shock' is unpredictable $\mathbb{E}(\epsilon_t) = 0$.
- Notation: $X_t \sim \mathrm{AR}(1 \mid \phi_1, \nu)$.
- AR(1) models are of major interest by themselves as simple models for many situations but also as building blocks for more complex time series.

Stationarity

- Stationarity: the n-variate joint distribution of $x_{s:s+n-1} = (x_s, x_{s+1}, \ldots, x_{s+n-1})^T$ doesn't depend on s for any $n \geq 1$.
- Weak stationarity refers to the mean and variance-covariance of the joint distribution, but in the case of linear, normal models those moments characterize the full joint distribution.
- For $n = 1$ - each x_t has the same distribution.
- For $n = 2$ - each pair (x_t, x_s) has the same distribution.
- Assume $\mathbb{E}(x_t) = m, \mathbf{V}(x_t) = s$, then for all t, $x_t \sim \mathcal{N}(m, s)$.

Stationarity

- Conditional distribution: $p(x_t \mid x_{t-1}) = \mathcal{N}(\phi_1 x_{t-1}, \nu)$.
- Implies the following:

$$
\begin{aligned}
m = \mathbb{E}(x_t) &= \mathbb{E}[\mathbb{E}(x_t \mid xt-1)] = \mathbb{E}(\phi x_{t-1}] = \phi m \\
s = \mathbf{V}(x_t) &= \mathbb{E}[\mathbf{V}(x_t \mid x_{t-1})] + \mathbf{V}[\mathbb{E}(x_t \mid x_{t-1})] = \nu + \phi^2 s \\
&\Rightarrow s = \nu/(1-\phi^2)
\end{aligned}
$$

- **This can only happen if $|\phi| < 1$: a characteristic of stationary AR(1) process.**

Linear Process

- Iterate the AR equation to get:

$$x_t = \epsilon_t + \phi_1 \epsilon_{t-1} + \cdots + \phi_1^k \epsilon_{t-k} + \cdots \qquad \text{(Linear)}$$

- Linear process: linear function of current and past innovations, and a sum of independent stochastic elements that are weighted by the AR parameter.
- If $|\phi| < 1$ - current value of x-process is less and less dependent on the past innovations. Otherwise, an explosive process results.
- Backshift operator: $Bx_t = x_{t-1}$, $B^k x_t = x_{t-k}$ $\forall k > 0$.
- $(1 - \phi B)x_t = \epsilon_t \equiv x_t = (1 - \phi B)^{-1}(\epsilon_t)$.
- Use $(1 - \phi B)^{-1} = 1 + \phi B + \phi^2 B^2 + \cdots$ to derive (Linear).
- Of course, linear processes can be non-Gaussian as well

Autocorrelations

- Covariance at lag k: $\gamma(k) = C(x_t, x_{t\pm k}) = \phi^k s$.
- Correlation at lag k: $\rho(k) = \rho^k$.
- Joint distribution $x_{1:n} \sim \mathcal{N}(0, s\Phi_n)$, where

$$\Phi_n = \begin{pmatrix} 1 & \phi & \phi^2 & \cdots & \phi^{n-1} \\ \phi & 1 & \phi & \cdots & \phi^{n-2} \\ \phi^2 & \phi & 1 & \cdots & \phi^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi^{n-1} & \phi^{n-2} & \phi^{n-3} & \cdots & 1 \end{pmatrix}$$

- Mostly we have $\phi > 0$ but sometimes "oscillatory" behaviour consistent with $\phi < 0$.
- Using the identity: $p\left(x_t\right) = \int p\left(x_t | x_{t-1}\right) p\left(x_{t-1}\right) dx_{t-1}$, we can also prove 'time-reversibility':

$$(x_t \mid x_{t-1}) \sim \mathcal{N}(\phi x_{t-1}, \nu), x_{t-1} \sim \mathcal{N}(0, s) \Rightarrow (x_{t-1} \mid x_t) \sim \mathcal{N}(\phi x_t, \nu)$$

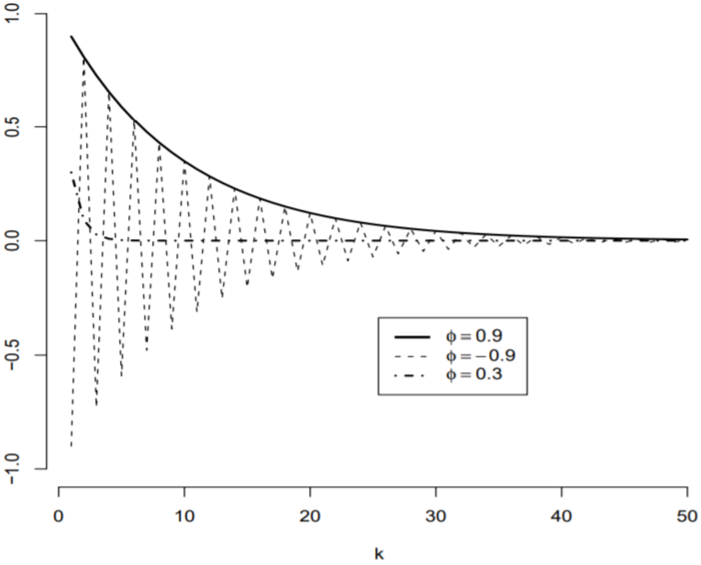

Figure: Auto-correlation functions for AR processes with parameters 0.9, -0.9

Posterior sampling

- In the second half of this talk, we will learn how to fit a stationary AR(1) proccess using `Stan` : a probabilistic programming language.
- Generate some synthetic AR(1) data and fit the reference Bayesian analysis. Sample the posterior distribution and explore histograms of posterior samples for each of ϕ and ν.
- We'll see what proportion of posterior samples of ϕ values fall in the stationary region: $|\phi| < 1$ - tell us if models are consistent with stationarity.
- Fit the Bayesian model to a real data set. Explore posterior histograms, means, etc.

Hidden Markov Models

- Now I will talk about two classes of Hidden Markov Models with AR(1) structure.
- Simplest Hidden Markov Model.
- Stochastic Volatility Model.
- I'll introduce the basic notions and also show the Stan code for fitting a Stochastic Volatility Model.

Simple HMM

- Observed value is now y_t:

$$y_t = x_t + \nu_t,\ \nu_t \sim \mathcal{N}(0, w)$$
$$x_t \leftarrow AR(1|\theta)$$

- Hidden Markov Model: one of the simplest models and yet very important.
- many real processes are not directly observable: measurement error and other forms of technical error, noise obscure the signal.
- In a stationary AR(1) model, $\mathbf{V}(x_t) = s = \nu/(1-\phi^2)$, so $\mathbf{V}(y_t) = s + w$.
- Signal-to-noise ratio: $s/(s+w)$.

- Stochastic Volatility Models (popular tool in quantitative finance).
- Returns on international exchange rate markets, where all the action is in the changes in variance of returns, and it is important to model and capture persistence in variances (volatilities).

$$
\begin{gathered}
y_t \sim \mathcal{N}(0, \sigma_t^2) \\
\sigma_t \sim \exp(\mu + x_t) \\
x_t \sim AR(1 \mid \theta)\ \theta = (\phi, \nu).
\end{gathered}
$$

- Implies that conditional on μ, x_t, $y_t^2 = \sigma_t^2 \kappa_t$, where $\kappa_t \sim \chi_1^2$.
- Take $z_t = \log(y_t^2)/2 = \log(|y_t|)$.

$$
\begin{gathered}
z_t = \mu + x_t + \nu_t, \quad \text{where } \nu_t = \log(\kappa_t)/2 \qquad (1) \\
x_t \sim AR(1 \mid \theta)\ \theta = (\phi, \nu) \qquad (2)
\end{gathered}
$$

- Observed z_t = intercept (defining the baseline volatility on the log scale) + a latent AR(1) process x_t (time-correlated changes in volatility)
- like HMM, but the noise is non-Gaussian - log of a χ_1^2.

Outline for Today's talk

AR(1) Models

Computing Tools

- Nice Markov Chains

MCMC

Convergence

Effective Sample Size

Stan: Building Blocks

Stan: Demonstration

Part II: Introduction to Modern Bayesian Computing Tools

R-Stan for Time Series Analysis

Bayesian Computation

- For a prior $\pi(\theta)$, and likelihood $L(Y^{(n)} \mid \theta)$, the posterior:

$$\pi_n(\boldsymbol{\theta} \mid Y^{(n)}) = \frac{\pi(\boldsymbol{\theta}) L(Y^{(n)} \mid \boldsymbol{\theta})}{\int \pi(\boldsymbol{\theta}) L(Y^{(n)} \mid \boldsymbol{\theta}) d\boldsymbol{\theta}} = \frac{\pi(\boldsymbol{\theta}) L(Y^{(n)} \mid \boldsymbol{\theta})}{L(Y^{(n)})}$$

- The posterior helps us in characterizing the uncertainty in parameters, in predictive summary and any functionals $\psi(\boldsymbol{\theta})$.
- In many interesting models, the posterior is analytically intractable, e.g. when $\boldsymbol{\theta}$ is high-dimensional.
- Approach 1: Posterior approximation
  1. Large sample approximation (Bernstein-von Mises) or Laplace approximation.
  2. Use approximating class $q(\boldsymbol{\theta})$, e.g. exponential family, and minimize discrepancy: variational Bayes, Expectation-Propagation etc.

MCMC

- Posterior approximation does not provide any UQ, and accurate approximations difficult outside limited settings.
- Approach 2: MCMC: sequential algorithm to obtain correlated draws from the posterior.

$$\pi_n(\boldsymbol{\theta} \mid Y^{(n)}) \propto \pi(\boldsymbol{\theta}) L(Y^{(n)} \mid \boldsymbol{\theta})$$

- MCMC: avoids the need to approximate the marginal likelihood $L(Y^{(n)})$. Also more useful than analytic approximation - use samples for posterior quantities of interest and predictive checks.
- How to calculate $\mathbb{E}_{\pi_n}(h(X))$ for arbitrary $h(X)$[1] when π_n is only available up to constants?

[1] measurable

Monte Carlo Integration

- Target posterior distribution: π, Goal: Calculate various posterior quantities.
- Goal: Evaluate the expectation (where π is a density):

$$\mathbb{E}_{\pi}(h(X)) = \int h(x)\pi(x)dx$$

Monte Carlo Integration

- Target posterior distribution: π, Goal: Calculate various posterior quantities.
- Goal: Evaluate the expectation (where π is a density):

$$\mathbb{E}_{\pi}(h(X)) = \int h(x)\pi(x)dx$$

- Idea: If we can draw samples $x^{(1)}, x^{(2)}, \ldots, x^{(N)} \sim \pi(x)$, then we can estimate

$$\mathbb{E}_{\pi}(h(X)) \approx \bar{h}_N = \frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}),\ x^{(t)} \sim \pi(x)$$

Monte Carlo Integration

- Target posterior distribution: π, Goal: Calculate various posterior quantities.
- Goal: Evaluate the expectation (where π is a density):

$$\mathbb{E}_{\pi}(h(X)) = \int h(x)\pi(x)dx$$

- Idea: If we can draw samples $x^{(1)}, x^{(2)}, \ldots, x^{(N)} \sim \pi(x)$, then we can estimate

$$\mathbb{E}_{\pi}(h(X)) \approx \bar{h}_N = \frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}),\ x^{(t)} \sim \pi(x)$$

- This is Monte Carlo Integration.

Motivation

- Useful for any arbitrary expectation (or integration).

Motivation

- Useful for any arbitrary expectation (or integration).
- The strong law of large numbers ensures that this estimate is consistent.

$$\frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}) \overset{\text{a.s.}}{\to} \int h(x)\pi(x)dx \tag{3}$$

Motivation

- Useful for any arbitrary expectation (or integration).
- The strong law of large numbers ensures that this estimate is consistent.

$$\frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}) \overset{\text{a.s.}}{\to} \int h(x)\pi(x)dx \tag{3}$$

- But, **in practice**, independent samples from π might be difficult.

Motivation

- Useful for any arbitrary expectation (or integration).
- The strong law of large numbers ensures that this estimate is consistent.

$$\frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}) \overset{\text{a.s.}}{\to} \int h(x)\pi(x)dx \tag{3}$$

- But, **in practice**, independent samples from π might be difficult.
- Or we might not be able to directly sample from π.

Markov Chains

- Thus it is necessary to relax the condition $x^{(t)} \overset{\text{iid}}{\sim} \pi(x)$

Markov Chains

- Thus it is necessary to relax the condition $x^{(t)} \overset{\text{iid}}{\sim} \pi(x)$
- Instead consider samples $x_1^*, x_2^*, \ldots, x_m^*$ that form a time series: a series of draws from $\pi(\mathbf{x})$ in which $\mathbf{x}_j^*$ may depend on $\mathbf{x}_{j'}^*$ for $j' < j$.

Markov Chains

- Thus it is necessary to relax the condition $x^{(t)} \overset{\text{iid}}{\sim} \pi(x)$
- Instead consider samples $x_1^*, x_2^*, \ldots, x_m^*$ that form a time series: a series of draws from $\pi(\mathbf{x})$ in which $\mathbf{x}_j^*$ may depend on $\mathbf{x}_{j'}^*$ for $j' < j$.
- In the pioneering paper, Metropolis et al. (1953) allowed for serial dependence of the $\mathbf{x}_j^*$ by combining von Neumann's idea of rejection sampling (published in 1951), with concepts from a subject in the theory of stochastic processes called Markov Chains.

Markov Chains

- Thus it is necessary to relax the condition $x^{(t)} \overset{\text{iid}}{\sim} \pi(x)$
- Instead consider samples $x_1^*, x_2^*, \ldots, x_m^*$ that form a time series: a series of draws from $\pi(\mathbf{x})$ in which $\mathbf{x}_j^*$ may depend on $\mathbf{x}_{j'}^*$ for $j' < j$.
- In the pioneering paper, Metropolis et al. (1953) allowed for serial dependence of the $\mathbf{x}_j^*$ by combining von Neumann's idea of rejection sampling (published in 1951), with concepts from a subject in the theory of stochastic processes called Markov Chains.
- Combining Monte Carlo sampling with Markov Chains give rise to the name now used for this technique: **Markov Chain Monte Carlo**.

MCMC

- Recall Monte Carlo Integration / Importance Sampling.

MCMC

- Recall Monte Carlo Integration / Importance Sampling.
- The strong law of large numbers ensures that this estimate is consistent.

$$\frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}) \overset{\text{a.s.}}{\rightarrow} \int h(x)\pi(x)dx \tag{4}$$

MCMC

- Recall Monte Carlo Integration / Importance Sampling.
- The strong law of large numbers ensures that this estimate is consistent.

$$\frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}) \overset{\text{a.s.}}{\rightarrow} \int h(x)\pi(x)dx \tag{4}$$

- It turns out that (4) still applies when the samples are generated from a **stationary Markov chain** !

MCMC

- Recall Monte Carlo Integration / Importance Sampling.
- The strong law of large numbers ensures that this estimate is consistent.

$$\frac{1}{N}\sum_{t=1}^{N} h(x^{(t)}) \overset{\text{a.s.}}{\to} \int h(x)\pi(x)dx \tag{4}$$

- It turns out that (4) still applies when the samples are generated from a **stationary Markov chain** !
- We construct an (ergodic) Markov chain with transition kernel Π chosen to have the same stationary distribution as $\boldsymbol{\pi}$. Then, samples from this Markov chain are samples from $\boldsymbol{\pi}$ if either:
  - We initialize the chain with a draw from π;
  - We run the chain long enough (infinitely long!) so that it converges to $\boldsymbol{\pi}$.

  The first is, again, impossible. Let's look more closely at the second.
- What is a Markov Chain?

Markov Chains

- A Stochastic Process is just a collection of random variables $\{\theta_t^*, t \in \mathrm{T}\}$ for some index set T, usually meant to stand for **time**.

Markov Chains

- A Stochastic Process is just a collection of random variables $\{\theta_t^*, t \in \mathrm{T}\}$ for some index set T, usually meant to stand for **time**.
- In practice T can be either discrete or continuous.

Markov Chains

- A Stochastic Process is just a collection of random variables $\{\theta_t^*, t \in \mathrm{T}\}$ for some index set T, usually meant to stand for **time**.
- In practice T can be either discrete or continuous.
- Intuitively speaking, a Markov chain is a stochastic process unfolding in time in such a way that **the past and future states of the process are independent given the present state**.

Markov Chains

- A Stochastic Process is just a collection of random variables $\{\theta^*_t, t \in \mathrm{T}\}$ for some index set T, usually meant to stand for **time**.
- In practice T can be either discrete or continuous.
- Intuitively speaking, a Markov chain is a stochastic process unfolding in time in such a way that **the past and future states of the process are independent given the present state**.
- More formally, a stochastic process $\{\theta^*_t, t \in \mathrm{T}\}$, $\mathrm{T} = \{0, 1, \ldots\}$ with state space S is a Markov Chain if, for any set $A \in S$,

$$P(\theta^*_{t+1} \in A \mid \theta^*_0, \theta^*_1, \ldots, \theta^*_t) = P(\theta^*_{t+1} \in A \mid \theta^*_t)$$

Example

- Discrete-time Markov chain with continuous-state space:

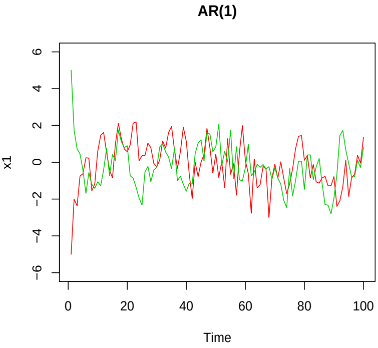

Figure: $\mathrm{AR}(1)$ with different starting points After $5 - 7$ iterations the chains seemed to have forgotten their starting positions.

Example

- Discrete-time Markov chain with continuous-state space:
- $\theta^*_{t+1} \sim \mathcal{N}(0.5 \times \theta^*_t, 1.0)$ $(\mathrm{AR}(1)$ with lag-1 auto-correlation 0.5)

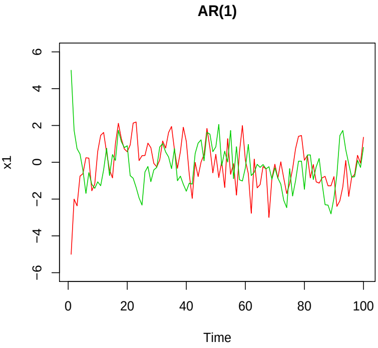

Figure: $\mathrm{AR}(1)$ with different starting points After $5-7$ iterations the chains seemed to have forgotten their starting positions.

Effective Sample Size

Stationary distribution

- At $t \to \infty$, the Markov chain converges to its stationary distribution.

- The stationary distribution does NOT depend on the initial value ($\theta^{(0)}$).
- Similarly the bounded random walk in $[-5, 5]$ converges to the discrete uniform distribution on $[-5, 5]$.
- Does this happen for all Markov chain?

Stationary distribution

- At $t \rightarrow \infty$, the Markov chain converges to its stationary distribution.
- For the auto-regressive process example, this distribution is:

$$\theta^{(t)} \mid \theta^{(0)} \sim \mathcal{N}(0, 1.33) \text{ as } t \rightarrow \infty$$

- The stationary distribution does NOT depend on the initial value ($\theta^{(0)}$).
- Similarly the bounded random walk in $[-5, 5]$ converges to the discrete uniform distribution on $[-5, 5]$.
- Does this happen for all Markov chain?

Nice Behaviour

Yes, if they satisfy three key properties:

- **Irreducibility**
- **Aperiodicity**
- **Positive Recurrence**

Nice Behaviour

Yes, if they satisfy three key properties:

- **Irreducibility**
- *No matter where it starts, the chain has to reach any other state in a finite number of iterations with positive probability.*
- **Aperiodicity**
- **Positive Recurrence**

Nice Behaviour

Yes, if they satisfy three key properties:

- **Irreducibility**
- **Aperiodicity**
- For all states i, the set of all possible **sojourn times**, time to get back to i, can have no divisor bigger than 1. [This is a technical condition, periodic chains are also nice, but aperiodic chains are nicer !]
- **Positive Recurrence**

Nice Behaviour

Yes, if they satisfy three key properties:

- **Irreducibility**
- **Aperiodicity**
- **Positive Recurrence**
- (a) For all states *i*, if the process starts at *i*, it will come back to *i* with probability 1, and (b) the expected length of waiting time till the first return to *i* is finite.

Nice Behaviour

Yes, if they satisfy three key properties:

- **Irreducibility**
- **Aperiodicity**
- **Positive Recurrence**
- The 'nicest' Markov chains have all three properties.

Outline for Today's talk

AR(1) Models

Computing Tools

- Nice Markov Chains

MCMC

Convergence

Effective Sample Size

Stan: Building Blocks

Stan: Demonstration

Ergodicity

- Question: How to calculate $\mathbb{E}_{\pi_n}(h(X))$ for arbitrary $h(X)$ when π_n is only available up to constants?
- Assume the Markov chain has the stationary distribution $\pi(\theta)$, and it's aperiodic and irreducible.
- Then we have an Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \rightarrow \mathbb{E}_{\pi}(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \rightarrow \infty$$

- In plain English: as long as the stationary distribution is $p(\boldsymbol{\theta} \mid \mathbf{y})$, you can learn (to an arbitrary accuracy) about things like posterior mean, and sd and so on just by running a Markov Chain for a long time.

Ergodicity

- Question: How to calculate $\mathbb{E}_{\pi_n}(h(X))$ for arbitrary $h(X)$ when π_n is only available up to constants?
- Assume the Markov chain has the stationary distribution $\pi(\theta)$, and it's aperiodic and irreducible.
- Then we have an Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \to \mathbb{E}_{\pi}(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \to \infty$$

- In plain English: as long as the stationary distribution is $p(\boldsymbol{\theta} \mid \mathbf{y})$, you can learn (to an arbitrary accuracy) about things like posterior mean, and sd and so on just by running a Markov Chain for a long time.
- Also, if $\sigma_h^2 = \mathbf{V}_{\pi}[h(\Theta)] < \infty$, Central Limit Theorem holds and convergence occurs geometrically !

Outline for Today's talk

AR(1) Models

Computing Tools

- Nice Markov Chains

MCMC

Convergence

Effective Sample Size

Stan: Building Blocks

Stan: Demonstration

Waiting for Stationarity

- In plain English: as long as the stationary distribution is $p(\boldsymbol{\theta} \mid \mathbf{y})$, you can learn (to an arbitrary accuracy) about things like posterior mean, and sd and so on just by waiting for stationarity to kick in and monitoring thereafter for a long enough period.

▸ Effective Sample Size

Waiting for Stationarity

- In plain English: as long as the stationary distribution is $p(\boldsymbol{\theta} \mid \mathbf{y})$, you can learn (to an arbitrary accuracy) about things like posterior mean, and sd and so on just by waiting for stationarity to kick in and monitoring thereafter for a long enough period.
- The Ergodic theorem is silent on these issues: **how long you have to wait for stationarity** and **how long to monitor after that?**

▸ Effective Sample Size

Waiting for Stationarity

- In plain English: as long as the stationary distribution is $p(\boldsymbol{\theta} \mid \mathbf{y})$, you can learn (to an arbitrary accuracy) about things like posterior mean, and sd and so on just by waiting for stationarity to kick in and monitoring thereafter for a long enough period.
- The Ergodic theorem is silent on these issues: **how long you have to wait for stationarity** and **how long to monitor after that?**
- A third issue is what to use for the initial value θ_0? Intuitively, the closer θ_0 is to the center of your target $p(\theta)$ the less time you have to wait for stationarity.

▸ Effective Sample Size

Stationarity

- The standard way to deal with waiting for stationarity is to: (A) run the chain from a **good starting value** θ_0^* for B iterations, until **equilibrium** has been reached, and (B) **discard** this initial **burn in** period.
- MCMC diagnostics tries to answer these questions:

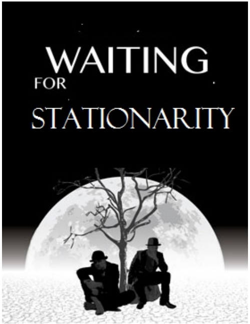

Figure: The Key Question: How long should MCMC run?

Stationarity

- The standard way to deal with waiting for stationarity is to: (A) run the chain from a **good starting value** θ_0^* for B iterations, until **equilibrium** has been reached, and (B) **discard** this initial **burn in** period.
- MCMC diagnostics tries to answer these questions:
  1. What should I use for the initial value θ_0^*?

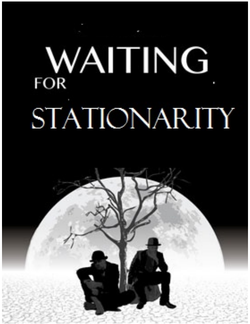

Figure: The Key Question: How long should MCMC run?

Stationarity

- The standard way to deal with waiting for stationarity is to: (A) run the chain from a **good starting value** θ_0^* for B iterations, until **equilibrium** has been reached, and (B) **discard** this initial **burn in** period.
- MCMC diagnostics tries to answer these questions:
  1. What should I use for the initial value θ_0^*?
  2. How do I know when I have reached equilibrium? (How big B should be?)

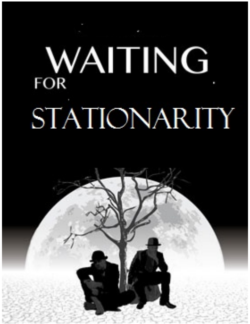

Figure: The Key Question: How long should MCMC run?

Stationarity

- The standard way to deal with waiting for stationarity is to: (A) run the chain from a **good starting value** θ_0^* for B iterations, until **equilibrium** has been reached, and (B) **discard** this initial **burn in** period.
- MCMC diagnostics tries to answer these questions:
  1. What should I use for the initial value θ_0^*?
  2. How do I know when I have reached equilibrium? (How big B should be?)
  3. Once I've reached equilibrium, how long should I monitor? (How big M should be?)

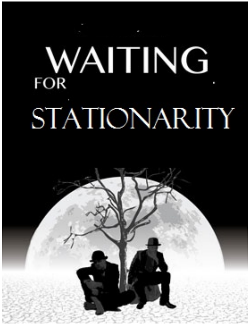

Figure: The Key Question: How long should MCMC run?

Numerical Standard Errors

- Remember the Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \to \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \to \infty$$

Numerical Standard Errors

- Remember the Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \rightarrow \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \rightarrow \infty$$

- We also said that if $\sigma_h^2 = \mathbf{V}_\pi(h(\theta))$ is finite, we can show that $\bar{h}_N$ will follow a Normal distribution with mean $\mathbb{E}_\pi(h(\theta))$. (Stronger the strong law of large numbers)

Numerical Standard Errors

- Remember the Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \to \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \to \infty$$

- We also said that if $\sigma_h^2 = \mathbf{V}_\pi(h(\theta))$ is finite, we can show that $\bar{h}_N$ will follow a Normal distribution with mean $\mathbb{E}_\pi(h(\theta))$. (Stronger the strong law of large numbers)
- We define numerical standard error of $\bar{h}_N$ as $\sqrt{\mathbf{V}_\pi(\bar{h}_N)}$, and for large N:

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{1 + 2\sum_{l=1}^{N-1}\rho_l(h)\right\}}$$

where $\rho_l(h)$ is the lag-l auto-correlation in $h(\theta^{(t)})$.

Numerical Standard Errors

- Remember the Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \to \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \to \infty$$

- We also said that if $\sigma_h^2 = \mathbf{V}_\pi(h(\theta))$ is finite, we can show that $\bar{h}_N$ will follow a Normal distribution with mean $\mathbb{E}_\pi(h(\theta))$. (Stronger the strong law of large numbers)
- We define numerical standard error of $\bar{h}_N$ as $\sqrt{\mathbf{V}_\pi(\bar{h}_N)}$, and for large N:

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{1 + 2\sum_{l=1}^{N-1}\rho_l(h)\right\}}$$

where $\rho_l(h)$ is the lag-l auto-correlation in $h(\theta^{(t)})$.

- In general no simpler expression exist for the nse.

Numerical Standard Errors

- Remember the Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \to \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \to \infty$$

- We also said that if $\sigma_h^2 = \mathbf{V}_\pi(h(\theta))$ is finite, we can show that $\bar{h}_N$ will follow a Normal distribution with mean $\mathbb{E}_\pi(h(\theta))$. (Stronger the strong law of large numbers)
- We define numerical standard error of $\bar{h}_N$ as $\sqrt{\mathbf{V}_\pi(\bar{h}_N)}$, and for large N:

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{1 + 2\sum_{l=1}^{N-1}\rho_l(h)\right\}}$$

  where $\rho_l(h)$ is the lag-l auto-correlation in $h(\theta^{(t)})$.
- In general no simpler expression exist for the nse.
- Many references: Geyer (1992), Besag and Green (1993) for ideas.

Charlie Geyer's Advice

- First rule of MCMC: compute standard errors. If you don't care how accurate your MCMC estimates are, then why should we take you seriously?
- Second rule of MCMC: variance estimation is not "diagnostic". If chain doesn't converge, then variance estimation is GIGO (garbage in, garbage out).

Multiple Chains: Gelman-Rubin's $\hat{R}$ [2]

- Most approaches for detecting convergence, both formal and informal, rest on the idea of starting multiple Markov chains and observing whether they come together and start to behave similarly (if they do, we can pool the results from each chain).
- It is typically recommended (e.g., Gelman and Rubin, 1992) to use overdispersed initial values, meaning "more variable than the target distribution" i.e., the posterior.

[2] Patrick Breheny's Notes

Multiple Chains: Gelman-Rubin's $\hat{R}$ [3]

- Although looking at trace plots is certainly useful, it is also desirable to obtain an objective, quantifiable measure of convergence.
- The basic idea is to quantify the between-chain and the within-chain variability of a quantity of interest if the chains have converged, these measures will be similar; otherwise, the between-chain variability will be larger.

[3] Patrick Breheny's Notes

Multiple Chains: Gelman-Rubin's $\hat{R}$ [4]

- The basic idea of the estimator is as follows (the actual estimator makes a number of modifications to account for degrees of freedom):
  - Let B denote the standard deviation of the pooled sample of all MT iterations (the between-chain variability).
  - Let W denote the average of the within-chain standard deviations
  - Quantify convergence with:

$$\hat{R} = \frac{B}{W}$$

- If $\hat{R} \gg 1$, this is clear evidence that the chains have not converged
- As $T \to \infty$, $\hat{R} \to 1$; $\hat{R} < 1.05$ is widely accepted as implying convergence for practical purposes.

[4] Patrick Breheny's Notes

Auto-correlation

Auto-correlation: $\rho_l(h) = \text{Correlation}\left(h(\theta^{(t)}), h(\theta^{(t+l)})\right)$.

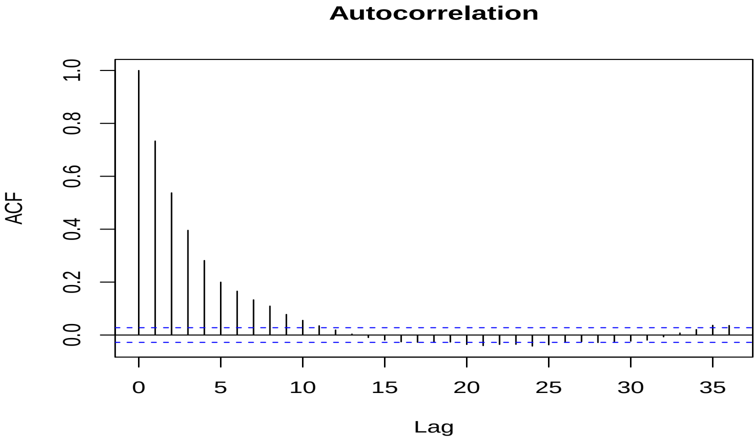

Figure: ACF for the Normal-Cauchy Example.

Numerical Standard Errors: Price of Markov Chain

- If $h(\theta^{(t)})$ can be approximated as a first order auto-regressive process, then

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{\frac{1+\rho}{1-\rho}\right\}}$$

- where ρ is the lag-1 auto-correlation in $h(\theta^{(t)})$.

[5] finite if the chain converges geometrically

Numerical Standard Errors: Price of Markov Chain

- If $h(\theta^{(t)})$ can be approximated as a first order auto-regressive process, then

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{\frac{1+\rho}{1-\rho}\right\}}$$

- where ρ is the lag-1 auto-correlation in $h(\theta^{(t)})$.
- The first factor is the usual term under **independent sampling**.

[5] finite if the chain converges geometrically

Numerical Standard Errors: Price of Markov Chain

- If $h(\theta^{(t)})$ can be approximated as a first order auto-regressive process, then

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{\frac{1+\rho}{1-\rho}\right\}}$$

- where ρ is the lag-1 auto-correlation in $h(\theta^{(t)})$.
- The first factor is the usual term under **independent sampling**.
- The second term is usually **greater than 1**.

[5] finite if the chain converges geometrically

Numerical Standard Errors: Price of Markov Chain

- If $h(\theta^{(t)})$ can be approximated as a first order auto-regressive process, then

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{\frac{1+\rho}{1-\rho}\right\}}$$

- where ρ is the lag-1 auto-correlation in $h(\theta^{(t)})$.
- The first factor is the usual term under **independent sampling**.
- The second term is usually **greater than 1**.
- Penalty to be paid for using a Markov chain.

[5] finite if the chain converges geometrically

Numerical Standard Errors: Price of Markov Chain

- If $h(\theta^{(t)})$ can be approximated as a first order auto-regressive process, then

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{\frac{1+\rho}{1-\rho}\right\}}$$

- where ρ is the lag-1 auto-correlation in $h(\theta^{(t)})$.
- The first factor is the usual term under **independent sampling**.
- The second term is usually **greater than 1**.
- Penalty to be paid for using a Markov chain.
- Moreover, the nse may not be finite in general [5].

[5] finite if the chain converges geometrically

Numerical Standard Errors: Price of Markov Chain

- If $h(\theta^{(t)})$ can be approximated as a first order auto-regressive process, then

$$\text{nse}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{\frac{1+\rho}{1-\rho}\right\}}$$

- where ρ is the lag-1 auto-correlation in $h(\theta^{(t)})$.
- The first factor is the usual term under **independent sampling**.
- The second term is usually **greater than 1**.
- Penalty to be paid for using a Markov chain.
- Moreover, the nse may not be finite in general [5].
- If the nse is finite, then we can make it as small as we like by increasing N. (Long chain!)

[5] finite if the chain converges geometrically

Outline for Today's talk

AR(1) Models

Computing Tools

- Nice Markov Chains

MCMC

Convergence

Effective Sample Size

Stan: Building Blocks

Stan: Demonstration

Effective Sample Size

- **Key question:** How many samples from π do we need?
- 'It depends on the problem' ... for now, suppose you want to calculate the posterior mean.
- Two possible sources of variance:
  - The inherent variance of the posterior (source of UQ)
  - Additional variance from approximating an integral by a summation ("Monte Carlo error")
- The first is irreducible error but the second can be reduced with more samples.

Effective Sample Size

- Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N} \sum_{i=1}^{N} h(\theta^{(t)}) \rightarrow \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \rightarrow \infty$$

Effective Sample Size

- Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \to \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \to \infty$$

- Numerical standard error of $\bar{h}_N$ as $\sqrt{\mathbf{V}_\pi(\bar{h}_N)}$, and for large N:

$$\text{NSE}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{1 + 2\sum_{l=1}^{N-1}\rho_l(h)\right\}} \doteq \sqrt{\frac{\sigma_h^2}{N_{ess}}}$$

where $\rho_l(h)$ is the lag-l auto-correlation in $h(\theta^{(t)})$.

Effective Sample Size

- Ergodic Theorem:

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(t)}) \to \mathbb{E}_\pi(h(\theta)) = \int h(\theta)\pi(\theta)d\theta, \text{ as } N \to \infty$$

- Numerical standard error of $\bar{h}_N$ as $\sqrt{\mathbf{V}_\pi(\bar{h}_N)}$, and for large N:

$$\text{NSE}(\bar{h}_N) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{1 + 2\sum_{l=1}^{N-1}\rho_l(h)\right\}} \doteq \sqrt{\frac{\sigma_h^2}{N_{ess}}}$$

where $\rho_l(h)$ is the lag-l auto-correlation in $h(\theta^{(t)})$.

- Effective sample size:

$$N_{ess} = \frac{N}{\left\{1 + 2\sum_{l=1}^{N-1}\rho_l(h)\right\}}$$

Effective Sample Size

- Since ESS controls the quality of our inference, ESS/second is the appropriate metric for comparing samplers.
- Different choices of the Markov transition kernel Π can give radically different ESS/second.
- Standard Metropolis-Hastings or Gibbs Samplers struggle for complex (hierarchical) and high-dimensional (many parameters) models.
- Hamiltonian Monte Carlo performs much more efficiently for a wide range of problems.

Hamiltonian Monte Carlo

- In its default mode, Stan uses a technique known as Hamiltonian Monte Carlo to sample from the posterior with minimal autocorrelation.
- These draws are typically more expensive than from other methods, e.g. Gibbs samplers, but the reduced correlation leads to a (much) higher ESS/second.
- Very roughly: Metropolis-Hastings methods move around the probability space randomly (without knowledge of the underlying geometry) and use a accept-reject step to adjust probabilities accordingly.
- Hamiltonian Monte Carlo gives a particle a random "kick" and samples based on the path of the particle: uses Hamiltonian mechanics to simulate the path of the particle in an energy field induced by the target density π.

Outline for Today's talk

- AR(1) Models
- Computing Tools
    - Nice Markov Chains
- MCMC
- Convergence
- Effective Sample Size
- Stan: Building Blocks
- Stan: Demonstration

Probabilistic Programming Language

- “A probabilistic programming language implementing full Bayesian statistical inference with MCMC sampling (NUTS, HMC) and penalized maximum likelihood estimation with Optimization (L-BFGS)”
- “black-box” MCMC samplers: BUGS, JAGS, Church, PyMC etc.
- Unlike BUGS and JAGS, not restricted to Gibbs sampling or conjugate (exponential family graphical) models.
- Stan is open source
- Built to be fast (about 10 times faster then BUGS according Gelman)
  Named after Stanislaw Ulam, co-inventor of Monte Carlo method.

Stan does ..

- Full Bayesian Inference (via Hamiltonian Monte Carlo)
- Variational Bayesian Inference (via ADVI)
- Penalized MLE (Bayesian MAP)
- Best thought of as a DSL for specifying a distribution and sampling from it.

Stan: Building Blocks

Stan provides a wide range of built-in data types:

- Data primitives: `real`, `int`
- Mathematical structures: `vector`, `matrix` can hold `real` and `int`
- Programming structures: `array` can hold any other `Stan` type
- Constrained structures: `ordered`, `positive_ordered`, `simplex`, `unit_vector`
- Matrix types: `cov_matrix`, `corr_matrix`, `cholesky_factor_cov`, `cholesky_factor_corr`

Building Blocks

A Stan model is defined by the following five programming blocks:

```
data

transformed data

parameters \\(required)

transformed parameters

model \\(required)

generated quantities
```

Building Blocks

- `data`: Defines the external data which Stan will read at the beginning of execution
- `parameters`: Defines the variables which will be inferred
- `model`: Defines the probability model relating the data and parameters. Both the prior and the likelihood are coded in this block
- Additional blocks, e.g., `transformed data`, `generated quantities` are useful for performing additional transformations within Stan.

Additional Blocks

- `transformed data` block allows for preprocessing of data.
- `transformed parameter` block allows for parameter transformation before the posterior is computed.
- `generated quantities` allows for post-processing the posterior samples.

Outline for Today's talk

AR(1) Models

Computing Tools

Nice Markov Chains

MCMC

Convergence

Effective Sample Size

Stan: Building Blocks

Stan: Demonstration

First Example

Mean only, Normal 'shocks' $y_t \sim \mathcal{N}(\theta, \sigma^2)$, $t = 1, \ldots, T$.

```
data {
    int<lower=1> T; //error checking for T
    real y[T];
}
parameters {
    real theta;
    real<lower=0> sigma;
}
model {
    y ~ normal(theta, sigma); // vectorized
}
```

Let’s look at this code in Stan