STA 6360: Bayesian Methods in Statistics

The Laplace Approximation

1 Dealing with the Challenges of Non-Conjugate Priors

Conjugacy is one of the things that facilitates easy computations in Bayesian statistics; however, when encountering real problems, these families might not be the most appropriate models, nor do they always exist in a meaningful way. Consider this example:[1]

Example: (Tennis Serves) Consider data $\mathbf{X} = (X_1, \ldots, X_n)$ on the first serve success rates of a tennis player from n tournament matches. Consider the model

$$X_i \overset{\text{IID}}{\sim} g(x_i|\theta) = \theta(\theta+1)x_i^{\theta-1}(1-x_i),$$

With $x_i \in (0,1)$, and $\theta > 0$. No useful conjugate prior exists for this model. Suppose we choose to assign a $\mathsf{Gamma}(\alpha, \beta)$ prior on θ. Then the posterior $\pi(\theta|\mathbf{x})$ is

$$\begin{aligned}
\pi(\theta|\mathbf{x}) &\propto l(\theta|x)\pi(\theta) \\
&\propto \left\{\theta^n(\theta+1)^n \prod_{i=1}^n x_i^\theta\right\} \times \theta^{\alpha-1}e^{-\beta\theta}, \quad \theta > 0 \\
&= \theta^{n+\alpha}(\theta+1)^n \exp\left\{-\left(\beta + \sum_{i=1}^n \log\left(\frac{1}{x_i}\right)\right)\theta\right\}, \quad \theta > 0.
\end{aligned}$$

This, as you can see, is not a standard pdf. When $\pi(\theta|\mathbf{x})$ is not a standard pdf, it can be difficult to compute quanitles or sample from it. Hence, inference becomes difficult in these non-trivial scenarios. It is of interest if there exists a method of approximating the posterior using a trivial distribution.

2 Normal Approximation for the Posterior

2.1 The Proof

For any pdf that is smooth and well peaked around its point of maxima, Laplace proposed to approximate it by a normal pdf. It's a simple second order Taylor expansion trick on the log pdf. If θ denotes the point of maxima of a pdf $h(\theta)$, then it is also the point of maxima of the log-pdf $q(\theta) = \log h(\theta)$ and we can write:

$$\begin{aligned}
q(\theta) &\approx q(\hat{\theta}) + (\theta - \hat{\theta})q'(\hat{\theta}) + \frac{(\theta-\hat{\theta})^2 q''(\hat{\theta})}{2} \\
&= q(\hat{\theta}) + 0 + \frac{(\theta-\hat{\theta})^2 q''(\hat{\theta})}{2} \\
&= c - \frac{(\theta-\hat{\theta})^2 q''(\hat{\theta})}{2} \\
&= c - \frac{(\theta-\tilde{a})^2}{2\tilde{b}^2}.
\end{aligned}$$

[1] This example was taken from Surya Tokdar's notes on this topic, and is available here.

This equality holds if we choose $\tilde{a} = \hat{\theta}$ and $\tilde{b} = \{-q''(\hat{\theta})\}^{-1}$, as $q''(\hat{\theta}) < 0$. Note that the right hand side of the last display resembles the log-pdf of a $\mathsf{Normal}(\tilde{a}, \tilde{b}^2)$ distribution. So then $q(\theta) = \log h(\theta)$ can be approximated using this Taylor series expansion.

2.2 The Conditions

The beautiful thing is all that is now need to approximate the pdf is to know the location of the maxima ($\hat{\theta}$), and the curvature of the pdf at that point, $q''(\hat{\theta})$. There is no information required to compute the often intractable normalizing constant in Bayesian inference. Hence, the pdf and log pdf must satisfy sufficient regularity conditions, but beyond that, little information is necessary.

2.3 Applying the Approximation

Example: (Tennis Serves, continued.) Having the log posterior,

$$q(\theta) = \log \pi(\theta|\mathbf{x}) \propto (n + \alpha + 1) \log \theta + n \log(\theta + 1) - \theta \left\{ \beta - \sum_{i=1}^{n} \log x_i \right\}$$

we have that

$$q''(\theta) = -\frac{n + \alpha - 1}{\theta^2} - \frac{n}{(\theta + 1)^2}.$$

Suppose recorded data shows $n = 20$, $\sum_{i=1}^{20} \log X_i = -4.59$. Also suppose we work with $a = 1$, $b = 1$. So we can find the maxima $\hat{\theta}$ by solving $\dot{q}(\theta) = 0$, i.e.,

$$\frac{20}{\theta} + \frac{20}{\theta + 1} - 5.59 = 0,$$

which is solved at $\hat{\theta} = 6.69$. The curvature at the maximum equals $-\ddot{q}(6.69) = 0.785$. Hence

$$\pi(\theta|\mathbf{x}) \approx \mathsf{Normal}(6.69, 1/0.785) = \mathsf{Normal}(6.69, 1.129^2).$$

In this example we were able to solve for the maxima and curvature theoretically, and then use the Laplace approximation to approximate the posterior. In practice, one would find $\hat{\theta}$ numerically using a root solver such as Newton's method.

2.4 The Laplace Approximation in Multivariate Inference

Consider the scenario where you possess a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^m$.[2] We can still take the Taylor series expansion at the mode $\hat{\boldsymbol{\theta}}$ again,

$$\log \pi(\boldsymbol{\theta}|\mathbf{x}) \approx \log \pi(\hat{\boldsymbol{\theta}}) - \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})' \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}),$$

where $\mathbf{H}$ is the Hessian matrix of second-order partial derivatives which describes the local curvature of $\log \pi(\boldsymbol{\theta}|\mathbf{x})$ at $\hat{\boldsymbol{\theta}}$. It can be written as

$$\mathbf{H}_{ij} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} \log \pi(\boldsymbol{\theta}|\mathbf{x}) \Big|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}}.$$

Exponentiating, we have the kernel of a multivariate normal distribution

$$\pi(\boldsymbol{\theta}|\mathbf{x}) \propto \exp\left\{ -\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})' \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \right\},$$

which implies that $\pi(\boldsymbol{\theta}|\mathbf{x}) \approx \mathsf{Normal}_m(\hat{\boldsymbol{\theta}}, \mathbf{H}^{-1})$.

[2] All of the information in this section was taken from this blog post.

2.5 Assessing the Laplace Approximation

Although Laplace's approximation yields information about the *ability* to approximate the posterior, we can actually find information about the *quality* of the approximation. There fortunately exists machinery to guarantee that this approximation is very good when the prior pdf is smooth and n is large. Our machinery is the analogue to the asymptotic normality result for the MLE.

Theorem 1 (Bernstein-von Mises Theorem)**.** *Consider the model* $X_1, \ldots, X_n \overset{IID}{\sim} g(x_i \mid \theta)$, $\theta \in \Theta$. *Under some regularity conditions on the pdfs/pmfs* $g(\cdot \mid \theta)$*, including that all of them have the same support, and that for each* x_i, $\theta \mapsto \log g(x_i \mid \theta)$ *is twice continuously differentiable, we have that for any prior* $\pi(\theta)$ *which is positive, bounded, and twice differentiable over* Θ*,*

$$\sup_z \left| P(\theta \leq z \mid X = x) - \Phi\left(\{-\ddot{q}(\hat{\theta})\}^{1/2}(z - \hat{\theta})\right) \right| \approx 0$$

for all large n*.*

Under the same regularity condition it turns out that $\hat{\theta} \approx \hat{\theta}_{\text{MLE}}(x)$ and that $-\ddot{q}(\hat{\theta}) \approx I_{\text{OBS}}(x)$.

2.6 A Note on Conjugate Posteriors

The Bernstein-von Mises theorem is applicable to the majority of standard models for which a conjugate prior family exists, including well-established regular families such as binomial, Poisson, and exponential distributions. However, uniform distributions do not fall within this category. Consequently, for large sample sizes, the corresponding conjugate posterior distributions are expected to be approximately normal, a result that can be verified through case-specific analysis or alternatively, obtained through simpler approximations, such as normal approximation, without relying on the Laplace's technique. For example, for a binomial model $X \sim \mathsf{Binomial}(n, p)$ and a $\mathsf{Beta}(a, b)$ prior on $p \in (0, 1)$, the posterior is $\mathsf{Beta}(x + a, n - x + b)$. To approximate, we can use moment-matching setting $\hat{\mu} = (x + a)/(a + b + n)$ and $\hat{\sigma}^2 = \hat{\mu}(1 - \hat{\mu})/(a + b + n + 1)$.

2.7 Computation of A Laplace Approximation

2.7.1 Univariate Example: A Gamma-Poisson Model

Consider the model

$$\begin{aligned} Y \mid \mu &\sim \text{Poisson}(\lambda) \\ \mu &\sim \text{Gamma}(\alpha, \beta) \end{aligned}$$

where β is the scale parameter.[3] This conjugate family yields a $\mathsf{Gamma}(y + \alpha, 1 + 1/\beta)$ posterior. Suppose we observe $y = 2$, with a $\mathsf{Gamma}(3, 3)$ prior. The posterior and prior would be represented as in Figure 1, as the solid and dashed black curves. Conveniently, we have a closed form for the mode of this posterior, which is

$$\hat{\lambda} = \frac{y + \alpha - 1}{1 + 1/\beta},$$

which yields the mode of 3 in our example. The curvature is computed numerically, and the Laplace approximation can be visualized as the red curve in Figure 1. Of course, this Laplace approximation is done with only a single observation. One would expect the approximation to improve as the

[3]This example was taken from this blog post: https://bookdown.org/rdpeng/advstatcomp/laplace-approximation.html on November 12, 2024.

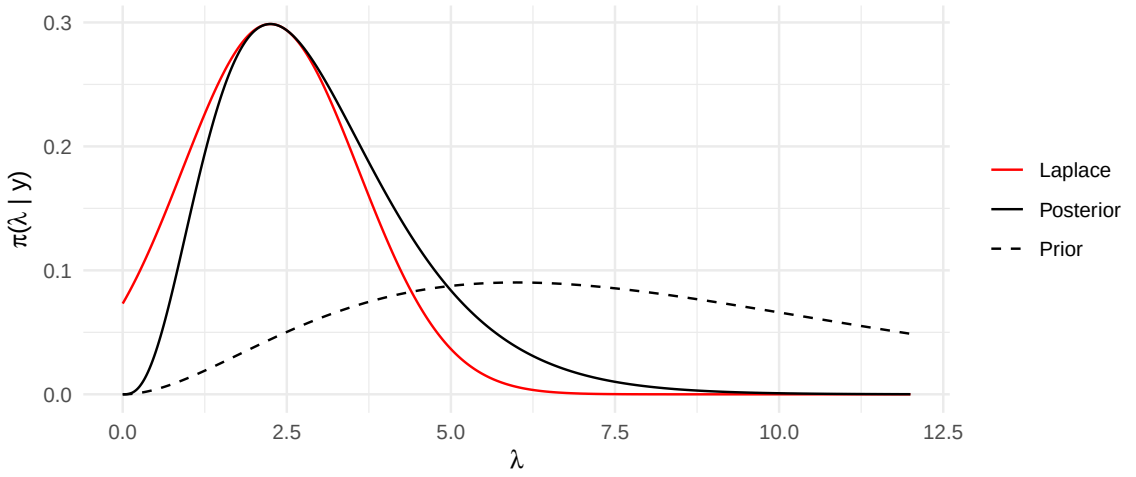

Figure 1: Prior and Posterior for Gamma-Poisson Model

sample size increases. In this case, with respect to the posterior mode as an approximation to the posterior mean, we can see that the difference between the two is simply

$$\hat{\lambda}_{\text{Mean}} - \hat{\lambda}_{\text{Mode}} = \frac{1}{n + 1/\beta}$$

which approaches 0 as $n \to \infty$.

2.7.2 Multivariate Example: Logistic Regression

Table 1 features a comparison of a Laplace approximation and an MCMC computation for a logistic regression model with five covariates.[4] We generated a data matrix $\mathbf{X}$ with to have standard Gaussian columns. This is not a benign choice as n gets big. With very high probability, the columns of will be almost orthonormal, which means that this is the best possible case for logistic regression. Generally speaking, design matrices from real data have a great deal of co-linearity in them and so algorithms that perform well on random design matrices may perform less well on real data. We fit $\beta_i \sim \mathsf{Normal}(0, 1)$ priors on all of the coefficient estimates. The model would be written as such:

$$\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6.$$

Coefficient	Mean	SE	Laplace Approx. Mean	Laplace Approx SE
(Intercept)	-0.1480	0.2217	-0.1456	0.2175
x_1	0.8198	0.4722	0.7807	0.4512
x_2	-0.8865	0.4565	-0.8476	0.4383
x_3	0.7451	0.4587	0.7102	0.4665
x_4	-0.6685	0.4579	-0.6212	0.4323
x_5	1.1573	0.4353	1.1070	0.4254

Table 1: Comparison of MCMC and Laplace approximation estimates.

[4]This example was commandeered from https://dansblog.netlify.app/posts/2024-05-08-laplace/laplace, and was transposed from Python into R using ChatGPT 4.0 (not 4o) on November 12, 2024.

As you can see from Table 1 the Laplace Approximation does a remarkable job approximating the coefficient estimates. Given computational capabilities, MCMC is preferred but in its absence, under moderate conditions, the Laplace approximation proves to be a viable option.

2.8 Code Appendix

For Section 2.7.1

```r
make_post <- function(y, shape, scale) {
  function(x) {
    dgamma(x, shape = y + shape, scale = 1 / (1 + 1 / scale))
  }
}

y <- 2
prior.shape <- 3
prior.scale <- 3
p <- make_post(y, prior.shape, prior.scale)

a <- prior.shape
b <- prior.scale
fhat <- deriv3(~ mu^(y + a - 1) * exp(-mu * (1 + 1/b)) / ((1/(1+1/b))^(y+a) * gamma(y + a)), "mu", f

post.shape <- y + prior.shape - 1
post.scale <- 1 / (length(y) + 1 / prior.scale)
pmode <- (post.shape - 1) * post.scale  # Assuming mode of posterior is used

# Define Laplace approximation function
lapprox <- Vectorize(function(mu, mu0 = pmode) {
  deriv <- fhat(mu0)
  grad <- attr(deriv, "gradient")
  hess <- drop(attr(deriv, "hessian"))
  f <- function(x) dgamma(x, shape = post.shape, scale = post.scale)
  hpp <- (hess * f(mu0) - grad^2) / f(mu0)^2
  exp(log(f(mu0)) + 0.5 * hpp * (mu - mu0)^2)
}, "mu")

# Define densities for plotting
x_vals <- seq(0, 12, length.out = 1000)
post_density <- dgamma(x_vals, shape = post.shape, scale = post.scale)
prior_density <- dgamma(x_vals, shape = prior.shape, scale = prior.scale)
laplace_approx <- lapprox(x_vals)

# Combine data into a data frame for ggplot
plot_data <- data.frame(
  x = x_vals,
  Posterior = post_density,
  Prior = prior_density,
  Laplace = laplace_approx
)

# Reshape data for ggplot
plot_data_long <- tidyr::pivot_longer(plot_data, cols = c("Posterior", "Prior", "Laplace"),
                                      names_to = "Density", values_to = "Value")
```

```r
# Plot with ggplot
ggplot(plot_data_long, aes(x = x, y = Value, color = Density, linetype = Density)) +
  geom_line() +
  scale_color_manual(values = c("Posterior" = "black", "Prior" = "black", "Laplace" = "red")) +
  scale_linetype_manual(values = c("Posterior" = "solid", "Prior" = "dashed", "Laplace" = "solid")) +
  labs(x = expression(lambda), y = expression(paste(pi, "(", lambda, " | y)"))) +
  theme_minimal() +
  theme(legend.title = element_blank())
```

For Section 2.7.2

```r
set.seed(1234)

# Load required packages
library("MASS") # for mvrnorm and matrix operations
library("rstanarm") # for Bayesian logistic regression
library("dplyr") # for data manipulation
library("tibble") # for tibble data structures

# Laplace Approximation Function
laplace <- function(f, x0) {
  # BFGS optimization using optim
  opt <- optim(x0, function(x) -f(x), method = "BFGS", hessian = TRUE)

  # Mode and Hessian
  mode <- opt$par
  H <- opt$hessian # Negative Hessian for Laplace approx

  list(mode = mode, H = H)
}

# Generate Data Function
make_data <- function(n, p) {
  X <- matrix(rnorm(n * p) / sqrt(p), nrow = n, ncol = p)
  beta <- 0.5 * rnorm(p)
  beta0 <- rnorm(1)
  y <- rbinom(n, 1, plogis(beta0 + X %*% beta))

  list(y = y, X = X)
}

# Log-posterior Function
log_posterior <- function(beta, X, y) {
  prob <- plogis(beta[1] + X %*% beta[-1])
  sum(y * log(prob) + (1 - y) * log(1 - prob)) - 0.5 * crossprod(beta)
}

# Set parameters and generate data
n <- 100
```

```r
p <- 5
data <- make_data(n, p)
y <- data$y
X <- data$X

# Run Laplace approximation
x0 <- rep(0, p + 1)
laplace_result <- laplace(\(beta) log_posterior(beta, X, y), x0)
post_mean <- laplace_result$mode
post_cov <- solve(laplace_result$H)

# Fit Bayesian Logistic Regression with rstanarm
logistic_reg <- stan_glm(y ~ X, family = binomial(link = "logit"), prior = normal(0, 1),
                         prior_intercept = normal(0, 1), chains = 4, iter = 2000)

# Extract posterior summaries
posterior_summary <- as.data.frame(summary(logistic_reg))
# Display results with correct alignment of laplace_mean and laplace_sd
posterior_summary <- posterior_summary |>
  rownames_to_column(var = "Variable") |>
  dplyr::select(Variable, mean, sd)

posterior_summary <- posterior_summary[-c(7:8),] |>
  mutate(
    laplace_mean = post_mean,
    laplace_sd = sqrt(diag(post_cov))
  )

posterior_summary |>
  `colnames<-`(c("Coefficient", "Mean", "SE",
                 "Laplace Approx. Mean", "Laplace Approx SE")) |>
  knitr::kable()
```