R Programming and Advanced Data Analysis

Bayesian Computing

Introduction

1. Approximating integrals
    a. Monte Carlo
    b. Importance sampling

2. Markov Chain Monte Carlo
    a. Gibbs sampler
    b. Metropolis-Hastings sampler

Monte Carlo

Approximating an Integral

- Suppose $\boldsymbol{X} = (X_1, \ldots, X_n)$ is an iid sample from Beta(a, b).
- We would like to compute the moment generating function

$$\theta = \int_0^1 e^{tx} \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)} dx.$$

- There are no unknown parameters involved, but exact calculation involves a difficult integration.
- However, we can use a Monte Carlo (MC) method to approximate θ, to any desired degree of accuracy.
- **Problem**: How many samples are required to construct a confidence interval for θ of width 0.1?

Approximating an Integral

- Let $g(x) = e^{tx}$ be the integrand.
- An MC estimate for θ is

$$\widehat{\theta} = \frac{1}{R}\sum_{r=1}^{R} g(X_r).$$

  By the Law of Large Numbers, $\widehat{\theta} \xrightarrow{\text{a.s.}} \theta$ as $R \to \infty$.

- We have $\text{Var}(\widehat{\theta}) = \frac{1}{R}\text{Var}[g(X)]$. Approximate $\sigma^2 = \text{Var}[g(X)]$ with

$$\hat{\sigma}^2 = \frac{1}{R}\sum_{r=1}^{R}\left(g(X_r) - \widehat{\theta}\right)^2.$$

- By the Central Limit Theorem

$$\frac{\sqrt{R}\left[\widehat{\theta} - E(\widehat{\theta})\right]}{\hat{\sigma}} \xrightarrow{\mathcal{L}} \text{N}(0,1).$$

- Using this, we can obtain an approximate $(1-\alpha)$ level CI for θ as

$$\widehat{\theta} \pm z_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{R}}.$$

Approximating an Integral

Algorithm

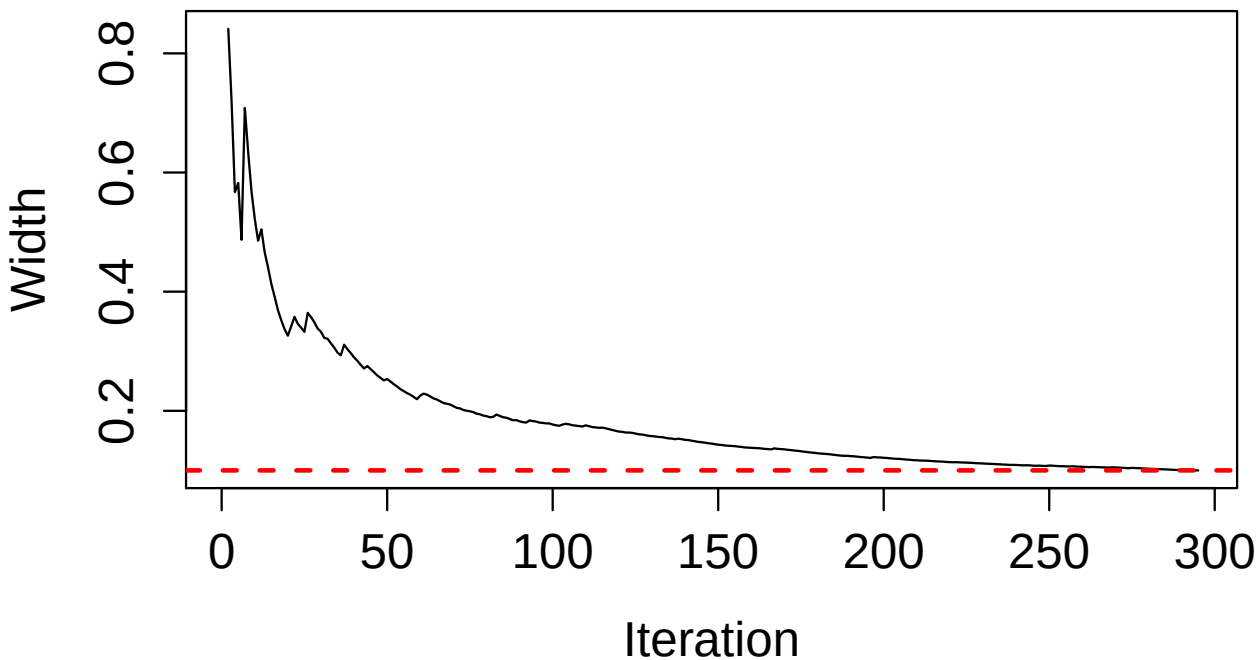
1. Sample X_j from Beta(a, b)
2. Until width of CI $= 2 \cdot z_{\alpha/2} \frac{\hat{\sigma}}{\sqrt{R}}$ is less than $\delta = 0.1$

```
t <- 1; a <- 2; b <- 1/2
g <- function(x) { exp(t*x) }

alpha <- 0.05
delta <- 0.1
z <- rbeta(1, a, b)
widths <- c(Inf)

while(tail(widths, 1) >= delta) {
    z <- c(z, rbeta(1, a, b))
    R <- length(z)
    est.mc <- mean( g(z) )
    var.mc <- 1/R * (R-1)/R * var(g(z))
    widths <- c(widths, 2 * qnorm(1 - alpha/2) * sqrt(var.mc))
}

ci <- est.mc + c(-1,1) * qnorm(1 - alpha/2) * sqrt(var.mc)
```

Approximating an Integral

```
> cat("Estimate is", est.mc, "\n")
> cat("CI = [", ci, "], width =", ci[2] - ci[1], "\n")
> cat("Number of samples needed:", length(z), "\n")

> plot(widths, type="l", xlab="Iteration", ylab="Width")
> abline(h = delta, lwd = 2, lty = 2, col = "red")
```

```
Estimate is 2.240291
CI = [ 2.190351 2.290232 ], width = 0.09988176
Number of samples needed: 295
```



Importance Sampling

Importance Sampling

- Importance Sampling is a technique to reduce the variance in Monte Carlo approximation (**??**).
- Let $\text{supp}(f) = \{x : f(x) \neq 0\}$ for a function f.
- Suppose we want to compute

$$\theta = \int_{\text{supp}(p)} g(x)p(x)dx$$

where p is a density ("the nominal distribution") and g is a given function.

Importance Sampling

- Suppose we have another density q ("the importance distribution") with $\text{supp}(g \cdot p) \subseteq \text{supp}(q)$. We may write

$$\begin{aligned}
\theta &= \int_{\text{supp}(p)} g(x)p(x)dx \\
&= \int_{\text{supp}(p)} \frac{g(x)p(x)}{q(x)} q(x)dx \\
&= \int_{\text{supp}(q)} \frac{g(x)p(x)}{q(x)} q(x)dx - \underbrace{\int_{\text{supp}(q)\setminus\text{supp}(p)} \frac{g(x)p(x)}{q(x)} q(x)dx}_{=0} \\
&= \mathrm{E}\left[\frac{g(X)p(X)}{q(X)}\right], \quad \text{where } X \sim q.
\end{aligned}$$

- Another Monte Carlo estimator is then

$$\hat{\theta}_q = \frac{1}{R}\sum_{r=1}^{R} \frac{g(X_r)p(X_r)}{q(X_r)}, \quad X_1, \ldots, X_R \overset{\text{iid}}{\sim} q.$$

- By the Law of Large Numbers, $\hat{\theta}_q \overset{a.s.}{\to} \theta$ as $R \to \infty$.

Importance Sampling

- Can also show that $\mathrm{E}(\hat{\theta}_q) = \theta$ and $\mathrm{Var}(\hat{\theta}_q) = \sigma_q^2/R$, where

$$\sigma_q^2 = \int_{\mathrm{supp}(p)} \frac{[g(x)p(x)]^2}{q(x)} dx - \theta^2$$

- To get a variance reduction, choose q so that it gives higher weight to the region of $\mathrm{supp}(p)$ where $|g(x)|p(x)$ is large.

- Also, q must give enough weight to regions where $|g(x)|p(x)$ is nonzero. I.e. tails of q cannot be too light.

- The optimal variance is achieved at

$$q^*(x) = \frac{|g(x)|p(x)}{\int_{\mathrm{supp}(p)} |g(z)|p(z)dz},$$

  but this result is usually not useful, since computing the denominator is as hard as the original integration problem.

Importance Sampling

- The variance σ_q^2 can be estimated by $\hat{\sigma}_q^2$, the sample variance of $g(X_r)p(X_r)/q(X_r)$ for $r = 1, \ldots, R$.
- The Central Limit Theorem gives that

$$\sqrt{R}\frac{(\hat{\theta}_q - \theta)}{\hat{\sigma}_q} \xrightarrow{\mathcal{L}} \mathsf{N}(0,1) \quad \text{as } R \to \infty.$$

- Using this, we can obtain an approximate $(1-\alpha)$ level confidence interval for θ as

$$\hat{\theta}_q \pm z_{\alpha/2}\frac{\hat{\sigma}_q}{\sqrt{R}}.$$

An Importance Sampling Example

- The following example is from **?**, Section 20.2.3.
- Suppose we would like to compute

$$\theta = \int_0^1 e^{-x^2/2} dx = \sqrt{2\pi}\, \mathrm{P}(0 < Z < 1), \quad \text{for } Z \sim \mathrm{N}(0,1).$$

- A simple/obvious Monte Carlo estimator is

$$\hat{\theta} = \frac{\sqrt{2\pi}}{R} \sum_{r=1}^{R} I(0 < Z_r < 1), \quad \text{for } Z_1, \ldots, Z_R \overset{\text{iid}}{\sim} \mathrm{N}(0,1)$$

- We have that

$$\sum_{r=1}^{R} I(0 < Z_r < 1) \sim \text{Binomial}(R, \theta)$$

$$\implies \text{Var}(\hat{\theta}) = \frac{2\pi}{R}\theta(1-\theta) \approx \frac{1.413}{R}$$

- It turns out that we can do much better...

An Importance Sampling Example

- Recall our objective

$$\theta = \int_0^1 e^{-x^2/2} dx = \sqrt{2\pi}\, \mathrm{P}(0 < Z < 1), \quad \text{for } Z \sim \mathrm{N}(0,1).$$

- A 2nd order Taylor series approximation around $x = 0$ yields

$$e^{x^2/2} \approx 1 + x^2/2 \quad \Longrightarrow \quad e^{-x^2/2} \approx \frac{1}{1 + x^2/2}$$

- The expression $1/(1 + x^2/2)$ can be turned into a density on $x \in (0, 1)$;

$$q(x) = \frac{1/(1 + x^2/2)}{\int_0^1 1/(1 + z^2/2) dz} = \frac{1/(1 + x^2/2)}{\sqrt{2}\arctan(1/\sqrt{2})}.$$

- Using q as the importance density, our estimator becomes

$$\hat{\theta}_q = \frac{1}{R} \sum_{r=1}^{R} \frac{g(X_r)p(X_r)}{q(X_r)} = \frac{1}{R} \sum_{r=1}^{R} e^{-X_r^2/2} \sqrt{2} \arctan(1/\sqrt{2})(1 + X_r^2/2),$$

  where $X_1, \ldots, X_R \overset{\text{iid}}{\sim} q$.

An Importance Sampling Example

- q is a nonstandard density, but we can sample from it using the inverse CDF method.
- The inverse CDF method can be used with continuous univariate distributions where the cumulative distribution function (CDF) is available.
- The CDF of q is

$$Q(x) = \int_0^x q(x)dx = \int_0^x \frac{1/(1+x^2/2)}{\sqrt{2}\arctan(1/\sqrt{2})}dx = \frac{\arctan(x/\sqrt{2})}{\arctan(1/\sqrt{2})}$$

- The inverse of $Q(x)$ is

$$Q^{-1}(u) = \sqrt{2}\tan\left(\arctan(1/\sqrt{2})\right)$$

- Taking $X = Q^{-1}(U)$ with $U \sim U(0,1)$, it can be shown that $X \sim q$.

An Importance Sampling Example

- The exact answer (computed numerically)

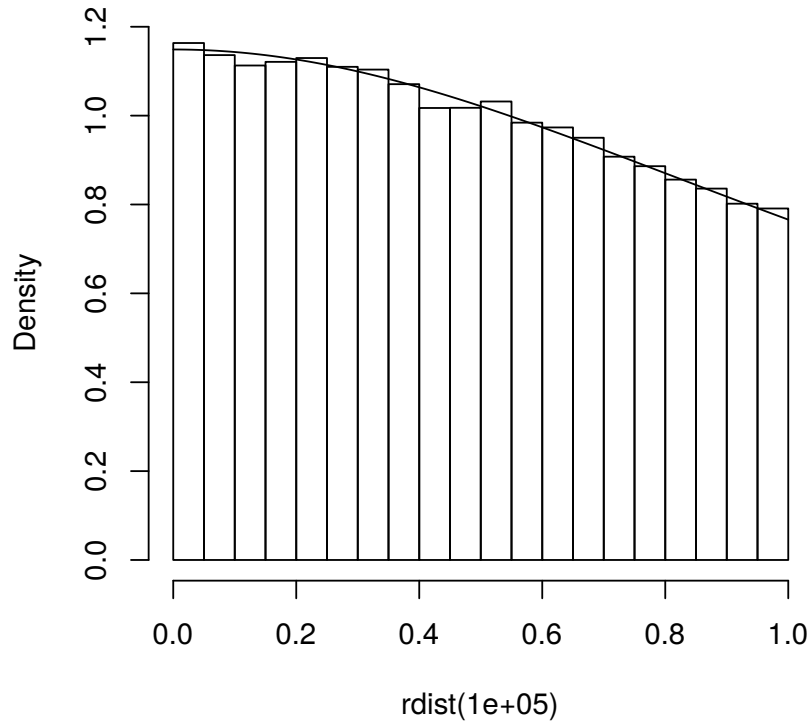
```
> sqrt(2*pi) * (pnorm(1) - pnorm(0))
[1] 0.8556244
```

- Simple Monte Carlo estimator. This time fix R.

```
g <- function(x) { sqrt(2*pi)*(x > 0 & x < 1) }
alpha <- 0.05
delta <- 0.1
R <- 10000

widths.simple <- numeric(R)
z <- rnorm(R)
for (r in 1:R){
    w <- g(z[1:r])
    est.mc <- mean(w)
    var.mc <- 1/r * (r-1)/r * var(w)
    widths.simple[r] <- 2 * qnorm(1 - alpha/2) * sqrt(var.mc)
}

ci <- est.mc + c(-1,1) * qnorm(1 - alpha/2) * sqrt(var.mc)
cat("Estimate is", est.mc, "\n")
cat("CI = [", ci, "], width =", ci[2] - ci[1], "\n")
```

```
Estimate is 0.8632828
CI = [ 0.839938 0.8866275 ], width = 0.04668948
```

```
# Density and drawing function for the truncated Cauchy-like
    distribution.
rdist <- function(n) {
    u <- runif(n)
    sqrt(2) * tan(u * atan(1/sqrt(2)))
}

ddist <- function(x) { 1 / ((1 + x^2/2) * sqrt(2) * atan(1 / sqrt(2))) }

# A quick test to make sure they are consistent with each other
hist(rdist(100000), freq = FALSE, main = NULL)
curve(ddist, add = TRUE)
```



An Importance Sampling Example

Importance sampling estimator

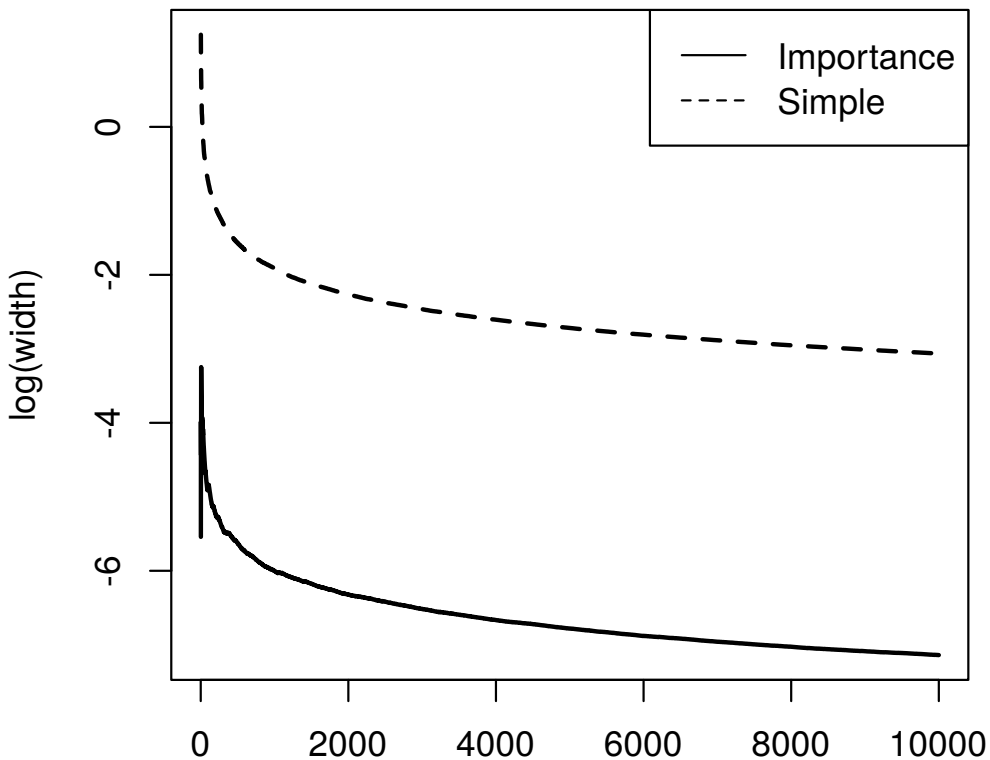
```
widths.imp <- numeric(R)
x <- rdist(R)
for (r in 1:R){
    w <- exp(-x[1:r]^2/2) * sqrt(2) * atan(1/sqrt(2)) * (1 + x[1:r]^2/2)
    est.mc <- mean(w)
    var.mc <- 1/r * (r-1)/r * var(w)
    widths.imp[r] <- 2 * qnorm(1 - alpha/2) * sqrt(var.mc)
}

ci <- est.mc + c(-1,1) * qnorm(1 - alpha/2) * sqrt(var.mc)
cat("Estimate is", est.mc, "\n")
cat("CI = [", ci, "], width =", ci[2] - ci[1], "\n")
cat("Number of samples used:", R, "\n")
```

```
> cat("Estimate is", est.mc, "\n")
Estimate is 0.8554805
> cat("CI = [", ci, "], width =", ci[2] - ci[1], "\n")
CI = [ 0.8550783 0.8558828 ], width = 0.0008045813
```

An Importance Sampling Example

Compare accuracy of simple and Importance Sampling estimator

```
ylim <- range(c(log(widths.simple), log(widths.imp)), na.rm = TRUE)
plot(log(widths.imp), type="l", lwd = 2, xlab="Iteration", ylab="log(
    width)",
         ylim = ylim)
lines(log(widths.simple), lty = 2, lwd = 2)
legend("topright", c("Importance", "Simple"), lty = c(1,2))
```



Markov Chain Monte Carlo

Markov Chain Monte Carlo

- Markov Chain Monte Carlo (MCMC) describes a class of algorithms which produce approximate draws from a given density $f(\boldsymbol{x})$.
- Often, the functional form of f is known except for a normalizing constant.
- Even when f is fully specified, may not be easy to draw directly from it.
- An MCMC algorithm produces a sequence $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ of dependent draws. The sequence is constructed in such a way that it "behaves like" draws from f more as we progress through the sequence.
- Technical justifications for MCMC are beyond the scope of the workshop, but are discussed in books **?** and **?**. A more applied book for practitioners is **?**.

Markov Chain Monte Carlo

- MCMC algorithms have found widespread use in the Bayesian paradigm of statistics.
- The fundamental assumption in Bayesian statistics is the framework

$$\begin{aligned} \boldsymbol{X} &\sim f(\boldsymbol{x} \mid \boldsymbol{\theta}) && \leftarrow \textbf{likelihood}, \\ \boldsymbol{\theta} &\sim f(\boldsymbol{\theta}) && \leftarrow \textbf{prior}. \end{aligned}$$

- The prior distribution represents our apriori belief about the parameter $\boldsymbol{\theta}$. Often, we have no idea, and the prior is chosen for mathematical convenience and/or to express our lack of knowledge.
- Using the likelihood, a prior, and observed data $\boldsymbol{x}$, all inference about $\boldsymbol{\theta}$ should be based on the **posterior** distribution

$$f(\boldsymbol{\theta} \mid \boldsymbol{x}) = \frac{f(\boldsymbol{x} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\boldsymbol{x})} \propto f(\boldsymbol{x} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})$$

- The distribution $f(\boldsymbol{\theta} \mid \boldsymbol{x})$ can sometimes be recognizable. E.g., sometimes $f(\boldsymbol{x} \mid \boldsymbol{\theta})$ and $f(\boldsymbol{\theta})$ may be chosen so that the posterior will be in the same family as the prior (i.e. *conjugate* prior).

Markov Chain Monte Carlo

- In most settings, $f(\boldsymbol{\theta} \mid \boldsymbol{x})$ will not have a recognizable form.
- Typically, the denominator of the posterior density,

$$f(\boldsymbol{x}) = \int f(\boldsymbol{x} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta}) d\theta,$$

  will be too complicated to evaluate analytically.
- MCMC algorithms allow us to generate draws $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots$ from the posterior without knowing the denominator.
- MCMC algorithms are also useful when $f(\boldsymbol{\theta} \mid \boldsymbol{x})$ is completely known, but does not have a familiar form.

Markov Chain Monte Carlo

- Another important distribution for Bayesians is the posterior predictive distribution,

$$f(\tilde{\boldsymbol{x}} \mid \boldsymbol{x}) = \int f(\tilde{\boldsymbol{x}} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta} \mid \boldsymbol{x}) d\theta,$$

  where $\tilde{\boldsymbol{x}}$ represents a new observation.

- The posterior predictive distribution allows inference on new observations through the posterior distribution.

- $f(\tilde{\boldsymbol{x}} \mid \boldsymbol{x})$ is usually too complicated to determine analytically. But we can generate draws from it by

$$\tilde{\boldsymbol{X}}^{(r)} \sim f(\tilde{\boldsymbol{x}} \mid \boldsymbol{\theta}^{(r)}),$$
$$\boldsymbol{\theta}^{(r)} \sim f(\boldsymbol{\theta} \mid \boldsymbol{x}).$$

  The draws $\boldsymbol{\theta}^{(r)}$, $r = 1, \ldots, R$, from the posterior are obtained using the MCMC sampler.

Probit Regression Application

Probit Regression Application

- Math Gym (`http://mathgym.umbc.edu`) is a program at UMBC which provides "workouts" to aid students in introductory math courses.
- **Question:** Is there evidence that it helps students pass the courses?
- Denote Y_i as the indicator of whether the ith student passed. Let $\boldsymbol{x}_i$ contain predictors computed from the course, section, FYI, # of days attending math gym, and Quiz Zero score.
- A Bayesian probit regression model is

$$\begin{aligned} Y_i &\sim \text{Ber}(p_i), \quad p_i = \Phi(\boldsymbol{x}_i^\top \boldsymbol{\beta}), && \leftarrow \textbf{likelihood}, \\ \boldsymbol{\beta} &\sim \text{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta), && \leftarrow \textbf{prior}. \end{aligned}$$

- Here, $\Phi(\cdot)$ is the CDF of $\text{N}(0,1)$; $\boldsymbol{\mu}_\beta$ and $\boldsymbol{\Sigma}_\beta$ are hyperparameters to describe our uncertainty of $\boldsymbol{\beta}$ before observing the data. We will take $\boldsymbol{\mu}_\beta = \mathbf{0}$ and $\boldsymbol{\Sigma}_\beta = \sigma_\beta^2 \boldsymbol{I}$ for some $\sigma_\beta^2 >> 1$.
- The joint distribution of $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ and $\boldsymbol{\beta}$ is

$$f(\boldsymbol{y}, \boldsymbol{\beta}) = \text{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta) \cdot \prod_{i=1}^{n} \left[\Phi(\boldsymbol{x}_i^\top \boldsymbol{\beta})\right]^{y_i} \left[1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\beta})\right]^{1-y_i}.$$

Probit Regression Application

- We will make use of the basis function decomposition

$$f(x) = \sum_{j=1}^{J} \beta_j h_j(x),$$

where $h_1(x), \ldots, h_J(x)$ are prespecified basis functions and $\beta_1, \ldots, \beta_J$ are unknown coefficients. Chapter 5 of **?** discusses the basics.

- We will specify K knot points $\xi_1 < \ldots < \xi_K$ so that the behavior of $f(x)$ can be modeled differently in the regions

$$(-\infty, \xi_1), (\xi_1, \xi_2), \ldots, (\xi_{K-1}, \xi_K), (\xi_K, \infty).$$

- To support a piecewise linear function, whose pieces are connected at the knots (i.e. continuous), an appropriate basis is

$$h_1(x) = 1,\ h_2(x) = x,$$
$$h_3(x) = (x - \xi_1) \cdot I(x > \xi_1),\ \ldots,\ h_{K+2}(x) = (x - \xi_K) \cdot I(x > \xi_K).$$

Basis Example

. . . Demonstration . . .

(See `basis-example.Rmd`)

Probit Regression Application

- The receiver operating characteristic (ROC) curve is often used to evaluate classifiers when there are two classes (**?**).
- There are four possible outcomes.

Classify	True Class = 1	True Class = 0
1	**T**rue **P**ositive	**F**alse **P**ositive
0	**F**alse **N**egative	**T**rue **N**egative

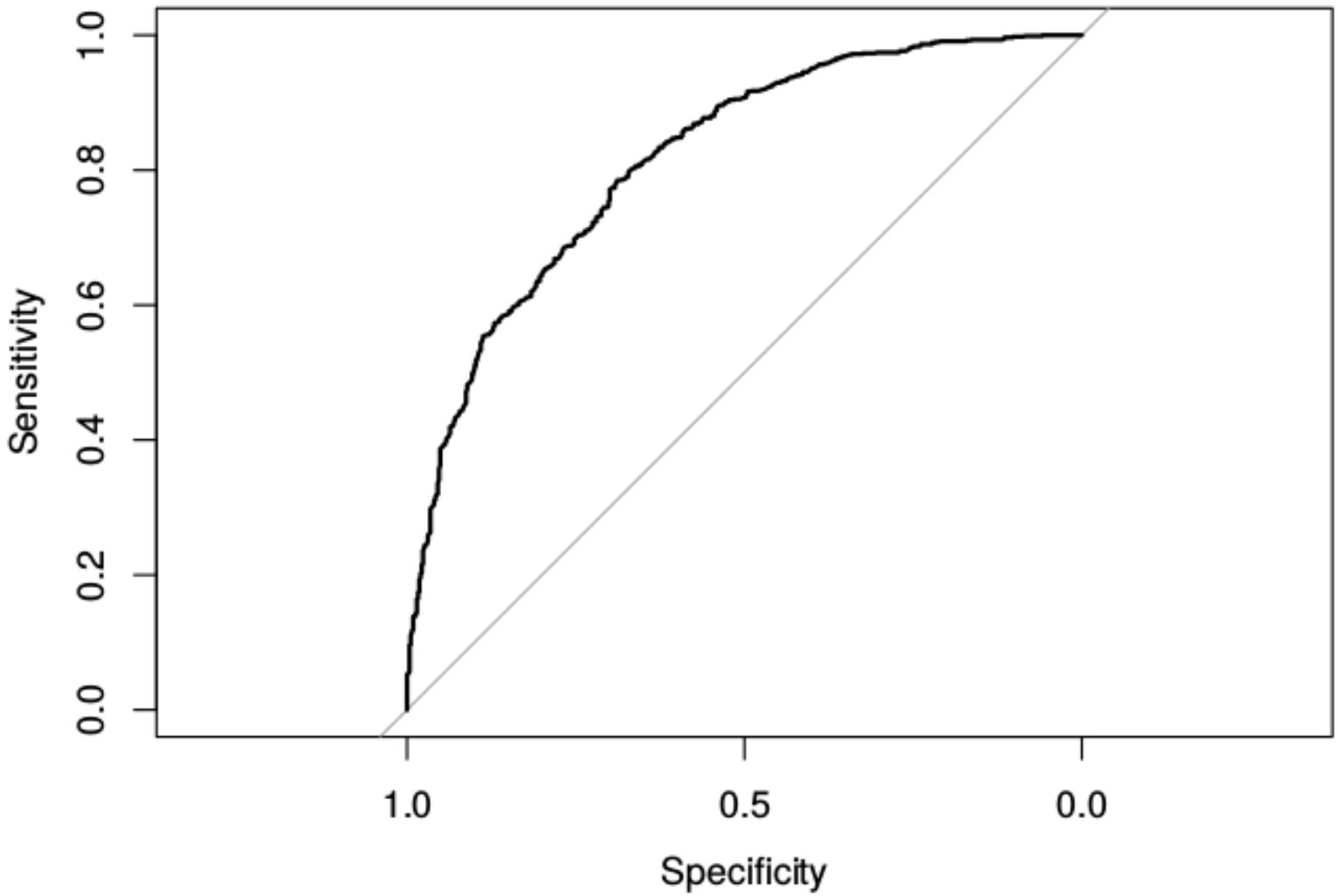
- An ROC plot displays Sensitivity $= \frac{\text{TP}}{\text{TP+FN}}$ vs. Specificity $= \frac{\text{FP}}{\text{FP+TN}}$.
- Classification methods such as probit regression produce an estimate $\hat{p}(\boldsymbol{x}) = \hat{\text{P}}(\text{Class} = 1 \mid \boldsymbol{x})$. To obtain a predicted class, we often consider

$$\widehat{\text{Class}}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \hat{p}(\boldsymbol{x}) > c, \\ 0 & \text{o.w.} \end{cases}$$

- We can produce an ROC curve by considering all the possible cutoffs

$$\Big\{(\text{Sensitivity}(c), \text{Specificity}(c)) : c \in [0, 1]\Big\}.$$

Probit Regression Application



AUC is the Area Under the Curve from $x = 1$ to $x = 0$.

Metropolis-Hastings Sampler

Metropolis-Hastings Sampler

- Suppose $f(\boldsymbol{x})$ is our target density, from which we want to draw. Let $q(\boldsymbol{x}^* \mid \boldsymbol{x})$ be a proposal density which should be easy to draw from.
- Starting with a given $\boldsymbol{X}^{(0)}$
  1. Draw $\boldsymbol{X}^* \sim q(\boldsymbol{X}^* \mid \boldsymbol{X}^{(0)})$
  2. Draw $U \sim \text{Uniform}(0, 1)$ and let

$$\boldsymbol{X}^{(1)} = \begin{cases} \boldsymbol{X}^* & \text{if } U < \min\left\{1, \frac{f(\boldsymbol{X}^*)}{f(\boldsymbol{X}^{(0)})} \frac{q(\boldsymbol{X}^{(0)} \mid \boldsymbol{X}^*)}{q(\boldsymbol{X}^* \mid \boldsymbol{X}^{(0)})}\right\}, \\ \boldsymbol{X}^{(0)} & \text{o.w.} \end{cases}$$

  Repeat starting from $\boldsymbol{X}^{(1)}$, then $\boldsymbol{X}^{(2)}$, etc to generate a chain which eventually behaves like a draw from $f(\boldsymbol{x})$.
- A common choice of $q(\boldsymbol{x}^* \mid \boldsymbol{x})$ is $\text{N}(\boldsymbol{x}^* \mid \boldsymbol{x}, \boldsymbol{V})$. The variance $\boldsymbol{V}$ controls how candidate points are drawn and is a tuning parameter. This choice simplifies step 2, since $q(\boldsymbol{x}^* \mid \boldsymbol{x}) = q(\boldsymbol{x} \mid \boldsymbol{x}^*)$.
- For Bayesians: the normalizing constant of f is not needed, since

$$\frac{f(\boldsymbol{\theta}^* \mid \text{Data})}{f(\boldsymbol{\theta} \mid \text{Data})} = \frac{f(\text{Data} \mid \boldsymbol{\theta}^*) f(\boldsymbol{\theta}^*)}{f(\text{Data} \mid \boldsymbol{\theta}) f(\boldsymbol{\theta})}, \quad \text{which is easy to compute.}$$

Metropolis-Hastings Example

. . . Demonstration . . .

(See `sim.Rmd` and `mathgym.Rmd`)

Gibbs Sampler

Gibbs Sampler

- Again suppose that $f(\boldsymbol{x})$ is our target density, and we wish to draw $\boldsymbol{X} = (X_1, \ldots, X_k)$ from f.
- Starting with a given $\boldsymbol{X}^{(0)}$, obtain $\boldsymbol{X}^{(1)}$ by the following.
  1. Draw $X_1^{(1)} \sim f(X_1 \mid X_2^{(0)}, X_2^{(0)}, \ldots, X_k^{(0)})$.
  2. Draw $X_2^{(1)} \sim f(X_2 \mid X_1^{(1)}, X_3^{(0)}, \ldots, X_k^{(0)})$.

  $\vdots$

  k. Draw $X_k^{(1)} \sim f(X_1 \mid X_1^{(1)}, X_2^{(1)}, \ldots, X_{k-1}^{(1)})$.
- Repeat starting from $\boldsymbol{X}^{(r)}$ to obtain $\boldsymbol{X}^{(r+1)}$ for $r = 1, 2, \ldots$.
- Eventually the chain starts to behave like a draw from $f(\boldsymbol{x})$.
- More generally, the Gibbs sampler can be applied when $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$ is decomposed into vectors of size ≥ 1.
- For many problems, we can readily sample from the k conditional distributions.

Gibbs Sampler

- For probit regression, **?** suggest an augmented data model which leads to a convenient Gibbs sampler,

$$
\begin{aligned}
Y_i &= I(Z_i > 0), \quad i = 1, \ldots, n \\
Z_i &\sim \mathrm{N}(\boldsymbol{x}_i^\top \beta, 1) \\
\beta &\sim \mathrm{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta).
\end{aligned}
$$

- The joint density of all random variables is

$$
f(\boldsymbol{y}, \boldsymbol{z}, \beta) = \prod_{i=1}^{n} \left\{ \left[I(z_i > 0) \right]^{y_i} \left[I(z_i \leq 0) \right]^{1-y_i} \mathrm{N}(z_i \mid \boldsymbol{x}_i^\top \beta, 1) \right\} \times \mathrm{N}(\beta \mid \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta).
$$

- Integrating $\boldsymbol{z}$ out of the augmented model yields the original model.

Gibbs Sampler

- To derive the step for drawing $\boldsymbol{z}$, we have

$$f(\boldsymbol{z} \mid \beta, \boldsymbol{y}) \propto \prod_{i=1}^{n} \left\{ \left[I(z_i > 0) \right]^{y_i} \left[I(z_i \leq 0) \right]^{1-y_i} \mathrm{N}(z_i \mid \boldsymbol{x}_i^\top \beta, 1) \right\} \times$$
$$\mathrm{N}(\beta \mid \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$
$$\propto \prod_{i=1}^{n} \left\{ \left[I(z_i > 0) \right]^{y_i} \left[I(z_i \leq 0) \right]^{1-y_i} \mathrm{N}(z_i \mid \boldsymbol{x}_i^\top \beta, 1) \right\}$$

- Therefore,

$$[\boldsymbol{z} \mid \beta, \boldsymbol{y}] = [z_1 \mid \beta, \boldsymbol{y}] \cdots [z_n \mid \beta, \boldsymbol{y}]$$

where

$$[z_i \mid \beta, \boldsymbol{y}] \sim \begin{cases} I(z_i > 0) \cdot \mathrm{N}(z_i \mid \boldsymbol{x}_i^\top \beta, 1), & \text{if } y_i = 1, \\ I(z_i \leq 0) \cdot \mathrm{N}(z_i \mid \boldsymbol{x}_i^\top \beta, 1), & \text{if } y_i = 0. \end{cases}$$

Gibbs Sampler

- To derive the step for drawing β, we have

$$f(\beta \mid \boldsymbol{z}, \boldsymbol{y}) \propto \prod_{i=1}^{n} \left\{ \left[I(z_i > 0) \right]^{y_i} \left[I(z_i \leq 0) \right]^{1-y_i} \mathrm{N}(z_i \mid \boldsymbol{x}_i^\top \beta, 1) \right\} \times \mathrm{N}(\beta \mid \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$

$$\propto \prod_{i=1}^{n} \left\{ \mathrm{N}(z_i \mid \boldsymbol{x}_i^\top \beta, 1) \right\} \mathrm{N}(\beta \mid \boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$$

- After some calculation, it can be shown that

$$[\beta \mid \boldsymbol{z}, \boldsymbol{y}] = [\beta \mid \boldsymbol{z}] \sim \mathrm{N}(\tilde{\boldsymbol{\mu}}_\beta, \tilde{\boldsymbol{\Sigma}}_\beta),$$

where $\tilde{\boldsymbol{\Sigma}}_\beta = (\boldsymbol{\Sigma}_\beta^{-1} + \boldsymbol{X}^\top \boldsymbol{X})^{-1}$ and $\tilde{\boldsymbol{\mu}}_\beta = \tilde{\boldsymbol{\Sigma}}_\beta (\boldsymbol{\Sigma}_\beta^{-1} \boldsymbol{\mu}_\beta + \boldsymbol{X}^\top \boldsymbol{z})$.

Gibbs Example

. . . Demonstration . . .

(See `sim.Rmd` and `mathgym.Rmd`)

References I