MATH 640 Note Set II

Mark J. Meyer

February 28, 2019

GEORGETOWN UNIVERSITY

Georgetown College

Department of Mathematics and Statistics

Course Organization

Topics are organized into Three parts:

- Introduction to Bayesian Thought
  - Note Set I, ∼ BDA Chapter 1 and Appendix A
- Part I: Bayesian Theory and Direct Sampling
  - Note Set I, ∼ BDA Chapters 2—5, 14, 16
- Part II: Bayesian Analysis and Computation
  - Note Set II, ∼ BDA Chapters 6, 7, 10—12, 14—16

Course Organization

Topics are organized into Three parts:

- Introduction to Bayesian Thought
  - Note Set I, ∼ BDA Chapter 1 and Appendix A
- Part I: Bayesian Theory and Direct Sampling
  - Note Set I, ∼ BDA Chapters 2—5, 14, 16
- Part II: Bayesian Analysis and Computation
  - Note Set II, ∼ BDA Chapters 6, 7, 10—12, 14—16

Part II: Bayesian Analysis and Computation

Chapters 6, 7, 10—12, 14—16

Part II Units

A. Model Checking, Evaluating, and Comparing

B. Introduction to Bayesian Computation

C. Markov Chain Monte Carlo Methods

D. Efficient Markov Chain Simulation

E. Additional Topics in Regression

Unit A: Model Checking, Evaluating, and Comparing

Chapters 6 and 7

II.A Model Checking, Evaluating, and Comparing

Checking the model is a crucial step in any statistical analysis

In the Bayesian context, we should at least check

- The adequacy of the fit of the model to the data
- The plausibility of the model for the purposes for which it will be used

Often cast as sensitivity to the prior, but the likelihood is can also be suspect

II.A Model Checking, Evaluating, and Comparing

Sensitivity Analysis

The basic question: “how much do posterior inferences change when other reasonable probability models are used in place of the present model?”

We should consider other plausible models for the data to see if choice of prior and likelihood have an impact on our posterior inference

II.A Model Checking, Evaluating, and Comparing

Ideal Analysis

In theory, model checking and sensitivity analysis can be incorporated into our usual analysis in the form of a mixture model of all possible ‘true’ models

Example II.1

Suppose y has a symmetric and unimodal shape to its distribution. There are many plausible models for this data: normal, t, Laplacian, Cauchy (to name a few).

II.A Model Checking, Evaluating, and Comparing

Example II.1 (cont.)

Thus y might have four plausible likelihoods with different parameters:

$$\mathcal{L}_1(y|\mu_n, \sigma^2_n), \mathcal{L}_2(y|\mu_t, \sigma^2_t), \mathcal{L}_3(y|\theta, \gamma), \text{ and } \mathcal{L}_4(\mu_l, \sigma_l)$$

each likelihood would then have it's own prior and we could set up the following mixture posterior:

$$\begin{aligned} &\propto \lambda_1\mathcal{L}_1(y|\mu_n, \sigma^2_n)\pi_1(\mu_n, \sigma^2_n) + \lambda_2\mathcal{L}_2(y|\mu_t, \sigma^2_t)\pi_2(\mu_t, \sigma^2_t) \\ &+ \lambda_3\mathcal{L}_3(y|\theta, \gamma)\pi_3(\theta, \gamma) + \lambda_4\mathcal{L}_4(\mu_l, \sigma_l)\pi_4(\mu_l, \sigma_l) \end{aligned}$$

where $\sum_i \lambda_i = 1$

Example II.1 (cont.)

But this only covers the model checking part! We'd ideally want to build into our sensitivity analysis different priors as well so we incorporate all possible prior beliefs—ideally using a mixture prior for each π_i

Such a model is called an *exhaustive* probability model as it automatically incorporates all 'sensitivity analyses' but is still predicated on the truth of some member of the larger class of models

II.A Model Checking, Evaluating, and Comparing

In practice, setting up such a model to include all possibilities and substantive knowledge is conceptually impossible and computationally infeasible.

The goal then is not to ask ‘is our model true or false’ but to ask ‘do the model’s deficiencies have a noticeable effect on the substantive inferences?’

Let’s consider some basics of model checking

II.A Model Checking, Evaluating, and Comparing

External Validation

Formally, we can use our current model to make predictions about future data, and then collect those data and compare to their predictions.

Potentially the most obvious form of model checking: either we go and collect more data to validate or find a similar data set

Often we need to check our model before obtaining new data, so we will discuss ways to approximate external validation using the data at hand

II.A Model Checking, Evaluating, and Comparing

Posterior Predictive Checking

If the model fits, then replicated data generated under the model should look similar to the observed data, i.e. the observed data should look plausible under the posterior predictive distribution

Basic technique for checking model fit:

- Draw simulated values from the joint posterior predictive distribution of replicated data
- Compare these samples to the observed data

II.A Model Checking, Evaluating, and Comparing

Let y^{rep} denote the *replicated* data that *could have been* observed—we distinguish this from $\tilde{y}$ which represents *new* or *future* data that has yet to be collected, y^{rep} is a replication of y

Given the current state of knowledge, the distribution of y^{rep} is

$$p(y^{\text{rep}}|y) = \int p(y^{\text{rep}}|\theta)p(\theta|y)d\theta$$

In this case, if y has explanatory variables x, y^{rep} will also have explanatory variables x

The easiest approach would be to draw y^{rep} from $p(y^{\text{rep}}|\hat{\theta})$

II.A Model Checking, Evaluating, and Comparing

Let $T(y, \theta)$ denote the *test quantity* which is a scalar summary of parameters and data used as the standard when comparing data to predictive simulations

In contrast, $T(y)$ is a *test statistics* which is a test quantity that only depends on the the data

We can use these measures to summarize discrepancies between the model and data

II.A Model Checking, Evaluating, and Comparing

Tail-area probabilities

Lack of fit of the data with respect to the posterior predictive distribution can be measured by the tail-area probability, or p-value of a test quantity computed using posterior simulations of (θ, y^{rep})

We take two approaches:

1. Classical p-values
2. Posterior predictive p-values

II.A Model Checking, Evaluating, and Comparing

Classical p-values

The mathematical definition of the classical p-value for the test statistic $T(y)$ is

$$p_C = Pr\left[T(y^{\text{rep}}) \geqslant T(y)|\theta\right]$$

where the probability is taken over the distribution of y^{rep} with fixed θ.

Classical p-values generally represent a summary measure of discrepancy between the observed data and what would be expected under a model with a particular value of θ

II.A Model Checking, Evaluating, and Comparing

Posterior Predictive p-values

In the Bayesian approach, test quantities can be functions of unknown parameters as well as the data (in contrast to the classical approach):

$$p_B = Pr[\mathsf{T}(y^{\mathsf{rep}}, \theta) \geqslant \mathsf{T}(y, \theta)|y]$$

where the probability is taken over the posterior distribution of θ and the posterior predictive distribution of y^{rep}, i.e.

$$p_B = \int\int I\{\mathsf{T}(y^{\mathsf{rep}}, \theta) \geqslant \mathsf{T}(y, \theta)\} p(y^{\mathsf{rep}}|\theta) p(\theta|y) dy^{\mathsf{rep}} d\theta$$

II.A Model Checking, Evaluating, and Comparing

Drawing from the Posterior Predictive Distribution

If we already have B simulations from the posterior density of θ, we draw one y^{rep} from the posterior predictive distribution for each simulated θ which gives us B draws from the joint posterior distribution of $p(y^{\text{rep}}, \theta|y)$

The posterior predictive check then compares $T(y, \theta^{b})$ to $T(y^{\text{rep}\,b}, \theta^{b})$, i.e. the estimated p-value is the proportion of B simulations for which $T(y^{\text{rep}\,b}, \theta^{b}) \geqslant T(y, \theta^{b}), b = 1, \ldots, B$

II.A Model Checking, Evaluating, and Comparing

Example II.2

Consider a sequence of binary outcomes $y_1, \ldots, y_n$ modeled with a common probability of success θ and a uniform prior. The posterior is then

$$p(\theta|y) \propto \theta^{\sum y+1-1}(1-\theta)^{n-\sum y+1-1}$$

Suppose the data is observed, in order, as

$$1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0$$

which may suggestion some degree of autocorrelation, thus the binomial model might not be the most appropriate

II.A Model Checking, Evaluating, and Comparing

Example II.2 (cont.)

To quantify this, we can perform a posterior predictive test of the number of switches between 0 and 1 in the sequence.

Our steps are now to

1. Take 10000 draws from the posterior distribution, i.e. from $Beta(8, 14)$,
2. Draw y^{rep} as independent Bernoulli trials using the results from 1 at each step
3. Count the number of switches in the replication and compare to the observed

II.A Model Checking, Evaluating, and Comparing

Example II.R1

Following the steps from the previous slide, check to see the appropriateness of the model fit. For reproducibility, set the seed to 1875. As a reminder, the sequence of the observed data is

$$1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0$$

`R` tip:

- A `for` loop may be needed to build the replicates, each of size 20, the length of the sequence
- Use `length(rle(yrep[b,])$lengths)-1` to count switches in runs where `yrep[b,]` is the sample at step `b`

II.A Model Checking, Evaluating, and Comparing

Example II.R2

Return to the data from the HERS study found in the data file `hers.txt`. The file contains the pre minus post change in LDL cholesterol for the 1291 women available for the study. We modeled this in Part I, Example I.R5 assuming y was normal and using a non-informative prior. Rerun this code and then build 10000 replicates of size 1291 for each pair of sampled (μ, σ^2) assuming normality. Then compare the observed central moments in the data to the distribution of central moments found in the replicates. Focus on mean, variance, skewness, and kurtosis.

`R` tip:

- The package `moments` contains built-in functions for skewness and kurtosis.

II.A Model Checking, Evaluating, and Comparing

Question: In the previous examples, do we have evidence to suggest the model is incorrectly specified?

II.A Model Checking, Evaluating, and Comparing

Question: In the previous examples, do we have evidence to suggest the model is incorrectly specified?

Interpreting posterior predictive p-values

A model is suspect if its observed value has a tail probability near 0 or 1, thus indicating that the observed pattern would be unlikely to be seen in replications if the data model were true

II.A Model Checking, Evaluating, and Comparing

Question: In the previous examples, do we have evidence to suggest the model is incorrectly specified?

Interpreting posterior predictive p-values

A model is suspect if its observed value has a tail probability near 0 or 1, thus indicating that the observed pattern would be unlikely to be seen in replications if the data model were true

Extreme values, < 0.01 or > 0.99,. indicate model inadequacy that needs to be addressed

II.A Model Checking, Evaluating, and Comparing

Question: In the previous examples, do we have evidence to suggest the model is incorrectly specified?

Interpreting posterior predictive p-values

A model is suspect if its observed value has a tail probability near 0 or 1, thus indicating that the observed pattern would be unlikely to be seen in replications if the data model were true

Extreme values, < 0.01 or > 0.99,. indicate model inadequacy that needs to be addressed

Ideal to consider multiple test quantities as well as note if model deficiencies do not substantially change posterior inferences

II.A Model Checking, Evaluating, and Comparing

Convergence Diagnostics

As we move into more complicated sampling schemes, we must check to see that we are actually sampling from the posterior distribution

In other words, we want to check to see that we have achieved convergence

This isn't so much a concern when we are directly sampling from the posterior, but we can use these scenarios for illustration

II.A Model Checking, Evaluating, and Comparing

There are several ways of assessing convergence, but three common approaches are

1. Visual inspection
2. Geweke's convergence statisitc
3. Gelman-Rubin statistic

The latter two will be more important when we begin fitting models using MCMC to ensure that our samples are being independently drawn

II.A Model Checking, Evaluating, and Comparing

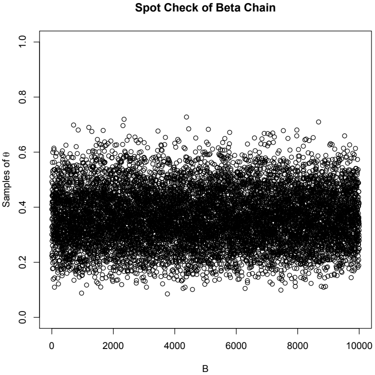
Visual inspection of posterior samples

Plot the samples against the order, i.e. $\theta^{(b)}$ against $b = 1, \ldots, B$

II.A Model Checking, Evaluating, and Comparing

Visual inspection of posterior samples

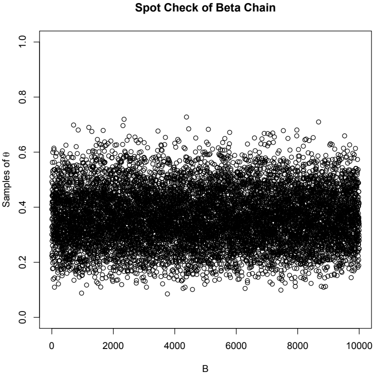
Plot the samples against the order, i.e. $\theta^{(b)}$ against $b = 1, \ldots, B$



II.A Model Checking, Evaluating, and Comparing

Visual inspection of posterior samples

Plot the samples against the order, i.e. $\theta^{(b)}$ against $b = 1, \dots, B$



Called a *trace plot*.

II.A Model Checking, Evaluating, and Comparing

Want to see

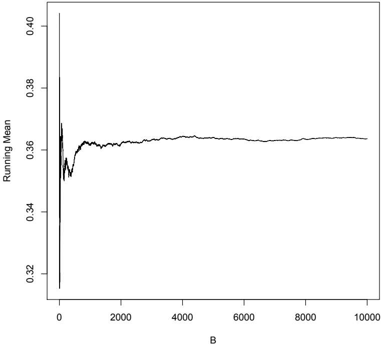
No trend, random scatter (though constrained to a small-ish region, low variability), want to see that the samples aren't still "moving"

Don't want to see

A clear trend, that the samples are moving in one direction or another, data cloud scatter across entire parameter space (large variability)

II.A Model Checking, Evaluating, and Comparing

We can also calculate a *running mean plot*:



Code

```
plot(cumsum(rtheta)/(1:B), type = 'l', ylab =
'Running Mean', xlab = 'B')
```

II.A Model Checking, Evaluating, and Comparing

Want to see

The running mean flattens out as B increases

Don't want to see

The running mean continue to move, as it did in the first 1000 samples or so, throughout the remainder of the plot

Could also use running median (median is more robust), depending on what posterior summary you want to use

II.A Model Checking, Evaluating, and Comparing

Autocorrelation Function (ACF) Plot

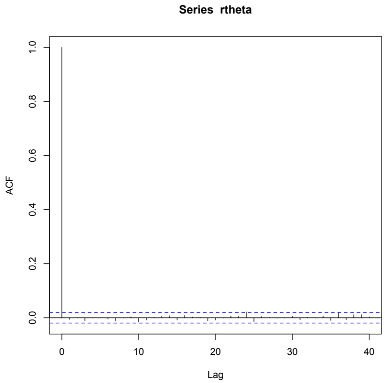
Let $\theta^{(b)}$ and $\theta^{(b+k)}$ denote the b and $b+k$ elements of our posterior sample for θ. The kth-order auto-correlation is then $\rho_k = \mathrm{corr}\left(\theta^{(b)}, \theta^{(b+k)}\right)$.

As k and m get further apart, we want ρ_k to get smaller

Idea: we want our samples to be independent of each other

II.A Model Checking, Evaluating, and Comparing

In R, use the function `acf()` which produces the following graph



Want the vertical lines to drop between the blue dashed lines quickly

II.A Model Checking, Evaluating, and Comparing

The package `mcmcplots` has a number of nicer functions we can use to visually inspect convergence

Example II.R3

Return to the samples of μ from Example II.R2. Put the samples into a matrix with one column. After loading `mcmcplots`, run the following functions on the sample:

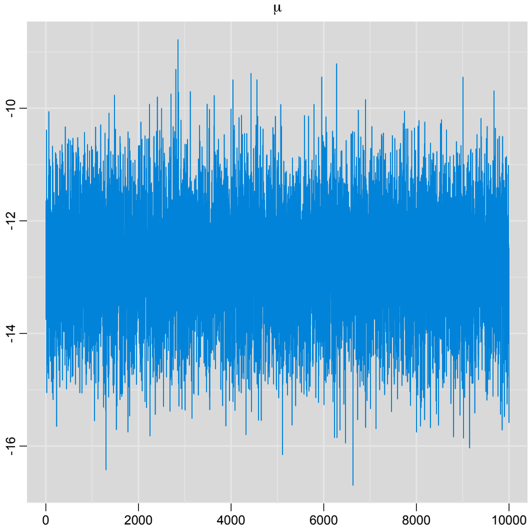
- `traplot()`
- `rmeanplot()`
- `autplot1()`, requires object have class `mcmc`
- `mcmcplot1()`, combines all three plus density plot

`R` tip:

- Use `mcmc.list(list(mcmc()))` to coerce to class `mcmc`

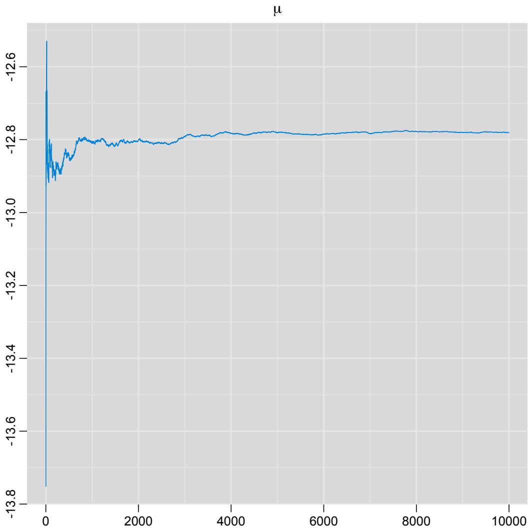
II.A Model Checking, Evaluating, and Comparing

Trace plot using `traplot()`:



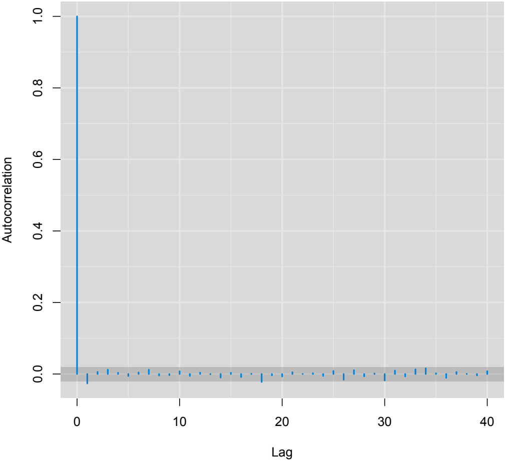
II.A Model Checking, Evaluating, and Comparing

Running mean plot using `rmeanplot()`:



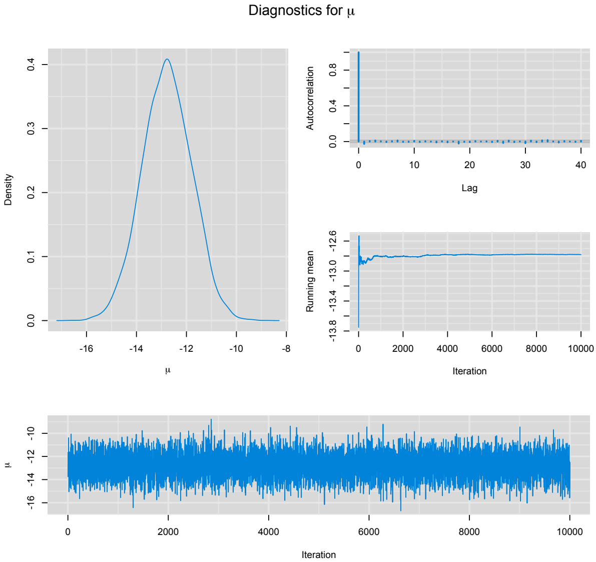
II.A Model Checking, Evaluating, and Comparing

Autocorrelation plot using `autplot1()`:



II.A Model Checking, Evaluating, and Comparing

Four-in-one plot using `mcmcplot1()`:



II.A Model Checking, Evaluating, and Comparing

We can also generate test statistics to assess convergence

Geweke Convergence Diagnostic

Takes two non-overlapping parts of the sample (often the first 10% and last 50%) and compares the means using a difference of means tests to see if the two parts of the sample are from the same distribution

The test statistic is a standard Z-score, standard error adjusted for auto-correlation

In `R`, use `geweke.diag()` from the package `coda()`

II.A Model Checking, Evaluating, and Comparing

Gelman-Rubin

Steps for each parameter:

1. Run $m \geqslant 2$ samples, or chains, of length $T = 2B$ samples
2. Discard the first B samples for each chain
3. Calculate the within-chain and between-chain variance
4. Calculate the estimated variance of the parameter as a weighted sum of the within-chain and between-chain variance
5. Calculate the potential scale reduction factor

II.A Model Checking, Evaluating, and Comparing

Within-chain Variance

$$WCV = \frac{1}{m}\sum_{j=1}^{m} s_j^2$$

where

$$s_j^2 = \frac{1}{B-1}\sum_{i=1}^{B}(\theta_{ij} - \bar{\theta}_j)^2$$

Note: s_j^2 is the variance for the jth sample, WCV is the mean of variances from each chain

II.A Model Checking, Evaluating, and Comparing

Between-chain Variance

$$\text{BCV} = \frac{B}{m-1}\sum_{j=1}^{m}(\bar{\theta}_j - \bar{\bar{\theta}})^2$$

where

$$\bar{\bar{\theta}} = \frac{1}{m}\sum_{j=1}^{m}\bar{\theta}_j$$

This is the variance of the chain means multiplied by B

II.A Model Checking, Evaluating, and Comparing

Estimated Variance

The estimated variance of the θ is the weighted sum of WCV and BCV

$$\widehat{Var}(\theta) = \left(1 - \frac{1}{\mathrm{B}}\right) WCV + \frac{1}{\mathrm{B}} \mathrm{BCV}$$

Potential Scale Reduction Factor

$$\hat{\mathrm{R}} = \sqrt{\frac{\widehat{Var}(\theta)}{WCV}}$$

If $\hat{\mathrm{R}}$ is large ($>$ 1.1 or 1.2), we should increase B

II.A Model Checking, Evaluating, and Comparing

In `R`, we need to run our sampler multiple times and classify the output as an `mcmc` object using the function `mcmc()` which we need to combine into an `mcmc.list()`

Finally, we can call `gelman.diag()` on the `mcmc.list()` object

Each of these functions can be found in the `coda` package

Example II.R4

Find Geweke and Gelman-Rubin statistics for the current example

II.A Model Checking, Evaluating, and Comparing

Once we know our chains have converged, we may want to consider a way to compare multiple models to each other.

Model Comparison

In the frequentist context, we use Akaike Information Criterion (AIC) to compare non-nested models

In the Bayesian context, we can use the Deviance Information Criterion (DIC) or Watanabe-Akaike Information Criterion (WAIC)

II.A Model Checking, Evaluating, and Comparing

DIC

This is a *somewhat* Bayesian version of AIC. The measure of predictive accuracy is

$$\mathrm{DIC} = -2\log\left[\mathcal{L}\left(y|\hat{\theta}_{Bayes}\right)\right] + 2p_{DIC}$$

where p_{DIC} can be calculated in one of two ways:

$$p_{DIC} = 2\left[\ell\left(y|\hat{\theta}_{Bayes}\right) - \frac{1}{B}\sum_{b=1}^{B}\ell\left(y|\theta^{(b)}\right)\right]$$

or

$$p_{DIC} = 2Var\left[\ell\left(y|\hat{\theta}_{Bayes}\right)\right]$$

II.A Model Checking, Evaluating, and Comparing

An alternative to the DIC is the Watanabe-Akaike Information Criterion (WAIC)

WAIC

This is a more fully Bayesian approach which relies on a penalty that is based on pointwise calculations which can be viewed as approximations of cross-validation

$$WAIC = -2\mathrm{lppd} + 2\mathrm{p}_{WAIC}$$

We define lppd and p_{WAIC} in the following slides

II.A Model Checking, Evaluating, and Comparing

lppd

Stands for log pointwise predictive density. Let $\theta^{(b)}$ denotes our samples $b = 1, \ldots, B$ from the posterior for θ, then

$$\mathrm{lppd} = \sum_{i=1}^{n} \log \left[\frac{1}{B} \sum_{b=1}^{B} \mathcal{L}\left(y_i | \theta^{(b)} \right) \right]$$

p_{WAIC}

$$p_{\mathrm{WAIC}} = \sum_{i=1}^{n} \frac{1}{B-1} \sum_{b=1}^{B} \left[\ell\left(y_i | \theta^{(b)} \right) - \frac{1}{B} \sum_{j=1}^{B} \ell\left(y_i | \theta^{(j)} \right) \right]^2$$

II.A Model Checking, Evaluating, and Comparing

Notes on DIC and WAIC:

- Similar to AIC, the smaller the value, the better the model
- Both DIC and WAIC are relatively easy to calculate given posterior estimates and samples
- WAIC has the advantage of averaging over the posterior distribution rather than conditioning on a point estimate (Gelman prefers WAIC)

Example II.R5

Determine DIC for the data from Example II.R1 and WAIC for the data from Example II.R2

II.A Model Checking, Evaluating, and Comparing

Inference

We have previously only briefly touched on inference in the Bayesian context, let's now give it a fuller treatment

There are many approaches to conducting inference:

- Credible intervals
- Posterior Probability
- Posterior statistics
- Bayes Factors
- Bayesian False Discovery Rate

II.A Model Checking, Evaluating, and Comparing

Credible Intervals

As we've previously discussed, credible intervals are the Bayesian equivalent to confidence intervals

Suppose we have the null hypothesis: $H_0 : \mu = c$ for some constant c

If our credible interval contains c, we fail to reject H_0, otherwise we reject it at level α, for a $100(1 - \alpha)\%$ Credible Interval

II.A Model Checking, Evaluating, and Comparing

Posterior Probability

Suppose $\theta^{(b)}$ is the bth draw from our retained posterior samples and suppose we wish to test $H_0 : \mu = c$ for some constant c

We may also be interested in calculating the posterior probability that θ is different from c:

$$\frac{1}{B}\sum_{b=1}^{B} 1\left(\theta^{(b)} > c\right) \text{ or } \frac{1}{B}\sum_{b=1}^{B} 1\left(\theta^{(b)} < c\right)$$

If either is sufficiently small, it provides favor in evidence of H_A

II.A Model Checking, Evaluating, and Comparing

We're not limited to inference on a single parameter alone, we can calculate and sample from the posterior distribution of any statistic we want

Example II.3

Suppose we want to compare the proportion of success between two groups, θ_1 and θ_2. For each draw b from the posterior, we could generate a posterior sample for:

- The risk difference: $\theta_1^{(b)} - \theta_2^{(b)}$
- The risk ratio: $\theta_1^{(b)} \Big/ \theta_2^{(b)}$
- The odds ratio: $\mathrm{Odds}\left(\theta_1^{(b)}\right) \Big/ \mathrm{Odds}\left(\theta_2^{(b)}\right)$

II.A Model Checking, Evaluating, and Comparing

Example II.4

Suppose we want to compare the means in two different groups, μ_1 and μ_2. For each draw b from the posterior, we could generate the difference:

$$\mu_1^{(b)} - \mu_2^{(b)}$$

In both of the previous examples, we could then conduct inference on the posterior distribution of the statistic using Credible Intervals or Posterior Probabilities or both

II.A Model Checking, Evaluating, and Comparing

DIC and WAIC evaluate models based on expected predictive accuracy

When a discrete set of competing models is considered, we can determine Bayes Factors

Bayes Factors

Suppose we have two competing models, H_1 and H_2, we may want to compare the posteriors:

$$\frac{p(H_2|y)}{p(H_1|y)} = \frac{p(H_2)}{p(H_1)} \times \mathrm{BF}(H_2; H_1)$$

II.A Model Checking, Evaluating, and Comparing

The last term from the previous slide is the Bayes Factor, which more explicitly defined is

$$\mathrm{BF}(H_2; H_1) = \frac{p(y|H_2)}{p(y|H_1)} = \frac{\int p(\theta_2, y|H_2)d\theta_2}{\int p(\theta_1, y|H_1)d\theta_1}$$

Ostensibly, integrating the parameter out of the posterior

The goal when using Bayes Factors is to choose a single model H_i or average over a discrete set using their posterior probabilities

Bayes factors work well when the underlying model is truly discrete

II.A Model Checking, Evaluating, and Comparing

Example II.5

Suppose we are studying five-year survival for lung cancer patients who are under 40. Prior research suggests that the proportion of lung cancer patients surviving five-years among those are over 40 is 8.2%. We followed 52 lung cancer patients under the age of 40. At the end of five years, only six were still alive.

1. Determine both models (one is fixed, the other is unknown—must find posterior)
2. Find the Bayes Factor

II.A Model Checking, Evaluating, and Comparing

Let X denote the number who survived and let H_1 be the model under the “null,” i.e. where $\theta = 0.082$. Thus H_1 is

$$P(X = 6|H_1, \theta = 0.082) = \binom{52}{6} 0.082^6 (1 - 0.082)^{46} = 0.1208$$

Under H_2, note the posterior, assuming a uniform prior, is $\propto \theta^{X+1-1}(1-\theta)^{n-X+1-1}$, thus

$$\begin{aligned} P(X = 6|H_2) &= \int_0^1 \binom{52}{6} \theta^{6+1-1}(1-\theta)^{46+1-1} d\theta \\ &= \binom{52}{6} \frac{\Gamma(7)\Gamma(47)}{\Gamma(54)} = \frac{52!}{6!42!} \frac{6!46!}{53!} = 0.01886 \end{aligned}$$

II.A Model Checking, Evaluating, and Comparing

Our Bayes Factor is then

$$\mathrm{BF}(H_1; H_2) = \frac{P(X = 6 | H_2)}{P(X = 6 | H_1, \theta = 0.082)} = 0.1561$$

What does this tell us?

II.A Model Checking, Evaluating, and Comparing

Our Bayes Factor is then

$$\mathrm{BF}(\mathrm{H}_1;\mathrm{H}_2) = \frac{\mathrm{P}(\mathrm{X}=6|\mathrm{H}_2)}{\mathrm{P}(\mathrm{X}=6|\mathrm{H}_1,\theta=0.082)} = 0.1561$$

What does this tell us?

Jeffreys' Guide

BF	Evidence
< 1	Supports H_1 (invert for strength)
$1 - 3.2$	Barely worth mentioning
$3.2 - 10$	Substantial
$10 - 31.6$	Strong
$31.6 - 100$	Very strong
> 100	Decisive

II.A Model Checking, Evaluating, and Comparing

Just as in the frequentist approach, we can easily run into the issue of multiple testing in the Bayesian context

When using credible intervals to conduct inference, we could simply lower α to give a more stringent control of Type I Error (i.e. a Family-wise Error Rate procedure like Bonferroni)

However, in some scenarios, we may want a procedure that results in greater power than the FWER procedures, albeit at the cost of increased Type I Error

II.A Model Checking, Evaluating, and Comparing

Some background:

Benjamini-Hochberg False Discovery Rate (FDR)

Suppose we observe M p-values.

1. Sort the p-values from smallest, $p_{(1)}$, to largest, $p_{(M)}$
2. For a desired α, find the largest k such that $p_{(k)} \leqslant \frac{k}{M}\alpha$
3. Reject H_0 for all ordered test $1, \ldots, k$

In the frequentist context, this approach results in higher power than the Bonferroni correction, though not as tight control of Type I Error

II.A Model Checking, Evaluating, and Comparing

In the Bayesian context, it had previously been thought that Posterior inference adjusts (inherently) for multiplicity, however Müller, Parmigiani, and Rice (2006) showed this only true under certain circumstances

They proposed a Bayesian version of the FDR for when examining genomic data, however it can be used for any setting where multiple testing occurs

II.A Model Checking, Evaluating, and Comparing

Suppose we wish to conduct inference on a model with M parameters, denoted $\theta_1, \ldots, \theta_M$

For a given threshold, δ, determine the posterior probabilities as

$$p(m) = \Pr(|\theta_m| \leqslant \delta | y) \approx \frac{1}{B} \sum_{b=1}^{B} 1 \left(\left| \theta_m^{(b)} \right| \leqslant \delta \right)$$

where $\theta_m^{(b)}$ is the bth posterior sample for the parameter θ_m

If a $p(m) = 0$, set $p(m) = (2M)^{-1}$

II.A Model Checking, Evaluating, and Comparing

Then for a pre-specified global FDR-bound α, we select the set of parameters satisfying

$$\psi = \{m : p(m) \leqslant \nu_\alpha\}$$

To obtain ν_α, we sort $p(m)$ from smallest to largest giving the set

$$\{p_{(r)}, r = 1, \ldots, R\}$$

and define

$$\lambda = \max \left[r^* : \frac{1}{r^*} \sum_{r=1}^{r^*} p_{(r)} \leqslant \alpha \right]$$

the cutoff is then $\nu_\alpha = p_{(\lambda)}$

Unit B: Introduction to Bayesian Computation

Chapter 10

II.B Introduction to Bayesian Computation

Thus far we have focused on the setting where the posterior and posterior predictive distributions could be computed analytically in closed form or approximated with a Gaussian density

But for complicated or unusual models or models with high dimensions, more elaborate algorithms are required to approximate posterior distributions

II.B Introduction to Bayesian Computation

Target distribution

The distribution to be simulated is the *target distribution* which we denote with $p(\theta|y)$

Unnormalized density

The *unnormalized density*, $q(\theta|y)$ is an easily computable function for which

$$\frac{q(\theta|y)}{p(\theta|y)} = \text{constant}$$

and depends only on y

Note: θ may be multi-dimensional

Log Densities

To avoid computational overflows and underflows, compute the logarithm of the posterior densities whenever possible and perform exponentiation only when necessary and as late as possible

For example if the algorithm requires the ratio of two densities, compute it as the exponential of the difference of two log densities

II.B Introduction to Bayesian Computation

Numerical Integration

Methods in which an integral over a continuous function is evaluated by computing the value of the function at a finite number of points

Increasing the number of points increases the accuracy

Can divide methods into

1. Stochastic simulation (Monte Carlo)
2. Deterministic (quadrature rule)

II.B Introduction to Bayesian Computation

1) Suppose we wish to estimate the posterior expectation of some function, say $h(\theta)$

Via stochastic simulation, based on random samples θ^b from the desired distribution, we compute

$$E[h(\theta|y)] = \int h(\theta)p(\theta|y)d\theta \approx \frac{1}{B}\sum_{b=1}^{B} h\left(\theta^b\right)$$

Need way to produce independent samples, θ^b

II.B Introduction to Bayesian Computation

2) Using a deterministic method, the integrand is evaluated at selected points based on a weighted version of the calculation from the previous slide:

$$E[h(\theta|y)] = \int h(\theta)p(\theta|y)d\theta \approx \frac{1}{B}\sum_{b=1}^{B} w_b h\left(\theta^b\right) p(\theta^b|y)$$

with weight w_b corresponding to the volume of space represented by the point θ^b

More elaborate rules can improve accuracy

II.B Introduction to Bayesian Computation

Deterministic methods typically have lower variance than stochastic simulation, but the selection of locations is difficult, particularly in high dimensions

For this reason, we will focus on stochastic simulation approaches, namely

1) Monte Carlo methods
2) Markov chain Monte Carlo methods

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Direction Approximation

For the simplest discrete approximation, compute the target density, $p(\theta|y)$, at a set of equally spaced values $\theta_1, \ldots, \theta_N$ that cover a broad range of the parameter space

Then approximate the continuous $p(\theta|y)$ by the discrete density at $\theta_1, \ldots, \theta_N$ with probabilities

$$\frac{p(\theta_i|y)}{\sum_{j=1}^{N} p(\theta_j|y)}$$

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Once the grid of density values is computed, generate a random draw from $p(\theta|y)$ by

1. Drawing U from $U(0,1)$
2. Transforming U using the probability integral transform for a discrete distribution (since the deterministic distribution is discrete)

Difficult for higher-dimensional models

Not as useful an approach...

Rejection Sampling

Suppose we want to obtain a single random draw from $p(\theta|y)$. Given a positive function $g(\theta)$ defined for all θ for which $p(\theta|y) > 0$, we can draw from the probability density proportional to g—it is not necessary that $g(\theta)$ integrate to 1, but it must have a finite integral

The *importance ratio* is defined as $p(\theta|y)/g(\theta)$ and must satisfy

$$\frac{p(\theta|y)}{g(\theta)} \leqslant M \ \forall \ \theta$$

where M is a constant

Rejection Sampling (cont.)

The rejection sampling algorithm proceeds in two steps:

1. Sample θ^b at random from the probability density proportional to $g(\theta)$ and sample U from a $U(0,1)$
2. If $U < \frac{p(\theta^b|y)}{Mg(\theta^b)}$, accept θ as a draw from p. If the draw is rejected, return to step 1

Repeat these steps until B samples have been accepted

Notes on rejection sampling:

- The distribution of drawn θ is, conditional on it being accepted, $p(\theta|y)$
- A a good approximate density $g(\theta)$ should be roughly proportional to $p(\theta|y)$ when considered as a function of θ
  - Ideal is $g \propto p$ where with a suitable M we can accept every draw w.p.1.
- If g is not proportional to p, M may have to be large meaning almost every draw is rejected
- It is good practice to track and report the acceptance rate as well, i.e. what proportion of θ's were accepted as draws from $p(\theta|y)$

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Example II.R6

For the sake of illustration, consider trying to sample from a $Beta(6, 3)$ distribution. A reasonable choice for g is the uniform distribution as it has the same support.

1. Find the value of M such that the $Mg(\theta) \geqslant p(\theta|y)$
2. Set up and run a rejection sampler to obtain 100 draws from $Beta(6, 3)$

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Importance Sampling

Related to rejection sampling and a precursor to the Metropolis algorithm that is used for computing expectations using a random sample drawn from an approximation to the target distribution.

Suppose we are interested in $E[h(\theta|y)]$ but we cannot directly generate random draws of θ from $p(\theta|y)$.

If $g(\theta)$ is a probability density from which we can generate random draws, we can re-express $E[h(\theta|y)]$ as

$$E[h(\theta|y)] = \frac{\int h(\theta)p(\theta|y)d\theta}{\int p(\theta|y)d\theta} = \frac{\int [h(\theta)p(\theta|y)/g(\theta)]g(\theta)d\theta}{\int [p(\theta|y)/g(\theta)]g(\theta)d\theta}$$

given B draws $\theta^1, \dots, \theta^B$ from $g(\theta)$, we can approximate this with

$$E[h(\theta|y)] \approx \frac{\frac{1}{B}\sum_{b=1}^{B} h(\theta^b)w(\theta^b)}{\frac{1}{B}\sum_{b=1}^{B} w(\theta^b)}$$

for $w(\theta^b) = p(\theta^b|y)/g(\theta^b)$, the *importance ratios*

Notes on Importance Sampling

- Let $h(\theta)$ be the posterior estimate we want to approximate, i.e. $h(\theta) = \theta$ for mean, etc.
- Generally advisable to use the same set of random draws for both the numerator and the denominator to reduce sampling error
- If $g(\theta)$ can be chosen such that $\frac{hp}{g}$ is roughly constant, then fairly precise estimates can be obtained
- Not useful when the importance ratios vary substantially
- Worst scenario: the importance ratios are small with high probability but with a low probability are huge
  - Occurs if p has wide tails compared to g, as a function of θ

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Example II.6

Suppose we take an iid sample of $x_i \sim Cauchy(\theta, 1)$ for $i = 1, \ldots, n$ and use a flat prior on θ. The posterior is then

$$\begin{aligned} p(\theta|x_i) &\propto \mathcal{L}(x|\theta)\pi(\theta) \\ &\propto \prod_{i=1}^{n} [1 + (x_i - \theta)]^{-1} \end{aligned}$$

A possible choice for $g()$ is the normal density $N(\mu, \sigma^2)$, which we can easily sample from

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Using the normal density for g, our importance ratio is

$$w(\theta^b) = \frac{\prod_{i=1}^n [1 + (x_i - \theta)]^{-1}}{\exp\left[-(\theta - \mu)^2/2\right]}$$

Then, given a sample $\theta^1, \ldots, \theta^B$ from a normal density, we can approximate the posterior mean using

$$E(\theta|y) = \frac{\sum_{b=1}^B \theta^b \exp\left[(\theta - \mu)^2/2\right] \prod_{i=1}^n [1 + (x_i - \theta)]^{-1}}{\sum_{b=1}^B \exp\left[(\theta - \mu)^2/2\right] \prod_{i=1}^n [1 + (x_i - \theta)]^{-1}}$$

Note: Since the Cauchy has fatter tails than the normal, this may not be the most efficient model

II.B Introduction to Bayesian Computation

Monte Carlo Methods

It is possible that we have missed some extremely large but rare importance ratios (or weights)

We may want to examine the distribution of the sampled weights

A histogram of the log of the largest importance ratios (the smallest ratios do not have much influence)

- Estimates will often be poor if the largest ratios are too large relative to the average

II.B Introduction to Bayesian Computation

If the variance of the weights are finite, we can determine the effective number of samples needed for *importance sampling*

Effective Number of Samples

The effective number of samples can be estimated using the approximation

$$\mathrm{B}_{\mathrm{eff}} = \frac{1}{\sum_{\mathrm{b}=1}^{\mathrm{B}} \left[\bar{w}\left(\theta^{\mathrm{b}}\right)\right]^2}$$

where $\bar{w}\left(\theta^{\mathrm{b}}\right) = w\left(\theta^{\mathrm{b}}\right) \Big/ \sum_{\mathrm{b}'=1}^{\mathrm{B}} w\left(\theta^{\mathrm{b}'}\right)$

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Interpreting B_{eff}

The effective number of samples, B_{eff}, is small if there are few extremely high weights which would unduly influence the distribution

If, however, the distribution has occasionally very large weights, than B_{eff} will be noisy, so this should really only be taken as a rough guide

To obtain independent samples with equal weights, we can use *importance resampling*

Importance Resampling

Also called *sampling-importance resampling* or SIR. Given draws $\theta^1, \dots, \theta^B$ from g, a sample of $k < B$ draws can be sampled using:

1. Sample a value θ from $\{\theta^1, \dots, \theta^B\}$ where the probability of sampling each θ^b is proportional to the weight $w(\theta^b) = p(\theta^b|y)/g(\theta^b)$
2. Sample a second θ, excluding the one already sampled
3. Repeatedly sample without replacement $k - 2$ more times

II.B Introduction to Bayesian Computation

Monte Carlo Methods

Notes on Importance Resampling:

- If the importance weights are moderate, sampling with replacement gives the same results
- If there are a few large weights and many small weights, then sampling *with* replacement will pick the same few values of θ repeatedly
- But if we sample *without* replacement, then these large values would like be removed early and the result would be a more intermediate approximation somewhere between the starting and target densities

Example II.7

Returning to our Cauchy example, note the weights have the form

$$w(\theta^b) = \exp\left[(\theta-\mu)^2/2\right]\prod_{i=1}^{n}[1+(x_i-\theta)]^{-1}$$

To implement importance resampling, we would

1. Sample $\theta^1, \ldots, \theta^B$ from $N(\mu, \sigma^2)$
2. Then we take repeated samples from $\left\{\theta^1, \ldots, \theta^B\right\}$ with probabilities $\left\{w\left(\theta^1\right), \ldots, w\left(\theta^B\right)\right\}$ without replacement until we have k total samples

Uses for Importance Sampling and Resampling:

Importance Sampling

- Can be used to improve analytic posterior approximations

Importance Resampling

- If IS does not yield an accurate approximation than IR can still be helpful for obtaining starting points for an iterative simulation
- Also useful when considering mild changes to the posterior, such as changing the likelihood or computing leave-one-out cross-validation

II.B Introduction to Bayesian Computation

Monte Carlo Methods

One common question is how many samples are needed?

II.B Introduction to Bayesian Computation

Monte Carlo Methods

One common question is how many samples are needed?

Our goal is to obtain a set of independent draws θ^b, $b = 1, \ldots, B$, from the posterior with enough draws that quantities of interest can be estimated with reasonable accuracy

Gelman notes that in the text, most examples rely on a moderate number of draws (between 100 and 2000)

Though a higher level of accuracy demands more draws

II.B Introduction to Bayesian Computation

We have now considered several approaches to generating samples via Monte Carlo methods

However, these approaches may not always be efficient

Thus we now turn to Markov Chain Monte Carlo (MCMC) approaches

Unit C: Markov Chain Monte Carlo Methods

Chapter 11

II.C Markov Chain Monte Carlo Methods

MCMC methods can be used to sample from an arbitrary posterior distribution

They are general methods based on drawing values of θ from approximate distributions and then correcting those draws to better approximate the target posterior

Sampling is done sequentially with the distribution of the sampled draws depending on the last value drawn—hence the draws form a Markov chain

II.C Markov Chain Monte Carlo Methods

Markov Chain

A *Markov Chain* is a sequence of random variables $\theta^{(1)}, \theta^{(2)}, \ldots,$ for which, for any b, the distribution of $\theta^{(b)}$ given all previous θ's depends only on the most recent value, $\theta^{(b-1)}$

The key is not the Markov property, but rather that the approximate distributions are improved at each step in the simulation—and thus converging to the target distribution

II.C Markov Chain Monte Carlo Methods

Markov Chain Simulation

We create several independent sequences where each sequence $\theta^{(1)}, \theta^{(2)}, \ldots,$ is produced by starting at some point $\theta^{(0)}$ and then, for each b, drawing $\theta^{(b)}$ from a *transition distribution*, $T_b\left(\theta^{(b)}|\theta^{(b-1)}\right)$.

The transition probability distributions must be constructed so that the Markov chain converges to a unique stationary distribution that is the posterior distribution

II.C Markov Chain Monte Carlo Methods

We use Markov chain simulation when it is not possible or is computationally inefficient to sample θ directly from $p(\theta|y)$

It is key that once the simulation algorithm has been implemented, we check the convergence of the simulated sequences

We now consider three common MCMC approaches:

1. Gibbs Sampler
2. Metropolis Algorithm
3. Metropolis-Hastings Algorithm

II.C Markov Chain Monte Carlo Methods

Gibbs Sampler

Suppose the parameter θ can be divided into d components or subvectors, $\theta = (\theta_1, \ldots, \theta_d)$. Each iteration of the Gibbs sampler cycles through the subvectors of θ, drawing each subset conditional on the value of all the others.

At each iteration, there are d steps with each component sampled from the full conditional

$$p\left(\theta_j | \theta_{-j}^{(b-1)}, y\right)$$

that is sampling the conditional posterior for θ_j given the values all parameters, except θ_j, from the previous step

II.C Markov Chain Monte Carlo Methods

The Gibbs Sampler relies on the posterior conditionals

Suppose our likelihood depends on two parameters, $\mathcal{L}(y|\mu, \tau)$

To use the Gibbs Sampler, we must find the conditionals:

$$p(\mu|\tau, y) = \frac{p(\mu, \tau|y)}{p(\tau|y)} \text{ and } p(\tau|\mu, y) = \frac{p(\mu, \tau|y)}{p(\mu|y)}$$

We then iterate the following steps:

1. Draw $\mu^{(b)}$ from $p\left(\mu|\tau^{(b-1)}, y\right)$
2. Draw $\tau^{(b)}$ from $p\left(\tau|\mu^{(b-1)}, y\right)$

II.C Markov Chain Monte Carlo Methods

Our first step in determining a sampler, is to see if the conditional posteriors are recognizable as the kernels of known densities

Example II.8

Let y_i be iid $N(\mu, \sigma^2)$ for $i = 1, \ldots, n$. Further, place the non-informative joint prior on μ and σ^2, i.e. $\pi(\mu, \sigma^2) \propto (\sigma^2)^{-1}$

Determine the likelihood, posterior, as well as the conditional posteriors

II.C Markov Chain Monte Carlo Methods

Using a Gibbs Sampler, and given starting values of μ^0 and ${\sigma^2}^0$, we can obtain draws from the posterior by iterating between draws of

1. ${\sigma^2}^{(t)}|\mu^{(t-1)}, y \sim \mathrm{IG}\left[n/2, (1/2)\sum(y_i - \mu^{(t-1)})^2\right]$
2. $\mu^{(t)}|{\sigma^2}^{(t-1)}, y \sim \mathrm{N}\left(\bar{y}, {\sigma^2}^{(t-1)}/n\right)$

Once we have drawn B samples, we can evaluate the model parameters using the graphical summaries we discussed in Unit A

We can also conduct inference as discussed in Unit A

II.C Markov Chain Monte Carlo Methods

Example II.R7

Suppose we are studying change in total cholesterol in the HERS study comparing those who were on placebo to those who were on hormone replacement therapy. Change in total cholesterol can bee assumed to be normally distributed. The data is in the file `hersct.txt` in the data folder.

Find the posterior mean and variance of total change in cholesterol for groups, also find the posterior distribution of the difference in means, and check to make sure that the chains have converged and there is no auto-correlation in the samples. Take a total of $\mathrm{B} = 10000$ samples and use the sample means and variances as starting values. Set the seed to 1222.

II.C Markov Chain Monte Carlo Methods

In Note Set I, used the marginal-conditional approach for regression

Alternatively, we can implement a Gibbs sampler

Example II.9

As before, we assume the likelihood has the form

$$y \sim MVN\left(X\boldsymbol{\beta}, \sigma^2 I_{n\times n}\right)$$

and use the noninformative prior

$$p(\boldsymbol{\beta}, \sigma^2) \propto (\sigma^2)^{-1}$$

Note: keep in mind, $\boldsymbol{\beta}$ is a vector

II.C Markov Chain Monte Carlo Methods

Example II.R8

Download the dataset `hersreg.txt` from the data folder. Build a Bayesian linear regression model using change in total cholesterol, `chtchol` as the outcome and `treatment` as the primary covariate of interest. Also control for baseline systolic blood pressure, `sbp`, as well as statin use, `statins`.

Implement a Gibbs sampler to run this model with $\mathrm{B} = 10000$, use `rmvnorm()` from the `mvtnorm` package, and set the seed to 50

II.C Markov Chain Monte Carlo Methods

The Gibbs sampler can actually be viewed as a special case of a broader family of algorithms known as *Metropolis-Hastings Algorithms*

The other two members of this family are

- Metropolis Algorithm
- Metropolis-Hastings Algorithm (for which the family is named)

II.C Markov Chain Monte Carlo Methods

Metropolis Algorithm

The Metropolis algorithm is an adaptation of a random walk with an acceptance/rejection rule to converge to the specified target distribution.

Algorithm:

1) Draw a starting point $\theta^{(0)}$ for which $p(\theta^{(0)}|y) > 0$ from a *starting distribution* $p_0(\theta)$. This could be selected using importance resampling, for example, or can be some crude approximate estimate

II.C Markov Chain Monte Carlo Methods

2) For $t = 1, 2, \ldots$

   (a) Sample a proposal θ^* from a *proposal distribution*, $J_b\left(\theta^*|\theta^{(b-1)}\right)$, at step b. $J_b\left(\theta^*|\theta^{(b-1)}\right)$ must be a symmetric distribution.

   (b) Calculate the ratio of the densities,

$$r = \frac{p(\theta^*|y)}{p\left(\theta^{(b-1)}|y\right)}$$

   (c) Generate $U \sim U(0, 1)$, set

$$\theta^{(b)} = \begin{cases} \theta^* & \text{w.p. } \min\left(r, 1\right) \\ \theta^{(b-1)} & \text{otherwise} \end{cases}$$

   that is, if $U < \min\left(r, 1\right)$, accept θ^* as a draw from the posterior, otherwise stay at $\theta^{(b-1)}$

II.C Markov Chain Monte Carlo Methods

Given the current value $\theta^{(b-1)}$ the transition distribution $T_b(\theta^{(b)}|\theta^{(b-1)})$ of the Markov chain is thus a mixture of a point mass at $\theta^{(b)} = \theta^{(b-1)}$ and a weighted version of the proposal distribution $J_b\left(\theta^*|\theta^{(b-1)}\right)$, that adjusts for the acceptance rate

The proposal distribution is sometimes called the *jumping distribution* as it determines whether or not the algorithm "jumps" from $\theta^{(b-1)}$ to a new draw, θ^*

Note: if θ^* is *not* accepted, then $\theta^{(b)}$ is set to the current value, i.e. to $\theta^{(b-1)}$ and the iteration is progressed

II.C Markov Chain Monte Carlo Methods

Relation to optimization

The acceptance/rejection rule of the Metropolis algorithm can be stated as follows:

(a) If the jump increases the posterior density, set $\theta^{(b)} = \theta^*$

(b) If the jump decreases the posterior density, set $\theta^{(b)} = \theta^*$ w.p.r and $\theta^{(b)} = \theta^{(b-1)}$ otherwise

In other words, the Metropolis algorithm always accepts steps that increase the density and only sometimes accepts downward steps (essentially a version of a stepwise mode-finding algorithm)

II.C Markov Chain Monte Carlo Methods

To show that the sequence of iterations $\theta^{(1)}, \theta^{(2)}, \ldots$ resulting from the Metropolis algorithm converges to the target distribution takes two steps:

First, we'd have to show that the simulated sequence is a Markov chain with a unique stationary distribution

Second, we'd have to show that the stationary distribution equals the target distribution

For more details, see Gelman page 279.

Example II.R9

Suppose we have n samples of Laplacian distributed random variables, $z_i \sim Laplace(\mu, 1)$, with unknown center μ and scale 1. Further suppose we place a flat prior on μ, thus $\pi(\mu) \propto 1$. Using a normal proposal density with variance equal to 1 and mean equal to the sample median, draw estimates from the posterior density $p(\mu|z)$. Take $B = 1000$ samples and set the seed to 8562. The data can be found in the file `laplace.txt`.

Hint: write a function for the log-likelihood and exponentiate

II.C Markov Chain Monte Carlo Methods

Note that the Metropolis algorithm requires a symmetric proposal distribution

If a target density is symmetric, this is a reasonable choice

However, if the target density is not-symmetric, a symmetric proposal density could

- Be inefficient
- Under-sample from the tail

II.C Markov Chain Monte Carlo Methods

A generalization of the Metropolis algorithm is the Metropolis-Hastings

Metropolis-Hastings Algorithm

This algorithm modifies the Metropolis algorithm in two ways

1. The proposal density need no longer be symmetric
2. To correct for the asymmetry, the ratio r is replaced by

$$\rho = \frac{p\left(\theta^{*}|y\right)/J_{b}\left(\theta^{*}|\theta^{(b-1)}\right)}{p\left(\theta^{(b-1)}|y\right)/J_{b}\left(\theta^{(b-1)}|\theta^{*}\right)}$$

II.C Markov Chain Monte Carlo Methods

Metropolis-Hastings Algorithm

We can then generate samples from an arbitrary posterior distribution using the algorithm below.

Algorithm:

1) Draw a starting point θ^0 for which $p(\theta^0|y) > 0$ from a *starting distribution* $p_0(\theta)$. This could be selected using importance resampling, for example, or can be some crude approximate estimate

II.C Markov Chain Monte Carlo Methods

2) For $b = 1, 2, \ldots$

   (a) Sample a proposal θ^* from a *proposal distribution*, $J_b\left(\theta^*|\theta^{(b-1)}\right)$, at step b.

   (b) Calculate the ρ

   $$\rho = \frac{p\left(\theta^*|y\right)/J_b\left(\theta^*|\theta^{(b-1)}\right)}{p\left(\theta^{(b-1)}|y\right)/J_b\left(\theta^{(b-1)}|\theta^*\right)}$$

   (c) Generate $U \sim U(0,1)$, set

   $$\theta^{(b)} = \begin{cases} \theta^* & \text{w.p. } \min\left(\rho, 1\right) \\ \theta^{(b-1)} & \text{otherwise} \end{cases}$$

   that is, if $U < \min\left(\rho, 1\right)$, accept θ^* as a draw from the posterior, otherwise stay at $\theta^{(b-1)}$

II.C Markov Chain Monte Carlo Methods

Notes on the Metropolis-Hastings Algorithm:

- Allowing asymmetric jumping rules (i.e. proposal distributions) can increase the speed of the random walk

- Convergence to the target density is proved in a similar manner to that of the Metropolis algorithm

- The proof of convergence to a unique stationary distribution is identical

- When the proposal density *is* symmetric, $J_b\left(\theta^*|\theta^{(b-1)}\right) = J_t\left(\theta^{(b-1)}|\theta^*\right)$ and algorithm reduces to the Metropolis

II.C Markov Chain Monte Carlo Methods

Example II.R10

Suppose we wish to take samples from an inverse Gaussian distribution which has density:

$$p(x) = \left(\frac{\lambda}{2\pi x^3}\right)^{1/2} \exp\left[-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right]$$

In `R`, first generate a plot of this density when $\lambda = 1$ and $\mu = 1$ and determine a relevant proposal density

II.C Markov Chain Monte Carlo Methods

Example II.R10 (cont.)

Given that the inverse Gaussian is skewed, a Gamma proposal density would be a reasonable choice

First write your sampler using a $\mathrm{Gamma}(1, 1)$ proposal, check convergence, ACF plots, and find the acceptance rate. Set the seed to 882.

Next, use a $\mathrm{Gamma}(2, 1)$ proposal and compare convergence, ACF, and acceptance rate. Once again, set the seed to 882.

Proposal Density Selection

Ideally, $J(\theta^*|\theta) \approx p(\theta^*|y)$ for all θ and thus ρ is essentially always 1. Thus we also accept the jump and are essentially directly sampling from the posterior

However, the Metropolis and M-H algorithms are usually applied to problems for which finding a known $J(\theta^*|\theta)$ to establish that approximation can be quite difficult (if not impossible)

II.C Markov Chain Monte Carlo Methods

Proposal density attributes

A good proposal density should have the following properties:

- For any θ, it is easy to sample from $J(\theta^*|\theta)$.
- It is easy to compute r or ρ
- Each jump goes a reasonable distance in the parameter space
  - Otherwise the random walk moves too slowly
- The jumps are not rejected too frequently
  - Otherwise the random walk stands still for too long

II.C Markov Chain Monte Carlo Methods

Acceptance Rate

Roberts, Gelman, and Gilks (1997) showed that for a one-dimensional Gaussian, the theoretically ideal acceptance rate is approximately 44% which decreases to 23% for an D-dimensional Gaussian

Thus we aim for acceptance rates between roughly 20% and 50%

II.C Markov Chain Monte Carlo Methods

Proposal Variance

If the proposal variance is too small, the acceptance rate will be high but successive samples will move around the space slowly and the chain will converge only slowly

But if the proposal variance is too large, the acceptance rate will be very low because the proposals are likely to land in regions of much lower probability density and again the chain will converge very slowly

Should consider a range of proposal variances to tune our sampler

II.C Markov Chain Monte Carlo Methods

Which Sampler to Choose?

The Gibbs Sampler is the simplest MCMC method and should be our first choice if possible

For example, when we have conditionally conjugate models, the Gibbs Sampler will be straight forward to implement

When the model is not conditionally conjugate, the Metropolis or Metropolis-Hastings algorithms can be implemented

II.C Markov Chain Monte Carlo Methods

In many cases, *some* parameters are conditionally conjugate while others are not

In these instances, a Gibbs-like approach can be taken that combines the Gibbs and M-H samplers

Gibbs-M-H Algorithm

This is a hybrid algorithm which samples from the conditional posterior distribution of each parameter, as in the Gibbs place, but implements an M-H step for non-recognizable conditional densities

II.C Markov Chain Monte Carlo Methods

Example II.10

Suppose we have three parameters we wish to obtain posterior samples for: θ, γ, and σ. Further suppose that θ and σ are conditionally conjugate but γ is not. Our algorithm would proceed as follows:

1. Draw $\theta^{(b)}$ from $\theta^{(b)}|\gamma^{(b-1)}, \sigma^{(b-1)}, y$
2. Draw $\sigma^{(b)}$ from $\theta^{(b)}|\gamma^{(b-1)}, \theta^{(b-1)}, y$
3. Draw $\gamma^{(b)}$ using a proposal density $J\left(\gamma^{*}|\gamma^{(b-1)}, \theta^{(b-1)}, \sigma^{(b-1)}, y\right)$

II.C Markov Chain Monte Carlo Methods

Example II.11

Suppose we take a sample of Laplacian random variables, i.e. $y_i \sim Laplace(\mu, \tau)$ with density

$$p(y) = \frac{\tau}{2} \exp\left(-\tau|y-\mu|\right)$$

Note this density is parameterized in terms of the precision, i.e. $\tau = 1/\sigma$ where σ is the scale

Using a non-informative prior of τ^{-1}, let's find the likelihood, posterior, and determine the full conditionals

II.C Markov Chain Monte Carlo Methods

Example II.R11

In R, implement the sampler from Example II.11. In particular, use the data in the file `laplace.txt` and implement a Gibbs-M-H algorithm to take posterior draws from the full conditionals of both μ and τ. What proposal density is appropriate?

Tune the variance of the proposal density to achieve a desired acceptance rate and check for convergence and auto-correlation in each parameter

II.C Markov Chain Monte Carlo Methods

Gibbs as Special Case of M-H

If we consider the M-H for a single parameter, say θ_j, and let the proposal density be conditional posterior density of θ_j given $\theta_{-j}^{(b-1)}$, then it can be shown that $\rho \equiv 1$.

Thus every jump is accepted

For a short proof, see Gelman page 281

II.C Markov Chain Monte Carlo Methods

Inference and Convergence

Inference from iterative simulation is ostensibly the same as for direct simulation from analytic posteriors:

- Use the collection of draws from $p(\theta|y)$ to summarize the posterior, compute quantiles, moments, and other summaries of interest
- Posterior predictive draws of $\tilde{y}$ can be obtained by simulation conditional on the drawn values of θ

However, additional considerations are needed when using iterative simulation

II.C Markov Chain Monte Carlo Methods

Difficulties of Inference

Iterative simulation adds two challenges to conducting Bayesian inference that we must address:

1. If iterations have not proceeded long enough, the simulations may grossly under represent the target density
   - Even if approximate convergence is reached, early draws may not reflect the target density
2. Draws based on iterative simulation have inherent within-sequence correlation
   - Inference from correlated draws is generally less precise than from the same number of independent draws

II.C Markov Chain Monte Carlo Methods

Additional notes:

- Serial correlation is not necessarily a problem since at convergence draws are i.d. and we ignore order when performing inference
- But such correlations can cause inefficiencies as longer chains may be needed to fully represent the posterior
- High correlation within-chain means the the chain is moving slowly and therefore not efficiently exploring the parameter space

II.C Markov Chain Monte Carlo Methods

We handle these issues in three ways:

1. Attempt to design simulation runs to allow for effective monitoring of convergence
    - In particular, we generate multiple chains with different starting points scattered throughout the parameter space

2. We monitor convergence of *all* quantities by comparing variation between and within chains
    - We must examine all since, by the nature of our algorithm, our primary parameter of interest is dependent upon other parameters

II.C Markov Chain Monte Carlo Methods

3 If the simulation efficiency is unacceptably low in the sense of requiring too much computation time to obtain approximate convergence, the algorithm can (and should) be altered

Now let's more directly address some of the problems that result from iterative sampling

II.C Markov Chain Monte Carlo Methods

Burn-in

To reduce the influence of the choice of starting values we can do two things: run multiple chains and mix across samplers or discard the first half (or more) of each chain.

The discarded portion of the chain is called the *burn-in*

Idea: discard samples prior to the iteration at which convergence is achieved

II.C Markov Chain Monte Carlo Methods

Thinning

Once approximate convergence has been reached, successive samples may still be highly correlated. As previously noted, this means the sampler can basically get stuck in one part of the parameter space. One way to help eliminate this is to keep every kth draw and discard the rest.

This process is called *thinning*

II.C Markov Chain Monte Carlo Methods

Multiple Chains

The best approach to assessing convergence is to run multiple chains each with different, over-dispersed starting values. We can then compare these chains to assess convergence, for example, the Gelman-Rubin diagnostic does this.

Scalar Estimands

We can also monitor scalar estimands as discussed in Unit A with a running mean plot

II.C Markov Chain Monte Carlo Methods

Splitting Chains

One approach we can use to generate multiple chains is to split a chain, after burn-in, in half. In general, suppose there are m chains after splitting. Provided we always use at least two starting values, then m is always at least 4.

Example II.12

Suppose we use five different starting values and thus start with five chains each of length 1000. Our burn-in discards the first 500, so we are left with five chains of length 500. We then split the chains in half giving $m = 10$ each with length 250.

II.C Markov Chain Monte Carlo Methods

Recall that we want to monitor convergence by comparing the variation between chains to the variation within

Convergence is achieved, roughly, when the within variation equals the between variation

The metric that is most useful for diagnosing this is the Gelman-Rubin

Let's take a moment to review slides 39—42

Example II.R12

Return to each of our examples: the Gibbs example from II.R7, the Metropolis-hastings example from II.R9, and the Gibbs-M-H from example II.R11. For each, generate one new chain. After discarding the burn-in, split each chain in half and run the Gelman-Rubin diagnostic.

Notes:

- The function `gelman.diag()` in the `coda` package can run this diagnostic, it requires an `mcmc.list()` object containing a list of `mcmc` objects

II.C Markov Chain Monte Carlo Methods

Stopping the Chain

Gelman recommends computing the scale reduction factor, $\hat{\text{R}}$ from the Gelman-Rubin diagnostic, for all scalar estimands of interest

If $\hat{\text{R}}$ is *not* near 1 for all parameters, continue running the simulation until $\hat{\text{R}}$ is close to 1

"Close to 1" depends on the problem, however a threshold of 1.1 is often sufficient

Thus we could monitor our sampler and stop it at the point this threshold is reached

II.C Markov Chain Monte Carlo Methods

Question: Do we only need to let the chains run long enough to obtain $\hat{R} < 1.1$ for all parameters?

II.C Markov Chain Monte Carlo Methods

Question: Do we only need to let the chains run long enough to obtain $\hat{R} < 1.1$ for all parameters?

Yes!

II.C Markov Chain Monte Carlo Methods

Question: Do we only need to let the chains run long enough to obtain $\hat{R} < 1.1$ for all parameters?

Yes!

But once that threshold is reached, we can treat any additional draws as being from the target distribution

II.C Markov Chain Monte Carlo Methods

Question: Do we only need to let the chains run long enough to obtain $\hat{R} < 1.1$ for all parameters?

Yes!

But once that threshold is reached, we can treat any additional draws as being from the target distribution

Which can help improve estimates

II.C Markov Chain Monte Carlo Methods

Question: Do we only need to let the chains run long enough to obtain $\hat{R} < 1.1$ for all parameters?

Yes!

But once that threshold is reached, we can treat any additional draws as being from the target distribution

Which can help improve estimates

Aim to take a long enough chain so that there are many samples available after threshold has been reached

II.C Markov Chain Monte Carlo Methods

Metropolis-Hastings Based GLM

The normal approximation approach relies on the fitting of a GLM model using Fisher's scoring but a more Bayesian approach would be to fully specify the likelihood and implement an M-H algorithm to generate posterior samples

Given that the link function linearizes the relationship between η_i and μ_i, it is reasonable to assume the posterior of the β's will be symmetric and unimodal (indeed GLM theory suggests)

II.C Markov Chain Monte Carlo Methods

Thus a reasonable proposal is the normal distribution centered at the previous step of β with variance either fixed at the last working covariance from the Fisher's scoring or updated as a working covariance amongst the coefficients

Additionally, a tuning parameter can be added to the variance of the proposal to help tune the acceptance rate

II.C Markov Chain Monte Carlo Methods

Logistic Regression via M-H

Suppose $y_i \sim Bern(\theta_i)$ where $logit(\theta_i) = \beta_0 + \beta_1 x_{1i}$

The likelihood for this model is then

$$\mathcal{L}(y_i|\beta, x_i) \propto \exp\left\{\sum_{i=1}^{n}\left[y_i(x_i'\beta) - \log\left(1 + e^{x_i'\beta}\right)\right]\right\}$$

for $\beta = \begin{bmatrix} \beta_0 & \beta_1 \end{bmatrix}'$ and $x_i' = \begin{bmatrix} 1 & x_{1i} \end{bmatrix}$

If we take the joint prior on β to be flat, i.e. $\pi(\beta) \propto 1$, the posterior will then have the same form

II.C Markov Chain Monte Carlo Methods

Thus the posterior is

$$P(\beta|y_i, x_i) \propto \exp\left\{\sum_{i=1}^{n}\left[y_i(x_i'\beta) - \log\left(1 + e^{x_i'\beta}\right)\right]\right\}$$

A reasonable choice for the proposal is then

$$J(\beta^*) \sim MVN(\beta^{(t-1)}, \tau V_\beta)$$

that is, the proposal is centered at the previous step of β with variance based on the last working variance from Fisher's scoring and a tuning parameter

II.C Markov Chain Monte Carlo Methods

Code for implementing a M-H based logistic regression is available on the website

Example II.R13

Consider again the data from Old Faithful to predict whether or not an eruption will last longer than three minutes based on how long it's been since the last eruption

Use the tuning parameter to obtain a reasonable acceptance rate and compare the results to the normal approximation approach

II.C Markov Chain Monte Carlo Methods

Poisson Regression via M-H

Suppose $y_i \sim \mathrm{Poisson}(\mu_i)$ where $\log(\mu_i) = \beta_0 + \beta_1 x_{1i}$

The likelihood for this model is then

$$\mathcal{L}(y_i|\beta, x_i) \propto \exp\left\{ \sum_{i=1}^{n} y_i x_i'\beta - e^{x_i'\beta} \right\}$$

again for $\beta = \begin{bmatrix} \beta_0 & \beta_1 \end{bmatrix}'$ and $x_i' = \begin{bmatrix} 1 & x_{1i} \end{bmatrix}$

As with the logistic model, if we take the joint prior on β to be flat, i.e. $\pi(\beta) \propto 1$, the posterior will then have the same form

II.C Generalized Linear Models

Code for implementing a M-H based Poisson regression is available on the website

Example II.R14

Consider again the data from the clinical trial on the seizure drug progabide. Use the M-H approach to generate posterior samples for the model with treatment, baseline number of seizures, and age as covariates.

Use the tuning parameter to obtain a reasonable acceptance rate and compare the results to the normal approximation approach

II.C Markov Chain Monte Carlo Methods

The basic Gibbs sampler and M-H algorithm are powerful tools that can be used to implement a wealth of statistical models

But both can also be seen as building blocks for more advanced Markov chain simulation algorithms

We will now discuss computationally efficient approaches to Markov chain simulation

Unit D: Efficient Markov Chain Simulation

Chapter 12

II.D Efficient Markov Chain Simulation

We begin this Unit by discussing ways to improve efficiency in simulation techniques we've already discussed, namely

1) Efficient Gibbs Sampler
2) Efficient M-H Jumping Rules

We will then discuss extensions to the Gibbs and M-H and close with a brief introduction to Hamiltonian Monte Carlo

II.D Efficient Markov Chain Simulation

1) Efficient Gibbs Sampler

To improve efficiency in the Gibbs, we can consider three approaches:

First, we can *transform and re-parameterize* model components

Second, we can use *auxiliary variables* or *data augmentation*

Third, we can use *parameter expansion*

II.D Efficient Markov Chain Simulation

Transformation and Re-parameterization

The Gibbs sampler is most efficient when parameterized in terms of independent components

Even using a Gibbs when components are highly dependent will result in slow convergence

The simplest way to re-parameterize is by a linear transformation of the parameters, though posteriors that are not approximately normal may require special methods

We also saw that a univariate transformation can reduce model complexity

II.D Efficient Markov Chain Simulation

Auxiliary Variables

Gibbs sampler computations can often be simplified or convergence accelerated by *auxiliary variables* which are variables we do not observe realizations from but, when added to a model, can help speed up computation

The idea of adding variables is also called *data augmentation*

II.D Efficient Markov Chain Simulation

We explore *auxiliary variables* via two examples

Example II.13

Suppose we wish to conduct inference for the parameters μ and σ^2 given n independent data points from the $t_\nu(\mu, \sigma^2)$ where we assume a uniform prior on μ and σ^2, i.e. $p(\mu, \sigma^2) \propto (\sigma^2)^{-1}$. First note that the t distribution can be represented as a mixture of normal distributions.

Thus the t likelihood can be represented as a mixture of n normals

II.D Efficient Markov Chain Simulation

Example II.12 (cont.)

For $i = 1, \ldots, n$, the t likelihood is then equivalent to the following latent parameter model

$$
\begin{aligned}
y_i &\sim N(\mu, V_i) \\
V_i &\overset{iid}{\sim} \mathrm{IG}\left(\frac{\nu}{2}, \frac{\nu}{2}\sigma^2\right)
\end{aligned}
$$

where the $V_i's$ are auxiliary variables that cannot be directly observed

If we perform inference using the joint posterior and then just consider the simulations for μ and σ^2, these will represent the posterior distribution the original t model

II.D Efficient Markov Chain Simulation

Example II.12 (cont.)

There is no direct way of sampling μ and σ^2 in the t model, but it is straight-forward using the Gibbs sampler on V, μ, σ^2 in the augmented model

We can produce estimates using a three step Gibbs:

1. Sample from full conditional of each V_i
   - Repeat this step n times
2. Sample from full conditional of μ
3. Sample from full conditional of σ^2

II.D Efficient Markov Chain Simulation

Step 1: Full conditional of each V_i

Conditional on the data y and the other parameters of the model, eachV_i is a normal variance parameter with a inverse gamma prior, thus its posterior is the same

$$V_i|\mu, \sigma^2, \nu, y \sim \mathrm{IG}\left(\frac{\nu+1}{2}, \frac{\nu\sigma^2 + (y_i - \mu)^2}{2}\right)$$

The n parameters V_i are independent in their conditional posterior distribution

II.D Efficient Markov Chain Simulation

Step 2: Full conditional of μ

Information on μ actually only comes from the y_i and V_i. Further, μ is normally distributed with the following form:

$$\mu|\sigma^2, V, \nu, y \sim N(\mu_p, \sigma_p^2)$$

where

$$\mu_p = \frac{\sum_{i=1}^n y_i/V_i}{\sum_{i=1}^n V_i^{-1}} \text{ and } \sigma_p^2 = \frac{1}{\sum_{i=1}^n V_i^{-1}}$$

II.D Efficient Markov Chain Simulation

Step 3: Full conditional of σ^2

Information on σ^2 comes only from V_i with conditional posterior

$$p(\sigma^2|\mu, V, \nu, y) \propto (\sigma^2)^{-1} \prod_{i=1}^{n} \sigma^{\nu} \exp\left[-\frac{\nu\sigma^2}{2V_i}\right]$$

Which can be shown to be a $Gamma$:

$$\sigma^2|\mu, V, \nu, y \sim Gamma\left(\frac{n\nu}{2}, \frac{\nu}{2}\sum_{i=1}^{n}\frac{1}{V_i}\right)$$

II.D Efficient Markov Chain Simulation

The previous example makes use of a well known mixing distribution that allows us to more easily use a t-distribution

Normal-based Mixing Distribution

Let ξ be represented as a scale mixture of a standard normal Z, i.e. $\xi = \sqrt{W}Z$

It is well known that if $W \sim \mathrm{IG}$, then $\xi \sim \mathrm{t}$

Thus using/placing a normal likelihood or prior along with an IG prior or hyper-prior on the variance is the same as t

II.D Efficient Markov Chain Simulation

Less well known though equally as useful is when $W \sim \mathcal{Gamma}$, then $\xi \sim \mathcal{Laplace}$ (also called Double-Exponential)

Thus the Laplacian can be used more readily as both a likelihood and a prior if the variance is given a $\mathcal{Gamma}$ prior

See Ding and Blitzstein (2017) for a not-all-that-awful proof

Under certain parameterizations of the Gamma, closed form conditionals can be found

II.D Efficient Markov Chain Simulation

Latent Variable Models

Albert and Chib (1993) proposed a latent variable (or auxiliary variable or data augmentation) representation for bernoulli and multinomial data that is equivalent to using the probit link in the GLM framework

The idea is to assume that the categorical y_i is the observable part of some underlying latent and normally distributed process

We begin with the case where y_i is binary

II.D Efficient Markov Chain Simulation

Let z_i denote subject i's latent normal response, then

$$y_i = \begin{cases} 0 & \text{if } z_i \leqslant 0 \\ 1 & \text{if } z_i > 0 \end{cases}$$

We then assume z_i is normally distributed mean $x_i'\beta$ and variance 1, thus

$$z_i \sim N(x_i'\beta, 1)$$

Note that z_i is never observed, but in using this representation, we've turned the likelihood into a mixture of normals

Further, assuming z_i is normal gives the Probit model

II.D Efficient Markov Chain Simulation

Identifiability

The latent variable representation is invariant to shift and scale, thus the intercept of the model and the variance of z_i are not (typically) identifiable

Typically, we assume the variance of z_i to be 1 and set the coefficient for the intercept, β_0, to 0

Note: Albert and Chib (1995) do suggest that updating the variance of z_i can improve model fit

II.D Efficient Markov Chain Simulation

Latent Variable Likelihood

The likelihood for a single subject, i, using the Albert and Chib approach is

$$\mathcal{L}(y_i|z_i, x_i, \beta) \propto \{1(z_i > 0)1(y_i = 1) + 1(z_i \leqslant 0)1(y_i = 0)\} \times \exp\left[-\frac{1}{2}(z_i - x_i'\beta)^2\right]$$

Note the likelihood for all subjects is just the product

II.D Efficient Markov Chain Simulation

Latent Variable Posterior

Assume a prior on β of $\pi(\beta)$ (typically a non-informative one), and using the full likelihood, the posterior is

$$P(\beta, z_i | y_i, x_i) \propto \pi(\beta) \prod_{i=1}^{n} \{1(z_i > 0)1(y_i = 1) + 1(z_i \leqslant 0)1(y_i = 0)\}$$
$$\times \exp\left[-\frac{1}{2}(z_i - x_i'\beta)^2\right]$$

From this, we can obtain samples of β and z_i using a Gibbs Sampler

II.D Efficient Markov Chain Simulation

Example II.14

Assume a flat prior on β, thus $\pi(\beta) \propto 1$. Working from the posterior below, we will derive the full conditionals of β and z_i.

$$P(\beta, z_i | y_i, x_i) \propto \pi(\beta) \prod_{i=1}^{n} \{1(z_i > 0)1(y_i = 1) + 1(z_i \leqslant 0)1(y_i = 0)\}$$
$$\times \exp\left[-\frac{1}{2}(z_i - x_i'\beta)^2\right]$$

Question: Are the full conditionals recognizable?

II.D Efficient Markov Chain Simulation

Full Conditionals for Binary Probit

The full conditionals we found in Example II.14 are

$$\beta|y, Z \sim MVN\left[\hat{\beta}_z, (X'X)^{-1}\right] \text{ for } \hat{\beta}_z = (X'X)^{-1}X'Z$$

To update the latent variables, we take case of y_i:

$$\begin{aligned} z_i|y_i = 0, \beta &\sim N(x_i'\beta, 1) \cdot 1(z_i \leqslant 0) \\ z_i|y_i = 1, \beta &\sim N(x_i'\beta, 1) \cdot 1(z_i > 0) \end{aligned}$$

Note: the full conditionals of the z_i's are truncated normals

II.D Efficient Markov Chain Simulation

Code for the Albert and Chib sampler can be found on the course webpage

Example II.R15

Load the dataset `trade.union` from the package `SemiPar` which contains information on 534 U.S. workers. We want to build a model to predict membership in a trade union (`union.member`) using as covariates years of education (`years.educ`), years of experience (`years.experience`), and gender (`female`).

Notes:

- Compare the results to the coefficients produced by the GLM
- See if you can estimate the intercept

II.D Efficient Markov Chain Simulation

Multinomial Outcomes

Albert and Chib (1993) also derived models for the two kinds of multinomial data we could encounter: nominal and ordinal

Recall that in the multinomial model, each y_i can take on a value from one of J categories for $J > 2$

When the categories have a natural ordering, they are said to be *ordinal*, otherwise they are said to be *nominal*

II.D Efficient Markov Chain Simulation

Ordinal Latent Variable Model

Suppose y_i can take on the ordered values $j = 1, \ldots, J$, the latent variable representation is then

$$y_i = \begin{cases} 1 & \text{if } \gamma_0 < z_i \leqslant \gamma_1 \\ 2 & \text{if } \gamma_1 < z_i \leqslant \gamma_2 \\ & \vdots \\ j & \text{if } \gamma_{j-1} < z_i \leqslant \gamma_j \\ & \vdots \\ J & \text{if } \gamma_{J-1} < z_i \leqslant \gamma_J \end{cases}$$

where $z_i \sim N(x_i'\beta, 1)$ and the cut-points γ_0 and γ_J are typically taken to be $-\infty$ and ∞, respectively

II.D Efficient Markov Chain Simulation

Ordinal Likelihood

Similar to the binary case, the ordinal likelihood for a single subject can now be represented as a mixture of normals:

$$\mathcal{L}(y_i|\beta, z_i, x_i) \propto \left\{ \sum_{j=1}^{J} 1(\gamma_{j-1} < z_i \leqslant \gamma_j) 1(y_{2i} = j) \right\}$$
$$\times \exp\left[-\frac{1}{2}(z_i - x_i'\beta)^2 \right]$$

where again the full likelihood is simply the product

II.D Efficient Markov Chain Simulation

Ordinal Posterior

Assuming a prior on β of $\pi(\beta)$ and taking the product, the posterior for the Albert and Chib Ordinal model is

$$p(\beta, z_i | y_i, x_i) \propto \pi(\beta) \prod_{i=1}^{n} \left\{ \sum_{j=1}^{J} 1(\gamma_{j-1} < z_i \leqslant \gamma_j) 1(y_{2i} = j) \right\}$$
$$\exp\left[-\frac{1}{2}(z_i - x_i'\beta)^2 \right]$$

Once again, estimates from this model can be drawn using a Gibbs sampler, however we must now also sample the γ_j's

See Albert and Chib (1993) for more details

II.D Efficient Markov Chain Simulation

Nominal Model

As previously discussed, Albert and Chib also derived the model for a nominal outcome

However, this model is a more complicated extension of the binary case than the ordinal case as it involves latent vectors for each subject, Z_i which are jointly assumed to follow a multivariate normal distribution with variance equal to $I_{n \times n} \otimes \mathbf{\Sigma}$

For supplemental reading, Albert and Chib (1993) is available alongside this note set

Parameter Expansion

For some problems, the Gibbs sampler can be slow to converge because of posterior dependence among parameters that cannot simply be resolved with a linear transformation

Somewhat paradoxically, adding an additional parameter and thus increasing the space for the random walk, can improve convergence

II.D Efficient Markov Chain Simulation

As with *auxiliary variables*, we illustrate *parameter expansion* with the t example

Example II.15

In the latent-parameter representation of the t model, convergence will be slow if a simulation draw of σ is close to zero as this will cause the sampled V_i's to be near zero

We can speed up convergence by adding a new parameter whose only role is to allow the Gibbs sampler to move in more directions and thus avoid getting stuck

II.D Efficient Markov Chain Simulation

Example II.15 (cont.)

Once again, we express the t likelihood as a mixture of normals but with an additional parameter. The expanded latent model is then

$$y_i \sim N\left(\mu, \alpha^2 U_i\right)$$
$$U_i \sim \mathrm{IG}\left(\frac{\nu}{2}, \frac{\nu}{2}\tau^2\right)$$

for $\alpha > 0$ which can be viewed as an additional scale parameter

As α has no meaning on its own, we assign it a noninformative prior, $\pi(\alpha) \propto \alpha^{-1}$

II.D Efficient Markov Chain Simulation

Example II.13 (cont.)

The Gibbs sampler for the expanded model now has four steps:

1. Sample from full conditional of each U_i
    - Once again, we repeat this step n times
2. Sample from full conditional of μ
3. Sample from full conditional of τ^2
4. Sample from full conditional of α^2

II.D Efficient Markov Chain Simulation

Step 1: Full conditional of each U_i

For each i, U_i is updated from a inverse gamma, similar to V_i from the auxiliary model:

$$U_i | \alpha, \mu, \tau^2, \nu, y \sim \mathrm{IG}\left(\frac{\nu + 1}{2}, \frac{\nu\tau^2 + [\{y_i - \mu\}/\alpha]^2}{2} \right)$$

The n parameters U_i are independent in their conditional posterior distribution and can be updated separately

II.D Efficient Markov Chain Simulation

Step 2: Full conditional of μ

Once again, μ is normal conditional on other parameters, however now it depends on y, U_i, and α;

$$\mu | \alpha, \tau^2, U, \nu, y \sim N\left(\mu_E, \sigma_E^2\right)$$

where

$$\mu_E = \frac{\sum_{i=1}^{n} y_i / (\alpha^2 U_i)}{\sum_{i=1}^{n} (\alpha^2 U_i)^{-1}} \text{ and } \sigma_E^2 = \frac{1}{\sum_{i=1}^{n} (\alpha^2 U_i)^{-1}}$$

II.D Efficient Markov Chain Simulation

Step 3: Full conditional of τ^2

The variance parameter τ^2 is only dependent upon n, ν, and U_i and, similar to the auxiliary model, has a $Gamma$ full conditional

$$\tau^2|\alpha, \mu, U, \nu, y \sim Gamma\left(\frac{n\nu}{2}, \frac{\nu}{2}\sum_{i=1}^{n}\frac{1}{U_i}\right)$$

II.D Efficient Markov Chain Simulation

Step 4: Full conditional of α^2

Holding U_i fixed, we can see that α^2 is simple a normal variance parameter and thus its full conditional should be a inverse gamma, thus

$$\alpha^2|\mu, \tau^2, U, \nu, y \sim \mathrm{IG}\left(\frac{n}{2}, \frac{1}{2}\sum_{i=1}^{n}\frac{[y_i - \mu]^2}{U_i}\right)$$

which depends on n, U, y, and μ

II.D Efficient Markov Chain Simulation

Notes:

- The parameters α^2, U, τ in the expanded model are not identified in the that the data does not supply enough information to estimate each of them

- The model as a whole is identifiable as long as we monitor convergence of the summaries μ, $\sigma = \alpha\tau$, and $V_i = \alpha^2 U_i$

- If the only goal is inference for the original t model, we can simply save μ and σ from the simulations—in other words, we don't necessarily need to monitor the non-identifiable parameters

- Because α breaks the dependence between τ and the V_i's, this model converges more reliably

II.D Efficient Markov Chain Simulation

Transformation and re-parameterization also applies to Metropolis and M-H jumps

In a normal or approximately normal setting, the jumping kernel should ideally have the same covariance structure as the target density

This structure can be approximately estimated based on the normal approximation at the mode (see Chapter 13 for more details)

II.D Efficient Markov Chain Simulation

2) Efficient M-H Jumping Rules

For any given posterior distribution, the Metropolis-Hastings algorithm can implemented in an infinite number of ways

Even after re-parameterizing, there are still endless choices in jumping rules, i.e. in proposal densities

Choice of jumping rule can have an impact on convergence

II.D Efficient Markov Chain Simulation

There are two main classes of simple jumping rules:

1. Essentially random walks around the parameter space
   - Often normal proposal densities with mean equal to the current value and variance set to obtain efficient algorithms
2. Proposal densities are constructed to closely approximate the target density

The goal in the second approach is to accept as many draws as possible with the M-H acceptance step being used primarily to correct the approximation

II.D Efficient Markov Chain Simulation

It is difficult to come up with a general set of efficient jumping rules, but some results have been obtained for random walk jumping distributions that have been useful in many problems

We illustrate one of these results in an example

Example II.16

Suppose there are d parameters and the posterior distribution of $\theta = (\theta_1, \ldots, \theta_d)$ is multivariate normal (after transformation) with known covariance matrix Σ

II.D Efficient Markov Chain Simulation

Example II.16 (cont.)

Further suppose we take draws using the Metropolis algorithm with a normal jumping kernel, namely

$$\theta^* | \theta^{t-1} \sim N(\theta^{t-1}, c^2 \Sigma)$$

Among this class of jumping rules, the most efficient has scale $c \approx 2.4/\sqrt{d}$ (efficiency defined relative to independent sampling)

The efficiency here can be shown to be roughly $0.3/d$ as opposed to the Gibbs efficiency of $1/d$—because every d iterations, a new independent draw of θ would be created

II.D Efficient Markov Chain Simulation

We have characterized our M-H samplers in terms of the acceptance rate which ideally is between 0.23 (for $\mathrm{d} > 5$) and 0.44 (one dimension)

Adaptive Algorithm

Noting our desire to tune the acceptance rate suggests an adaptive simulation algorithm where the acceptance rate is monitored and the covariance altered accordingly to obtain a target rate

II.D Efficient Markov Chain Simulation

In general, an adaptive algorithm works as follows:

1. Start parallel simulations with a fixed algorithm
   - I.e. a Gibbs or M-H with a normal proposal
2. After some simulations, update the M-H jumping rule:
   (a) Adjust the covariance of J to be proportional to the current posterior covariance
   (b) Increase or decrease the scale of J if the acceptance rate is too low or too high

Gelman notes that even in this simple form, this algorithm can be useful for drawing posterior simulations from some problems with d ranging from 1 to 50 (yes, fifty)

II.D Efficient Markov Chain Simulation

Adaptive Algorithms

When an iterative simulation algorithm is tuned, care must be taken to avoid converging to the wrong distribution

If the updating rule depends on the previous step, the transition probabilities may be more complicated than stated in the M-H and iterations will not converge to the target distribution (in general)

To protect against this, run adaptive algorithms in two phases:

1. An *adaptive phase* where parameters are tuned as often as needed/desired
2. A *fixed phase* where the adapted algorithm is run long enough for approximate convergence

Only simulations from the fixed phase are used for inference

Unit E: Additional Topics in Regression

Chapter 14

II.E Additional Topics in Regression

More Bayesian Regression

Recall the likelihood and prior we use:

$$y \sim MVN\left(X\boldsymbol{\beta}, \sigma^2 I_{n\times n}\right)$$
$$p(\boldsymbol{\beta}, \sigma^2) \propto (\sigma^2)^{-1}$$

Checking the Posterior is Proper

With the improper joint prior, the posterior is proper under the following conditions:

- $n > k$
- $\mathrm{Rank}(X) = k$

II.E Additional Topics in Regression

Return and re-run the following example but before implementing the model, randomly hold out half of the dataset (the function `sample()` can be useful here)

Example II.R16

Download the dataset `hersreg.txt` from the data folder. Build a Bayesian linear regression model using change in total cholesterol, `chtchol` as the outcome and `treatment` as the primary covariate of interest. Also control for baseline systolic blood pressure, `sbp`, as well as statin use, `statins`.

Implement a Gibbs sampler to run this model with $\mathrm{B} = 10000$, use `rmvnorm()` from the `mvtnorm` package, and set the seed to 808

II.E Additional Topics in Regression

Prediction is a common goal of regression modeling

Posterior Predictive Simulation

Suppose we have a set of new data $\tilde{X}$ with which we wish to predict $\tilde{y}$

To draw from the posterior predictive distribution of $\tilde{y}$, we first draw $(\boldsymbol{\beta}, \sigma^2)$ from their joint using either the hierarchical model or the Gibbs sampler

Then, we draw

$$\tilde{y} \sim MVN\left(\tilde{X}\boldsymbol{\beta}, \sigma^2 I_{n\times n}\right)$$

Example II.R17

Returning to the `hersct.txt` data, use the holdout data to generate the posterior predictive distribution of $\tilde{y}$ using the seed 8675309. The function `rmvnorm()` may be inefficient, so see if an assumption we've made can streamline the code.

Generate a graph of the prediction error for the model, i.e. $y - \tilde{y}$

Question: Is the model a good fit for the data?

II.E Additional Topics in Regression

Model Checking and Robustness

The standard methods for assessing model fit, such as examining residual plots, can be directly interpreted as posterior predictive checks and conducted in an analogous manner

An additional advantage of the Bayesian model, however, is that we can generate the posterior predictive distribution for any data summary

II.E Additional Topics in Regression

Example II.17

Suppose we wish to assess the selection of the normal likelihood for our model. One approach is to try to assess the tails of the posterior predictive distribution and compare them to the tails of the observed data. If the tails in the observed data have more mass than the tails in the predictive distribution, the normal likelihood may not be a good fit for the model. There are several approaches we could take here:

1. Determine the proportion of outliers in the observed data and compare to the same from the posterior predictive distribution
2. Calculate the Kurtosis for the observed data and compare it to the posterior distribution of Kurtosis based off the model

Example II.R18

Now run the Gibbs sampler on the whole dataset and save the residuals. Simultaneously, generate a hypothetical replicate, $y^{\text{rep}} \sim MVN\left(X\boldsymbol{\beta}, \sigma^2 I_{n \times n}\right)$, using the given draw of $(\boldsymbol{\beta}, \sigma^2)$ and X at each step. Then run a regression of y^{rep} on X and save the residuals. Set the seed to 2335.

For the residuals based on the observed data, determine the proportion of subjects with $|e| > 0.2$. For the residuals based on y^{rep}, determine and plot the proportion of residuals greater than 0.2 in absolute value for each replicate. Compare.

II.E Additional Topics in Regression

Full details on the systematic component will largely be left to another class, however we will briefly consider some details related to assembling the design matrix

Identifiability and Collinearity

If the columns of X are not linearly independent, the regression parameters cannot be uniquely estimated, the data are said to be *collinear*

If the data are nearly collinear, then they supply little information about some linear combinations of the β's and transformation may be warranted

II.E Additional Topics in Regression

Nonlinear Relations

Once variables have been selected, it may make sense to transform them so that the relation between x and y is close to linear

Transformations such as logarithms and logits have been found useful in a variety of applications

Care must be taken as transformations change the interpretation of the regression coefficient

II.E Additional Topics in Regression

Indicator Variables

Inclusion of categorical predictors involves the construction of a set of indicators variables, i.e. binary variables that indicate group membership

If there are k levels to a categorical variable, we need to construct $k - 1$ indicator variables

These are particularly important as we are constructing a numerical matrix, so all categorical variables must be appropriately converted

II.E Additional Topics in Regression

Interactions

Occasionally, it is useful to see if the relationship between y and a variable x_i is modified by a second variable, x_j

The systematic component may then look like

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon_i$$

Considering the effect of x_1, we see it is now $\beta_1 + \beta_3 x_2$, thus the effect of x_1 is modified by x_2

To reduce collinearity, it is useful to center the variables first (this also improves interpretability)

II.E Additional Topics in Regression

Regularization and Dimension Reduction

Approaches such as stepwise regression and subset selection are traditional non-Bayesian methods for choosing a set of explanatory variables to include in a regression

Mathematically, not including a variable is equivalent to setting its coefficient to exactly zero

Thus, selection procedures in classical regression set coefficients to zero with some probability, however the procedures are heavily influenced by the quality of the data

II.E Additional Topics in Regression

In the Bayesian context, with many explanatory variables, each with a fair probability of being irrelevant to y, one can given each coefficient a prior distribution with a peak at zero but long tails

Example II.18

If we let each β_j come from a t distribution centered at zero

Such a prior says that each variable is probably unimportant, but if it has predictive power, it could be large

II.E Additional Topics in Regression

Regularization and Shrinkage

Regularization is a general term used for statistical procedures that give more stable estimates

Shrinkage is another term that might be applied as these procedures usually shrink coefficients toward zero

In the frequentist context, these terms applies to procedures like Ridge Regression or Lasso

The issue arises from the fact that for a large number of predictors, the least-squares estimates can be noisy

II.E Additional Topics in Regression

In the Bayesian context, we can use informative prior distributions to regularize our estimates

Three choices are involved in Bayesian regularization:

1. The location and scale of the prior
   - A more concentrated prior does more regularization
2. The analytic form of the prior
   - A normal pulls estimates toward the prior mean by a constant proportion
   - A Laplacian shifts estimates by a constant amount
   - A long-tailed distribution, does more regularization for coefficients near the mean and less for those that are far away
3. How the posterior is summarized

II.E Additional Topics in Regression

Let $\|\cdot\|_p$ denote the ℓ^p-norm, that is

$$\|\beta\|_p = \left(\sum_{i=1}^{n} |\beta|^p\right)^{1/p}$$

Classical Lasso

Classical Lasso is a penalized likelihood model using the ℓ^1-norm

Thus, the regression estimates from the lasso come from the following Lagrangian:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \|y - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}$$

II.E Additional Topics in Regression

Put another way, we minimize the following:

$$Q_{\ell^1} = (y - X\boldsymbol{\beta})'(y - X\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

where λ is a control or tuning parameter that guides the regularization

In the frequentist context, λ is typically selected via generalized cross validation

II.E Additional Topics in Regression

Bayesian Lasso

Considering the form of Q_{ℓ^1} above should suggest how to implement the lasso in the Bayesian context

In particular, the ℓ^1-norm should remind us of the exponent of the Laplacian distribution

Thus the Bayesian Lasso uses the prior $\beta_j \sim Laplace(0, \gamma)$ where

$$\pi(\beta_j) = \frac{1}{2\gamma} \exp\left(-\frac{1}{\gamma}|\beta_j|\right)$$

II.E Additional Topics in Regression

Example II.19

Let's now consider the resulting posterior when using the lasso prior, i.e. the Laplacian prior on β_j

Let $y \sim MVN\left(X\boldsymbol{\beta}, \sigma^2 I_{n\times n}\right)$ and place the Laplace prior on β_j using $\lambda = 1/\gamma$, the precision instead of the scale

Also use the non-informative prior on σ^2, $\pi(\sigma^2) \propto (\sigma^2)^{-1}$ and the non-informative hyper-prior on λ, $\pi(\lambda) \propto \lambda^{-1}$

Determine the posterior distribution and find the full conditionals

II.E Additional Topics in Regression

Ridge Regression

If instead of the ℓ^1-norm, we use the square of the ℓ^2-norm as our penalty, we get a Ridge Regression

Thus for Ridge Regression, we minimize

$$Q_{\ell^2} = (y - X\boldsymbol{\beta})'(y - X\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'D\boldsymbol{\beta}$$

Where D is a penalty matrix that dictates which coefficients get penalized (if it's all of them, $D = I_{p\times p}$)

In the frequentist setting, λ is determined via GCV or the penalty can be enforced using a mixed model formulation and REML

Bayesian Ridge Regression

Once again, we can use Q_{ℓ^2} as a guide for determining the form of Bayesian Ridge Regression

Suppose we wish to penalize all coefficients, thus D is the identity matrix and the penalty term is

$$\lambda \boldsymbol{\beta}' \boldsymbol{\beta} = \lambda \sum_{j=1}^{p} \beta_j^2$$

If we place a mean zero normal prior on β_j, we have the ridge prior

II.E Additional Topics in Regression

Example II.20

Let's now consider the resulting posterior when using the ridge prior, i.e. the mean zero normal prior on β_j

Let $y \sim MVN\left(X\boldsymbol{\beta}, \sigma^2 I_{n\times n}\right)$ and place the mean zero normal prior on β_j using $\tau^2 = 1/\gamma^2$, the precision instead of the variance

Also use the non-informative prior on σ^2, $\pi(\sigma^2) \propto (\sigma^2)^{-1}$ as well as $\pi(\tau^2) \propto (\tau^2)^{-1}$, the non-informative prior for τ^2

Determine the posterior distribution and find the full conditionals

II.E Additional Topics in Regression

Modeling Assumptions

Thus far, our regression models have made several assumptions including linearity of the expected value as a function of X, normality of the error terms, and independent observations with equal variance

Many of these assumptions are not, in practice, true

We now consider how to incorporate departures from these assumptions into our Bayesian regression models

II.E Additional Topics in Regression

Nonlinearity

The simplest approach to dealing with nonlinearity is to make appropriate transformations of the components of X

Transformations such as the log, $\sqrt{\cdot}$, square, etc. are often useful

Basis function representations and smoothed approaches may also be appropriate

If X is large, this may be infeasible or require more advanced techniques

II.E Additional Topics in Regression

Non-normality

The outcome, y, may not necessarily be normally distributed

Occasionally, a transformation of y can improve normality, for instance if y is skewed

If y is discrete, the problem may not be solvable via transformation and a generalized linear model may be more appropriate

Unequal Variances

Unequal variance of the regression errors can be detected in residual plots (this is the classic "fanning" pattern) which suggests assuming σ^2 may in fact vary by i

This issue can often be addressed by adding more explanatory variables

However, in some cases, the issue must be addressed with more a sophisticated modeling approach

II.E Additional Topics in Regression

Correlations

Correlations between $(y_i - X_i \boldsymbol{\beta})$ and $(y_i - X_i \boldsymbol{\beta})$, conditional on X and model parameters, can sometimes be detected by examining the correlation of residuals

If correlation exists in the data but is not included in the model, then the posterior inference about model parameters will typically be falsely precise

We will consider approaches to deal both with unequal variances and correlations, but first let's examine the OLS extension

Generalized Least Squares

Suppose our model errors, ϵ, have finite variance that is not the identity matrix scaled by σ^2 but something more general

That is, instead of $Var(\epsilon) = \sigma^2 I_{n \times n}$, $Var(\epsilon) = \Omega$

We would then minimize

$$Q_G = (y - X\boldsymbol{\beta})'\Omega^{-1}(y - X\boldsymbol{\beta})$$

to obtain estimates of $\boldsymbol{\beta}$

II.E Additional Topics in Regression

The Generalized Least Squares or GLS estimates are then

$$\hat{\beta}_G = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}y$$

where, it should be noted, Ω is potentially an unstructured covariance matrix

We can impose various structures on Ω depending on the nature of the data

Once again, the GLS estimates will show up in the Bayesian model

Bayesian Modeling of Unequal Variances and Correlated Errors

Unequal variances and correlated errors can be included in the linear model by allowing a data covariance matrix, $\boldsymbol{\Sigma}_y$, that is not necessarily proportional to the identity matrix:

$$y \sim MVN(X\boldsymbol{\beta}, \boldsymbol{\Sigma}_y)$$

Modeling and estimation are, in general, more difficult than in OLS

Importantly, the symmetric, positive definite $n \times n$ data variance matrix $\boldsymbol{\Sigma}_y$ must be specified in some fashion

Bayesian Regression with Known Covariance

We first consider the simplest case of unequal variances and correlated errors, where the variance matrix $\boldsymbol{\Sigma}_y$ is known

Further, we assume a non-informative uniform prior density for β

The posterior is nearly identical to the ordinary linear regression with known variance if we apply a simple linear transformation to X and y

II.E Additional Topics in Regression

Let $\mathbf{\Sigma}_y^{1/2}$ be the a Cholesky factor, i.e. an upper triangular 'matrix square root,' of $\mathbf{\Sigma}_y$

Multiplying both sides of the regression by $\mathbf{\Sigma}_y^{-1/2}$ yields the model

$$\mathbf{\Sigma}_y^{-1/2} y | \boldsymbol{\beta}, X \sim MVN\left(\mathbf{\Sigma}_y^{-1/2} X\boldsymbol{\beta}, I_{n \times n}\right)$$

In other words, we apply the Cholesky factor as our transformation to both X and y

Example II.21

It can be shown that in performing this transformation, the resulting posterior distribution of $\boldsymbol{\beta}$ is multivariate normal with mean $\hat{\boldsymbol{\beta}}$ and variance $V_{\boldsymbol{\beta}}$ where

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (X'\boldsymbol{\Sigma}_y^{-1}X)^{-1}X'\boldsymbol{\Sigma}_y^{-1}y\\ V_{\boldsymbol{\beta}} &= (X'\boldsymbol{\Sigma}_y^{-1}X)^{-1}\end{aligned}$$

This is relatively straight forward to show by assuming $\sigma^2 = 1$ and working with the likelihood based off the Cholesky-transformed X and y

II.E Additional Topics in Regression

Of course, assuming a known covariance matrix (even if based on the data) is problematic

For example, if we base the 'known' $\boldsymbol{\Sigma}_y$ on y alone, we are assuming $Var(y|X, \boldsymbol{\beta})$ is not dependent upon either X or $\boldsymbol{\beta}$

Thus it is useful to develop the Bayesian model for unknown covariance

II.E Additional Topics in Regression

Bayesian Regression with Unknown Covariance matrix

We now derive the posterior distribution when the covariance matrix is unknown

First, we assume $y \sim MVN(X\boldsymbol{\beta}, \boldsymbol{\Sigma}_y)$ where $\boldsymbol{\Sigma}_y$ is now unknown

We then use the noninformative joint prior $\pi(\boldsymbol{\beta}, \boldsymbol{\Sigma}_y) \propto \boldsymbol{\Sigma}_y^{-(p+\nu)/2}$ where p is the number of coefficients and ν is a desired degrees of freedom

II.E Additional Topics in Regression

Example II.22

Suppose we wish to implement a Gibbs sampler for obtaining estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$. Using the information on the previous slide, we now derive the full conditionals for the normal model with an unknown covariance matrix.

Question: What are the forms of the full conditionals?

Hint: we may need to recall the multivariate generalization of the Gamma and Inverse Gamma distributions

II.E Additional Topics in Regression

Weighted Least Squares

Suppose we impose some level of structure on $\boldsymbol{\Sigma}$, namely we still assume independence between elements of ϵ but now allow the variances to be dependent upon a subject-specific weight

Thus, the ith diagonal of $\boldsymbol{\Sigma}$ might have the form

$$\boldsymbol{\Sigma}_{ii} = \frac{\sigma^2}{w_i}$$

where $w_i, i = 1, \ldots, n$, are known weights and σ^2 is an unknown variance parameter

II.E Additional Topics in Regression

Weights for WLS commonly depend on various calculations from the model, usually finding fitted values from a standard deviation or variance function

The model is then iteratively re-weighted until a convergence criteria is met—an approach we could certainly use in the Bayesian context

But for now, we will assume these weights are fixed and known

Example II.23

Assume that $y_i \sim N(X_i\boldsymbol{\beta}, \sigma^2/w_i)$. Further, place a non-informative joint prior on $\boldsymbol{\beta}, \sigma^2$, namely $\pi(\boldsymbol{\beta}, \sigma^2) \propto (\sigma^2)^{-1}$

Determine the full posterior and derive the full conditionals for the Gibbs sampler

Question: How does this differ from the setting where we assume equal variances?

II.E Additional Topics in Regression

The WLS model assumes the variance of y_i is known up to a constant of proportionality

Further, we may not want to iterate our procedure to obtain the weights (or have to worry about picking them)

Thus a model that allows the variance components to vary more freely might be of interest

II.E Additional Topics in Regression

Estimating Unknown Variance Parameters

We can allow the variance components to completely by subject, thus we would model σ_i^2

We have some flexibility with how we apply this assumption

Most obviously we can let each individual observation have its own variance

Alternatively, we could let batches of observations share the same variance, say observations are divisible into G groups, then we could let the variance vary for $g = 1, \ldots, G$

Example II.24

Let $y_i \sim N(X_i\boldsymbol{\beta}, \sigma_i^2)$, thus the y_i's are independent but not identically distributed. We can place a joint non-informative prior on all the components of $\boldsymbol{\beta}$ and all the σ_i^2, namely

$$\pi(\boldsymbol{\beta}, \sigma_1^2, \ldots, \sigma_n^2) \propto \prod_{i=1}^{n} (\sigma_i^2)^{-1}$$

Determine the form of the posterior

<u>Question</u>: Can we identify the full conditionals?

II.E Additional Topics in Regression

One way to model covariance structure in regression is to use Hierarchical regression models which are useful as soon as there are predictors at different levels of variation

Example II.25

In studying scholastic achievement, we may have information about individual students (for example, family background), classroom-level information (teach characteristics), as well as information about the school (educational policy, type of neighborhood)

Another example of where hierarchical models naturally arise is in the analysis of data obtained by stratified, clustered, or otherwise repeated sampling

II.E Additional Topics in Regression

With predictors at multiple levels of variation, the assumption of exchangeability of units or subjects at the lowest level breaks down

Thus a common assumption is to assume the coefficients of regression model are exchangeable in groups or batches

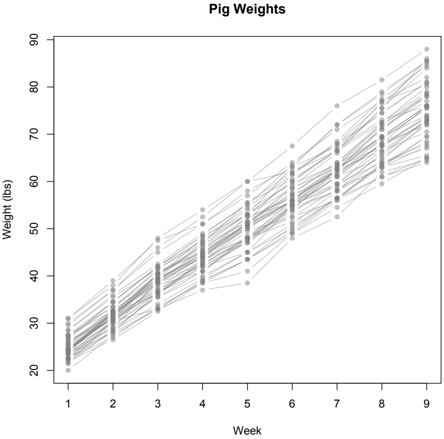
There are several approaches we could consider:

- Varying-coefficients models
- Intraclass correlation models
- Mixed-effects models

We will restrict our investigation to Mixed-effects models

II.E Additional Topics in Regression

To motivate our discussion of mixed models, let's consider a study of pig weight data:

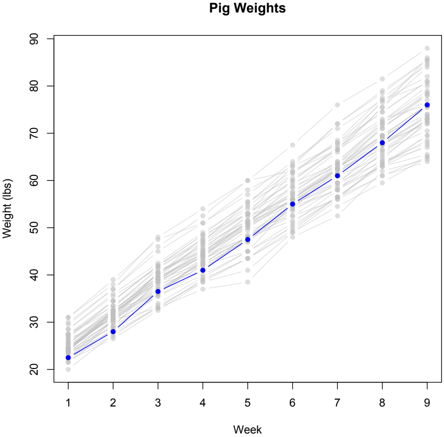


The study simply tracks the weight of each pig over nine weeks

II.E Additional Topics in Regression

There are 48 pigs in total and each has 9 measurements

For example, the third pig in the study is highlighted:



II.E Additional Topics in Regression

Individually, it appears that each pig's profile (the relationship between week, x, and weight, y) is roughly linear, however it's clear that additional variability has been introduced into the sample because of the repeated measures on each pig

Further, this data looks like a good candidate for a model with a subject specific intercept as it looks like each profile may in fact just be shifted vertically depending on the subject

Random Intercept Model

For subject i's jth observation, the random intercept model is defined as

$$y_{ij} = \beta_0 + x_{ij}\beta_1 + u_i + \epsilon_{ij}$$

where u_i is a subject-specific intercept and, in the pig weight example, x_i is the week at measurement occurrence j for pig i

We then assume that $u_i \sim N(0, \sigma_u^2)$ and that $\epsilon_{ij} \sim N(0, \sigma_e^2)$

II.E Additional Topics in Regression

Let n denote the number of subjects and N be the number of observations

In a balanced design, each subject has the same number of measurement occurrences which we denote with m

Thus, $N = n \cdot m$

In general, we could allow m to vary by subject, i.e. m_i

II.E Additional Topics in Regression

Example II.26

Let's consider the Bayesian approach to the random intercept model. First, denote the model in terms of vectors:

$$Y = X\boldsymbol{\beta} + ZU + E$$

where Y is the vectorized version of the matrix of outcomes, thus it is $N \times 1$. X is a matrix of covariates built by stacking the subject specific matrices on top of each other, it has dimension $N \times p$. U is an $n \times 1$ vector of subject-specific intercepts and Z is its $N \times n$ design matrix. Finally, E is the $N \times 1$ vector of model errors.

Example III.11 (cont.)

Y is assumed to be multivariate normal with mean $X\boldsymbol{\beta} + ZU$ and variance $(1/\tau_e^2)I_{N\times N}$. Further, place the noninformative prior on $\boldsymbol{\beta}$ and τ_e^2, $\pi(\boldsymbol{\beta}, \tau_e^2) \propto (\tau_e^2)^{-1}$

Next place a normal prior on u_i, specifically,

$$U \sim MVN(\mathbf{0}, (1/\tau_u^2)I_{n\times n})$$

Using a noninformative hyperprior on τ_u^2, determine the posterior and find the full conditionals for a Gibbs sampler

II.E Additional Topics in Regression

The full conditionals for the mixed effects model should have the following forms:

$$\beta|U,\tau_e^2,\tau_u^2,Y,X,Z \sim MVN\left[(X'X)^{-1}X'(Y-ZU),\frac{1}{\tau_e^2}(X'X)^{-1}\right]$$

$$U|\beta,\tau_e^2,\tau_u^2,Y,X,Z \sim MVN\left[(\tau_e^2 Z'Z+\tau_u^2 I_{n\times n})^{-1}Z'(Y-X\beta),(\tau_e^2 Z'Z+\tau_u^2 I_{n\times n})^{-1}\right]$$

$$\tau_e^2|U,\beta,\tau_u^2,Y,X,Z \sim Gamma\left[\frac{N}{2},\frac{1}{2}(Y-X\beta-ZU)'(Y-X\beta-ZU)\right]$$

$$\tau_u^2|U,\beta,\tau_e^2,Y,X,Z \sim Gamma\left(\frac{n}{2},\frac{1}{2}U'U\right)$$

which can be updated iteratively

The structure, complexity, and size of Z (and therefore U) can be increased to accommodate other kinds of mixed models

II.E Additional Topics in Regression

The form of Z ultimately dictates the nature of the hierarchy

And previously noted, it depends on what is considered random in the model

For the random intercept model, Z has a block diagonal form:

$$Z = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Example II.R19

Install and load the library `SemiPar`. Once loaded, enter the command `data(pig.weights)` into the console to load the pig weight data from the motivating example. Since the profiles are reasonably parallel, but shifted, a random intercept model should be appropriate for this data.

Sample code is available on the course page for this problem

Run this code and explore the resulting posterior estimates

II.E Additional Topics in Regression

For longitudinal models, the random intercept model is commonly used, despite potential issues with the induced covariance structure (namely, time-independent covariances, an uncommon feature of longitudinal data)

A model with both a random intercept and a random slope can be useful for models as well and will induce more reasonable covariance structures

However, we will leave random slopes for your own investigation (or take MATH 426 to learn about longitudinal models!)