Bayesian Workflow: Lecture 2

STAT 447

Caden Hewlett

2024-03-14

MCMC Diagnostics.

The notion of "mixing," as in "how quickly do we forget the initial distribution?" We have seen MCMC is consistent; however the speed of convergence can vary considerably due to the dependence between the successive draws.

Recall if the state space of chian $X^{(m)}$, Birkhoff's Ergodic Theorem applies:

$$\frac{1}{M}\sum_{m=1}^{M} g(X^{(m)}) \overset{\infty}{\to} \mathbb{E}_\pi\left[g(X)\right]$$

Fast Mixing

In this situation, the chain is almost *iid* just a **constant time** slower. The technical name is "geometric ergodicity," which is a property of Markov Chains. Fast mixing happens when the dependence between time step i and $i + m$ decays exponentially in m. Recall, a Markov Chain is ergodic if it is aperiodic and irreducible.

Slow/Torpid Mixing

The MCMC is still consistent, but you may have to wait a long time for the answer! In this case, we need changes.

How to detect slow-mixing chains

Use several "over-dispersed" chains. *Over-Dispersed*: use at least as much noise as the prior (roughly.) Then, check for differences between the independent chains via rank plots, trace plots, etc.

Worked Example

Fast-mixing: a beta-binomial problem.

```
data {
  int<lower=0> n_trials;
  int<lower=0> n_successes;
}
parameters {
  real<lower=0, upper=1> p;
}

model {
  p ~ uniform(0, 1);
  n_successes ~ binomial(n_trials, p);
}
```

Slow-mixing: binomial likelihood, but with “too many parameters.” As you can see p_1 and p_2 are inextricably linked and cannot be easily separated or distinguished from one another.

```
data {
  int<lower=0> n_trials;
  int<lower=0> n_successes;
}
parameters {
  real<lower=0, upper=1> p1;
  real<lower=0, upper=1> p2;
}

model {
  p1 ~ uniform(0, 1);
  p2 ~ uniform(0, 1);
  n_successes ~ binomial(n_trials, p1*p2);
}
```

Then, the MCMC process is here.

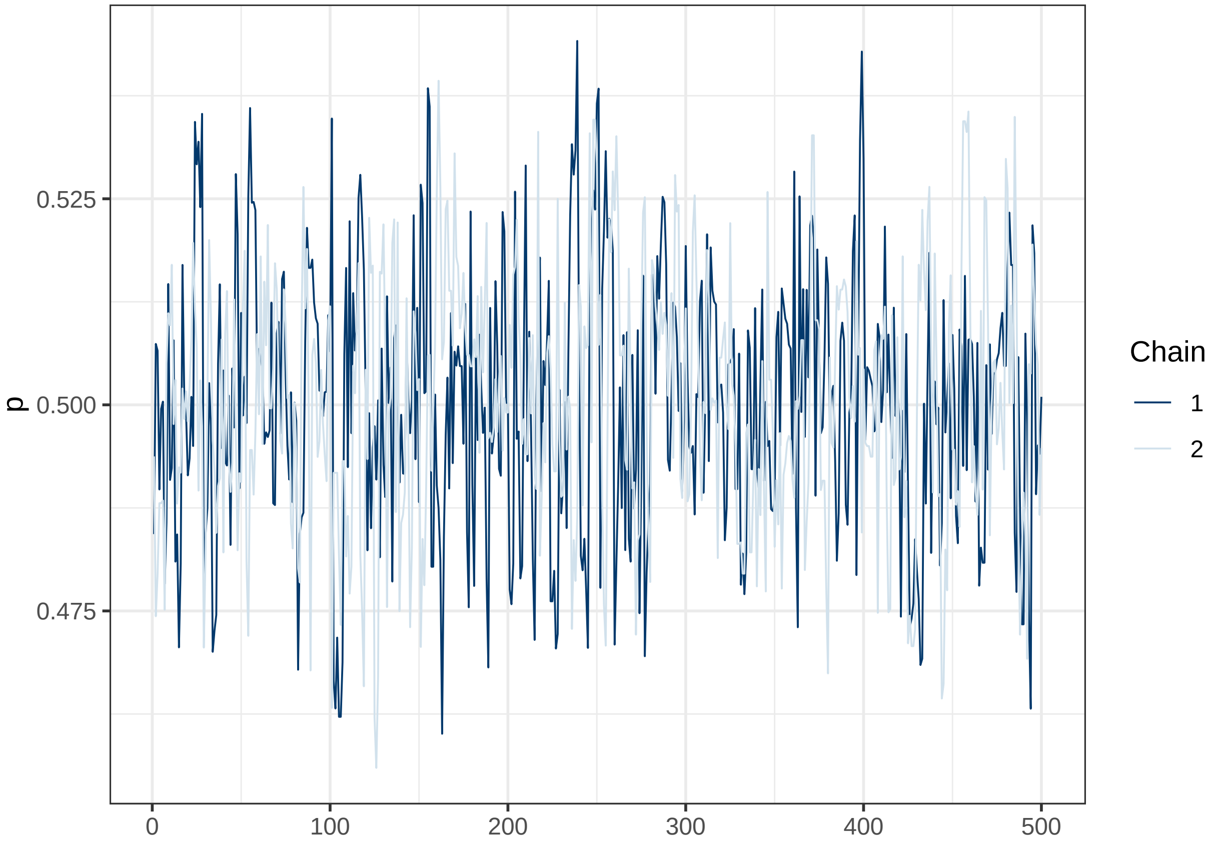
```
M = 1000
fit_easy = sampling(
  easy,
  seed = 1,
  chains = 2,
  refresh = 0,
  data = list(n_trials=M, n_successes=M/2),
  iter = 1000
)

fit_hard = sampling(
  unidentifiable,
  seed = 1,
  chains = 2,
  refresh = 0,
  data = list(n_trials=M, n_successes=M/2),
  iter = 1000
)
```

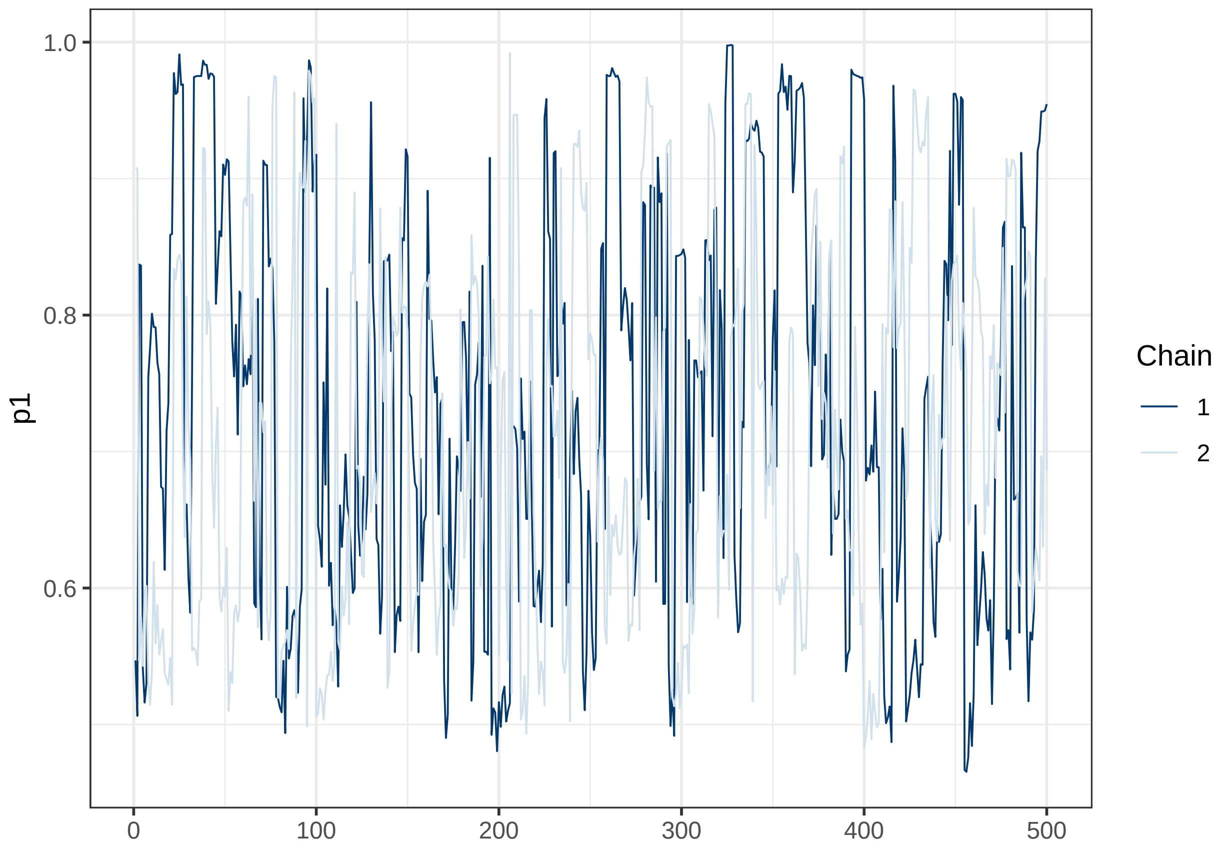
Then, we can run diagnostics!

Trace Plots

```
mcmc_trace(fit_easy, pars = c("p")) + theme_bw()
```



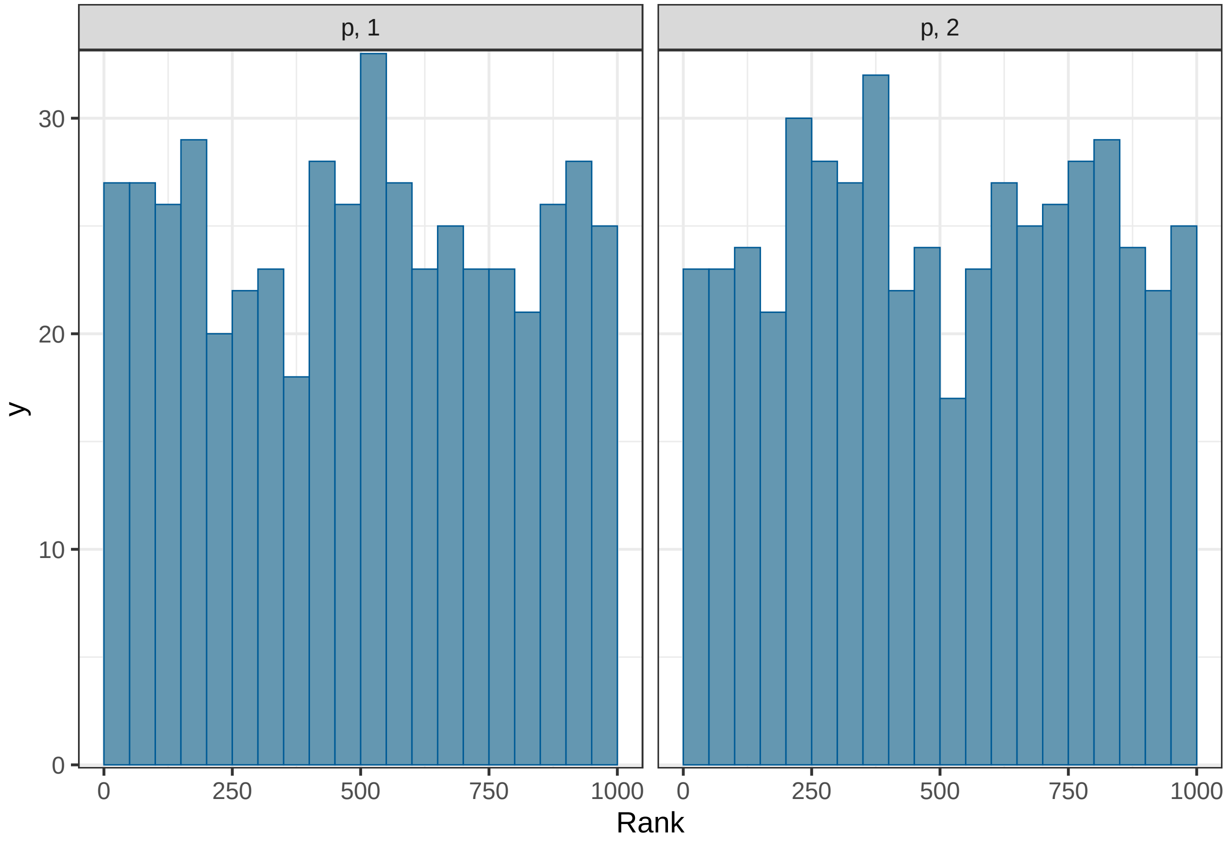
```
mcmc_trace(fit_hard, pars = c("p1")) + theme_bw()
```



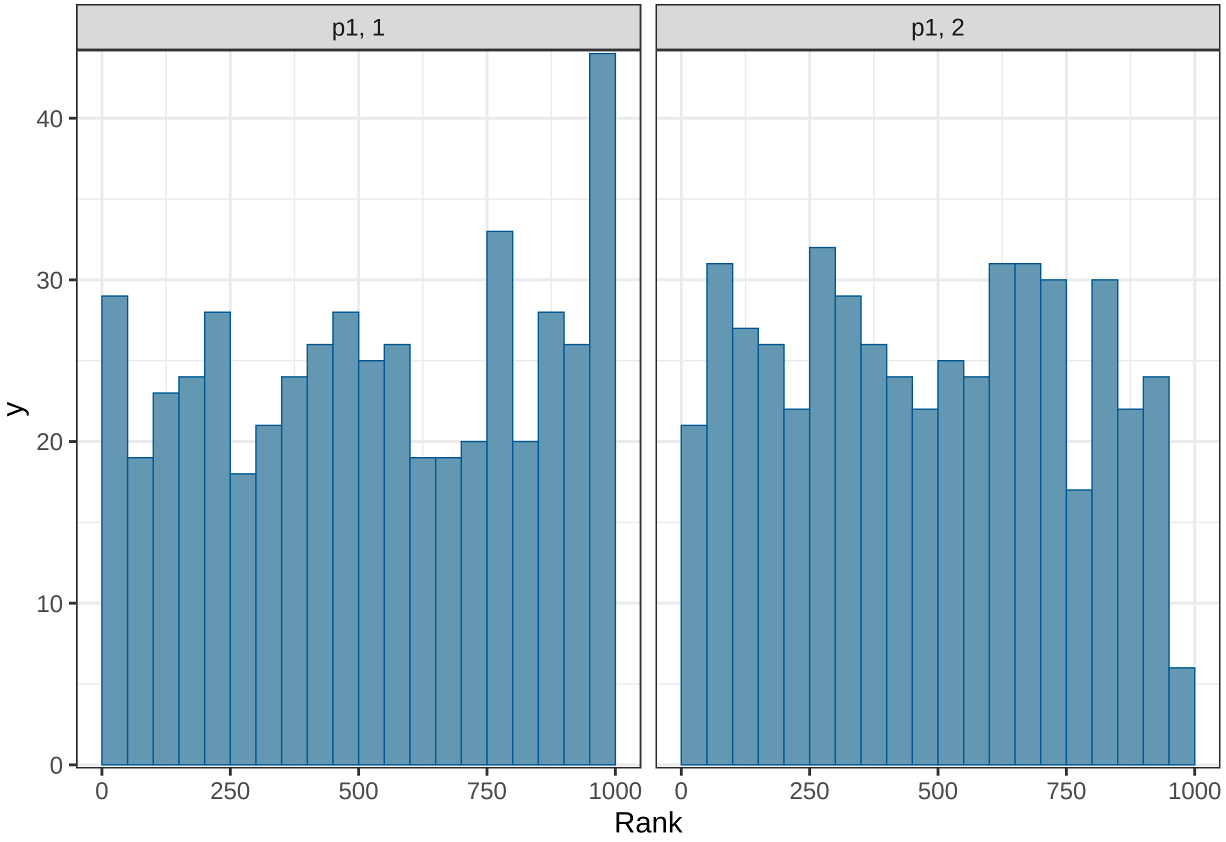
Rank Histograms

In a rank histogram, there will be an obvious dispersion of ranks between the parameters. In the fast mixing case, all histograms will be roughly equivalent, but in a slow case, there will be a noteable difference between them.

```
mcmc_rank_hist(fit_easy, pars = c("p")) + theme_bw()
```



```
mcmc_rank_hist(fit_hard, pars = c("p1")) + theme_bw()
```



Effective Sample Size

Now, we consider Markov Chain Standard Error, Effetive Sample Size and teh CLT for Markov Chains (in the fast mixing case.) Once we've concluded the chain is mixing well, the quesiton is: how many digits aare reliable when reporting an MCMC approximation?

There are two types of errors Bayesian analysis based on Monte Carlo.

Statistical Error: inherent uncertainty due to, for example, the fact we have finite data, we make assumptions, we have unidentifiability, etc.

Computation Error: additional error due to the fact that we use an approximation of the posterior instead of the exact posterior.

We discuss Computational Error below. It's actually done via **normal approximation**!

Ok, but why are we okay with this? When building a credible interval (a Bayesian method of *statistical error*), we avoided Central Limit Theorems. In short, in MCMC iterations it is really easy to increase the number of iterations M compared to the number of data points n. So, there's no cost really with using Frequentist reasoning.

(Also, as an extra, some approaches use the Laplace approximation instead of MCMC, which is "CLT-like" processes motivated by the Bernstein-von Mises theorem.)

Executive Version

How many digits are reliable?

Suppose you are approximating a posterior mean in Stan. We now show how to determine how many digits are reliable:

1. Print the `fit` object
2. Roughly twice ($Z_{0.975} \approx 1.96$) the column `se_mean` provides a 95% confidence interval.

```
data {
  real y;
}

parameters {
  real<lower=0, upper=5> x;
}

model {
  x ~ uniform(0, 5);
  y ~ uniform(0, x);
}
```

Then, the MCMC Sampling procedure:

```
fit = sampling(
  doomsday,
  data = list(y = 0.06),
  chains = 1,
  seed = 1,
  refresh = 0,
  iter = 20000
)
```

With the `fit` given by…

```
information = extract(fit)
fit
```

```
## Inference for Stan model: anon_model.
## 1 chains, each with iter=20000; warmup=10000; thin=1;
## post-warmup draws per chain=10000, total post-warmup draws=10000.
##
##       mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## x     1.16    0.03 1.29  0.07  0.18  0.57  1.76  4.56  1624    1
## lp__ -0.39    0.02 0.68 -2.44 -0.43 -0.12 -0.04 -0.01  1580    1
##
## Samples were drawn using NUTS(diag_e) at Thu Mar 14 10:56:28 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Results and Computational Error Interval

We observed `se_mean = 0.03` and `mean = 1.16`. The true value was 1.117

The 95% confidence interval is then…

$$\mathrm{CI}_{\mathrm{err}}(\alpha) = \bar{x}_M \pm Z_{1-\alpha/2}\,\mathrm{se}(\bar{x}_M)$$

Where…

$$\bar{X} = M^{-1}\sum_{m=1}^{M} X^{(m)}$$

In our case, this is

$$\mathrm{CI}_{\mathrm{err}}(0.95) = [1.10, 1.22]. \text{ Note, } 1.117 \in \mathrm{CI}_{\mathrm{err}}(0.95).$$

We expect the interval to contain the truth in approximately 95% of seeds.

Mathematical Underpinnings

We answer two important questions.

Firstly, how can we compute Monte Carlo Standard Errors? (MCSE) We saw these already as `se_mean`, but like how are they found?

What underlying theory justifies that computation?

Along the way, we define the notion of Effective Sample Size.

Background

Recall the entral limit theorem for *iid* rvs.

Let $V_1, V_2 \ldots V_n$ be *iid* with finite variance. Then,

$$\sqrt{n}(\bar{V} - \mathbb{E}(V)) \to N\Big(0, \sqrt{\mathrm{var}(V)}\,\Big)$$

Where $\mathbb{E}(V) = \mu$.

From the CLT, recall the standard frequentist argument gives:

$$\mathbb{P}\Big(\mu \in \big[\bar{V} \pm Z_{\alpha/2}\mathrm{se}(V)\big]\Big) = 1 - \alpha$$

For Markov Chains

We need this to generalize for Markov Chains!

Markov Chain

Let $X^{(1)}, X^{(2)}, \ldots X^{(N)}$ be random variables in a Markov Chain, such that they satisfy Markov's Property: $\mathbb{P}(X^{(n+1)} = x \mid X^{(n)}, X^{(n-1)} \ldots X^{(1)}) = \mathbb{P}(X^{(n+1)} = x \mid X^{(n)})$.

Let $\pi(x) = \gamma(x)/Z$. Let $\mu = \int x\pi(x)\mathrm{d}x$.

Theorem

Assuming $\sigma^2 = \int (x-\mu)^2 \pi(x) \mathrm{d}x < \infty$, in our context that the posterior variance is finite, under fast mixing assumptions we know that:

$$\sqrt{n}(\bar{X} - \mu) \to N(0, \sigma_a)$$

Where the constant $\sigma_a^2 \in \mathbb{R}^+ \setminus \{0\}$ is known as the asymptotic variance. This also requires the reversibility of Markov Chains, too.

Effective Sample Size

The ESS is an answer to the following: How many *iid* samples n_e would be equivalent (same variance) to the M samples obtained from MCMC? By default, Stan is run for $2,000$ iterations, but this is not equivalent to *iid* sampling. For example, if $n_e = 1,000$, then you're roughly two times slower than MCMC.

Note, $\mathbb{V}(aX + b) = a^2 \mathbb{V}(X)$. So, using...

$$\sqrt{n_e}(\bar{X}_{\text{iid}} - \mu) \to N(0, \sigma)$$
$$\sqrt{M}(\bar{X}_{\text{MCMC}} - \mu) \to N(0, \sigma_a)$$

So...

$$\text{var}\Big(\sqrt{n_e}(\bar{X}_{\text{iid}} - \mu)\Big) \approx \sigma^2$$
$$\text{var}\Big(\sqrt{M}(\bar{X}_{\text{MCMC}} - \mu)\Big) \approx \sigma_a^2$$

Implying that:

$$n_e \text{var}(\bar{X}_{\text{iid}}) \approx \sigma^2$$
$$M \text{var}(\bar{X}_{\text{MCMC}}) \approx \sigma_a^2$$

Then, if we have n_e defined such that $\text{var}(\bar{X}_{\text{iid}}) = \text{var}(\bar{X}_{\text{MCMC}})$, then we have a closd form for the effective sample size!

$$\text{Effective Sample Size} = n_e = M \frac{\sigma^2}{\sigma_a^2}$$

Sometimes, the effective sample size is greater than the actual sample size!

Such an example is known as "super-efficient" Markov Chains.

What is the Asymptotic Variance?

We can estimate σ_a^2 in many different ways.

We start here with the simplest possible setting. Let C be the number of chains and S be the number of MCMC samples. We then have $\{E_1, E_2, \dots E_C\}$ from the estimators. We also then have the voerall estimator.

$$E = (C)^{-1} \sum_{c=1}^{C} E_c$$

Also, we know that $\forall c \in [1, C]$

$$\forall c \in [1, C] \, S \cdot \text{var}[E_c] \approx \sigma_a^2$$

Finally, by *iid*,

$$\mathrm{var}[E_c] \approx \frac{1}{C}\sum_{c=1}^{C}(E_c - E)^2$$

Thus, we can denote the estimator for σ_a^2 as…

$$\sigma_a^2 \approx \frac{S}{C} \cdot \sum_{c=1}^{C}(E_c - E)^2$$

The "one long chain" is a common approach is to view a trace of length M as C subsequent batches of length S