REVIEW OF MATH 459

刘谦

Table of Contents

1 Introduction to posterior distribution

1.1 One parameter model

Recall: $X^{(n)} = \{X_1, \ldots, X_n\}$, $X_i \overset{iid}{\sim} f(x|\theta)$

Likelihood: $L(\theta) \equiv L(\theta; X^{(n)}) = \prod_{i=1}^{n} f(x_i|\theta)$

(log likelihood: $\log L(\theta) \equiv \ell(\theta)$

Posterior: $P(\theta|X^{(n)}) = \dfrac{p(\theta)\cdot f(x^{(n)}|\theta)}{\int_\theta p(\tilde{\theta}) f(x^{(n)}|\tilde{\theta}) d\tilde{\theta}}$

Question: If the $p(\theta)$ is a density, then is $p(\theta)f(x^{(n)}|\theta)$ also a density?

Where does this ~~from~~? posterior ~~go~~ come from?

- Imagine θ is a random variable
  obviously X is a random variable
- Can not assume θ, X are indep
- Let $g(\theta, X)$ be the joint density
  $f(X|\theta) = \dfrac{g(\theta, X)}{p(\theta)}$, $p(\theta) > 0$

the postoriot is $P(\theta|x) = \dfrac{g(\theta, x)}{m(x)}$, $m(x) > 0$

- in general, ~~not~~ $m(x)$ unknown.
- Can find $m(x) = \int_\theta g(\theta, x) d\theta$
- and we know
  $g(\theta, X) = f(x|\theta) \cdot p(\theta)$
- Thus $p(\theta|x) = \dfrac{g(\theta, X)}{m(x)} = \dfrac{p(\theta)\cdot f(x|\theta)}{\int_\theta p(\tilde{\theta}) f(x|\tilde{\theta}) d\tilde{\theta}}$

$$\int_\theta P(X^{(m)} \mid \tilde{\theta}) \, P(\tilde{\theta}) \, d\tilde{\theta}$$

① $m(x) = \dfrac{P(\theta) f(x \mid \theta)}{P(\theta \mid x)}$

Basic Marginal Likelihood Identity.

of data

② $m(x)$: normalizing constant C

so that $\int_\theta P(\theta \mid x) \, d\theta = 1$

Interpretation :

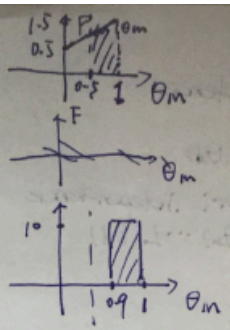
posterior $\propto$ prior $\times$ likelihood

$P_m(\theta_m) = \theta_m + 0.5$, $\theta_m \in (0,1)$

$P_{rm}(\theta_m > 0.5) = \int_{0.5}^{1} P_m(\theta) d_m = 0.625$

$P_N(\theta_s) = 10$, $\theta \in (0.9, 1)$

$P_{rm}(\theta_m > 0.5) = \int_{0.5}^{1} P_m(\theta) d_m = 1$



prior dist

⇓

posterior dist

⇓

posterior confidence interval

$P(\theta_m \mid y_m = 0) \propto P(\theta_m) \cdot P(y_m \mid \theta_m)$

$\propto \dfrac{\theta_m}{2} - \theta_m^2 + \dfrac{1}{2}$

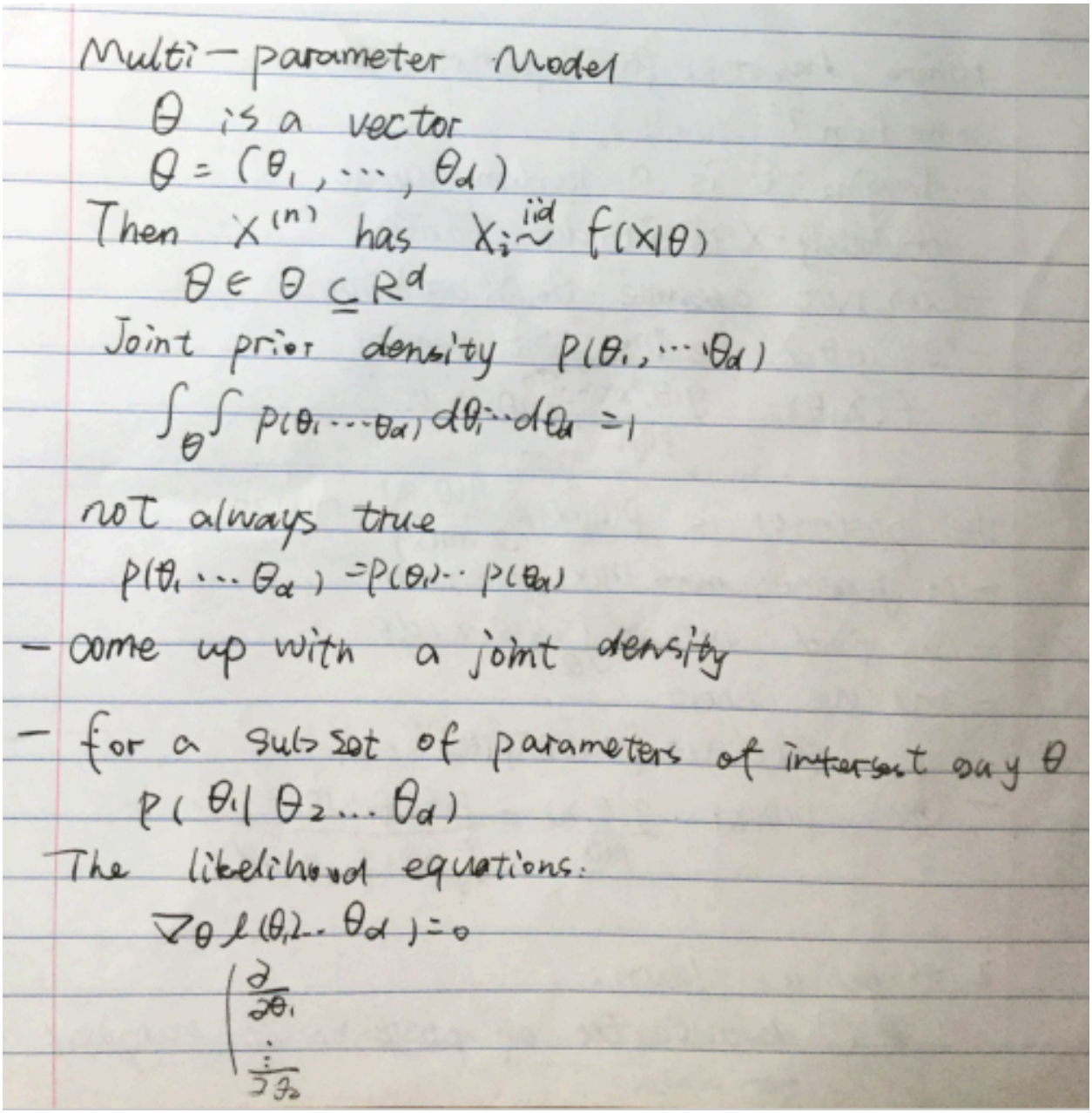
$P_{r\theta_m \mid Y_m}(\theta_m > 0.5) = \int_{0.5}^{1} P_m(\theta_m \mid y_m) d\theta_m = 0.35$

give up after one try

prior allow us to incorporate common sense

Lecture 9

1.2 multi-parameter model



E.X.

2. Consider the Gaussian model from Lecture 3. Assume $X_1, \ldots, X_n$ are i.i.d. $\mathcal{N}(\mu, \sigma^2)$ where both μ and σ^2 are unknown. Find the marginal posterior for μ, assuming the prior

$$p(\mu, \sigma^2) = p(\mu|\sigma^2)p(\sigma^2)$$

where $p(\sigma^{-2}) = e^{-\sigma^{-2}}, \quad \sigma^{-2} > 0$. You will need to use a change of variables to find $p(\sigma^2)$. Also, assume that $p(\mu|\sigma^2) = (2\pi\sigma^2)^{-1/2} \exp\{-\mu^2/(2\sigma^2)\}$. You may ignore constants of proportionality. Can you identify which type of distribution this marginal posterior density represents?

Solution.

Since $p(\sigma^{-2}) = e^{-\sigma^{-2}}$, $\sigma^{-2} > 0$, we know $p(\sigma) = e^{-\sigma}$, $\sigma > 0$. By change-of-variables argument, we know $p(\sigma^2) = \sigma^{-4} e^{-\sigma^{-2}}$, $\sigma > 0$. Then, the prior

$$\begin{aligned} p(\mu, \sigma^2) &= p(\mu|\sigma^2)p(\sigma^2) \\ &= (2\pi\sigma^2)^{-1/2} \exp\{-\mu^2/(2\sigma^2)\} \frac{1}{\sigma^4} \exp\{-\frac{1}{\sigma^2}\} \\ &\propto \sigma^{-5} \exp\left\{ -\frac{\mu^2 + 2}{2\sigma^2} \right\} \end{aligned}$$

We start with the most basic likelihood function for assumed normally distributed data, and rearrange it slightly:

$$\begin{aligned} L(\mu, \sigma^2|\mathbf{x}) &= (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 \right\} \\ &= (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} [(x_i - \bar{x}) - (\mu - \bar{x})]^2 \right\} \\ &= (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \left(\sum_{i=1}^{n} (x_i - \bar{x})^2 - \underbrace{2 \sum_{i=1}^{n} (x_i\mu - x_i\bar{x} - \bar{x}\mu + \bar{x}^2)}_{=0} + n(\mu - \bar{x})^2 \right) \right\} \\ &\propto \sigma^{-n} \exp\left\{ -\frac{1}{2\sigma^2} \Big((n-1)s^2 + n(\mu - \bar{x})^2 \Big) \right\} \end{aligned}$$

where

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

is the sample variance. The resulting joint posterior distribution for (μ, σ^2) is then

$$\begin{aligned}\pi(\mu, \sigma^2|\mathbf{x}) &\propto L(\mu, \sigma^2|\mathbf{x})p(\mu, \sigma^2) \\ &\propto \sigma^{-n}\exp\left\{-\frac{1}{2\sigma^2}\Big((n-1)s^2 + n(\mu-\bar{x})^2\Big)\right\}\sigma^{-5}\exp\left\{-\frac{\mu^2+2}{2\sigma^2}\right\} \\ &= (\sigma^2)^{-\frac{n+5}{2}}\exp\left\{-\frac{1}{2\sigma^2}\Big((n-1)s^2 + n(\mu-\bar{x})^2 - \mu^2 - 2\Big)\right\}\end{aligned}$$

The marginal posterior distribution for μ can be obtained by integrating σ^2 out of the joint posteri distribution:

$$\begin{aligned}\pi(\mu|\mathbf{x}) &= \int_0^\infty \pi(\mu, \sigma^2|\mathbf{x})d\sigma^2 \\ &\propto \int_0^\infty (\sigma^2)^{-\frac{n+5}{2}}\exp\left\{-\frac{1}{2\sigma^2}\Big((n-1)s^2 + n(\mu-\bar{x})^2 - \mu^2 - 2\Big)\right\}d\sigma^2\end{aligned}$$

This integral can be evaluated using the substitution

$$z = \frac{A}{2\sigma^2},$$

$$A = (n-1)s^2 + n(\mu-\bar{x})^2 - \mu^2 - 2.$$

Then,

$$\pi(\mu|\mathbf{x}) \propto A^{-\frac{n+3}{2}}\int_0^\infty z^{\frac{n+1}{2}}e^{-z}dz$$

This integrand is the inverse gamma density and thus the integral is a constant. Therefore,

$$\pi(\mu|\mathbf{x}) \propto A^{-\frac{n+3}{2}} = [(n-1)s^2 + n(\mu-\bar{x})^2 - \mu^2 - 2]^{-\frac{n+3}{2}}$$

which is a scaled non-central t distribution.

2 Decision rule

2.1 What is decision rule

The rule of choosing an estimator.

Math 1061

What is a good ~~dis~~ decision rule?

Ideal case: $\exists\ d \in D$ s.t.

$R(\theta, d)$ is uniformly smallest $\forall\ \theta \in \Theta$

Not practical.

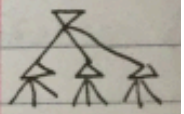
Admissibility: Given two decision rules d, d' d strictly dominates d' if $R(\theta, d) \leq R(\theta, d')$ $\forall\ \theta \in \Theta$ and is strictly less for at least one θ

Any rule which is strictly is strictly dominated is inadmissible.

If d is not strictly ~~rule~~ dom. it is admissible

Minimaxity: The max risk of d is

$$MR(d) = \sup_{\theta \in \Theta} R(\theta, d)$$



d is minimax if $MR(d) \leq MR(d')\ \forall d' \in D$

Equivalently, d must satisfy $\sup_\theta R(\theta, d) = \inf_{d' \in \Theta} \sup R(\theta, d')$

If The supremum and infimum are ~~actually~~ attained.

$$\max_{\theta \in \Theta} R(\theta, d) = \min_{d' \in D} \max_{\theta \in \Theta} R(\theta, d')$$

Unbiasedness

d is unbiased if $E_\theta L(\theta', d(X)) \geq E_\theta L(\theta, d(X))$

θ: true, θ': different

d is able to ~~find out~~ the optimal θ from other θ'

for $\forall\ \theta, \theta' \in \Theta$

Recall, in estimator:

$d(X)$ is unbiased for θ, if $E_\theta d(X) = \theta$

If $L(\theta, d) = (\theta - d)^2$ these definition are the same

Statistical Decision Problem

① Parameter space

② $\Theta \subseteq \mathbb{R}^d$

Set of possible true states of nature

③ Sample space $\mathbb{X}$

where data live

typically n observations

generic element $x \in \mathbb{X}$ is $x = (x_1, \cdots, x_n) \in \mathbb{R}^n$

④ family of prob. dist on $\mathbb{X}$

indexed by values $\theta \in \Theta$

$\{P_\theta(x) : x \in \mathbb{X}, \theta \in \Theta\}$

⑤ Action Space A

Set of all actions available to the experimenter

(a) Hypothesis testing

decide b/w H_0, H_1

$A = \{a_0, a_1\}$ → reject H_0

Fail to reject

(b) Estimation

estimate θ by a func of $X^{(n)}$, e.g

$\bar{X} = n^{-1} \sum_{i=1}^{n} X_i$, or $X_1^3 + X_2 \cos\{\sqrt{X_3}\}$

⑥ Loss function

$L : \Theta \times A \mapsto \mathbb{R}$

Links action to unknow Parameter. If we take action a when true state of nature is $\theta \in \Theta$, incur loss $L(\theta, a)$

⑦ Set of ~~desi~~ decision rule D

An element $d \in D$, $d : \mathbb{X} \mapsto A$

s.t each $x \in \mathbb{X}$ is assoc. with $d(x) \in A$

Risk function

For $\theta \in \Theta$ the risk of decision d based on random X

$R(\theta, d) = E_\theta[L(\theta, d(X))]$ ← ~~given~~ θ average over data. [given θ]

$$= \begin{cases} \int_x L[\theta, d(x)]\, f(x,\theta)\, dx \\ \sum_x L(\theta, d(x))\, f(x;\theta), \end{cases}$$

→ frequentist risk over θ (fp)

Bayesian on (RL) $\underset{H(p,d)}{\Rightarrow}$ $E\{\underbrace{E_\theta[L\theta, d(x)]}_{\text{over } x}\}$

key notion

Different decision rules should be composed in terms of their risk as function θ

$\int dx\, d\theta$

Common loss Functions for ~~estimation~~.

(a) $L(\theta, a) = (\theta - a)^2$

(b) $L(\theta, a) = |\theta - a|$ absolute ~~error~~

(c) $L(\theta, a) = \begin{cases} 0 & \text{if } |\theta - a| \le \delta \\ 1 & \text{if } |\theta - a| > \delta \end{cases}$

mean square error

Common loss func in Testing.

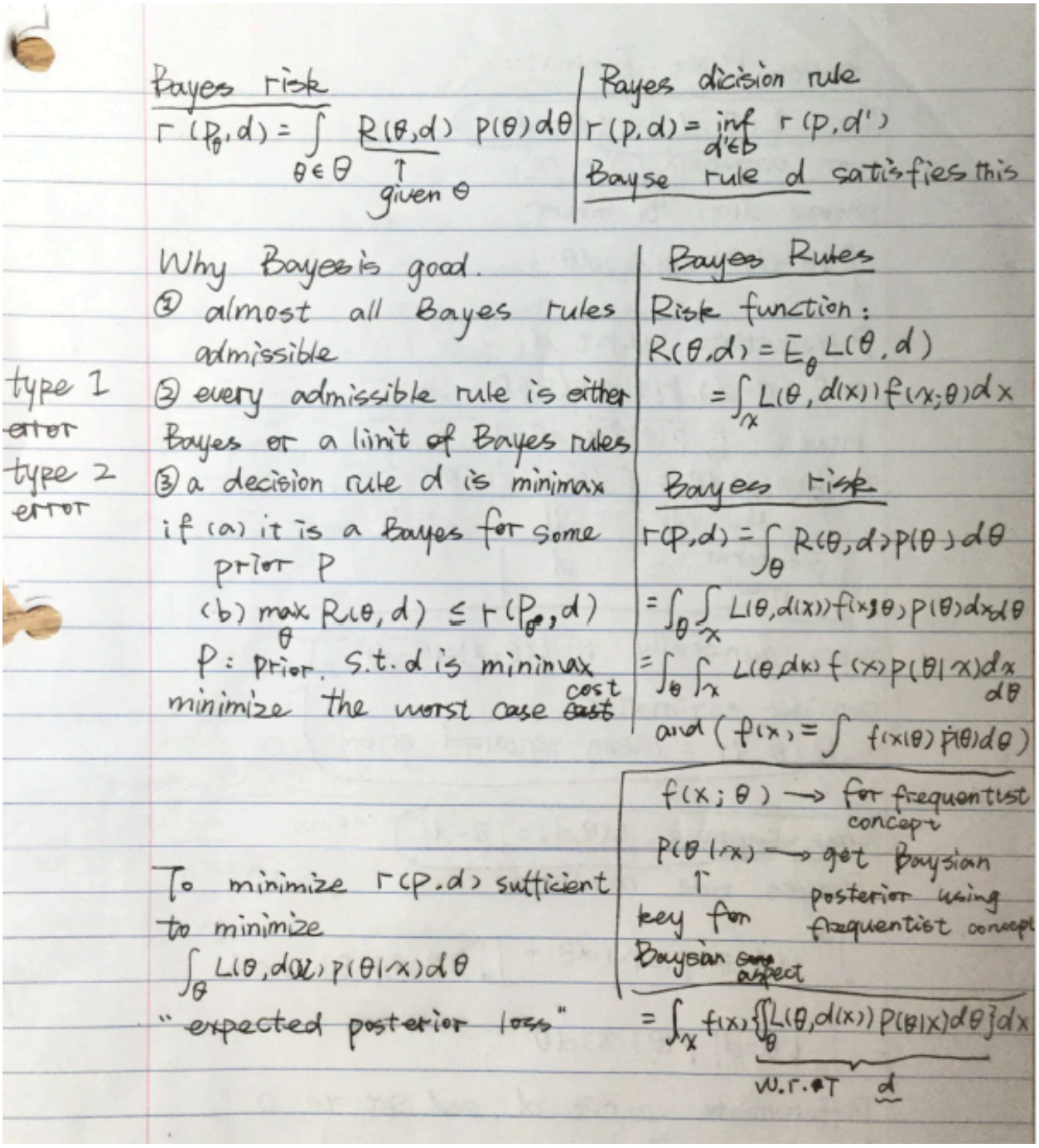
$$L(\theta, a) = \begin{cases} 1 & \text{if } \theta \in H_0 \text{ and } a = a_1 \\ 1 & \text{if } \theta \in H_1 \text{ and } a = a_0 \\ 0 & \text{Otherwise} \end{cases}$$

Risk associated with these losses.

Type I and Type II error

Here $R(\theta, d) = \begin{cases} P_{r_\theta}\{d(X) = a_1\} & \text{if } \theta \in H_0 \\ P_{r_\theta}\{d(X) = a_0\} & \text{if } \theta \in H \end{cases}$

2.2 Bayes Decision rule



Bayes risk

$r(P_\theta, d) = \int_{\theta \in \Theta} R(\theta, d)\, p(\theta)\, d\theta$

$R(\theta, d)$: given θ

Bayes decision rule

$r(p, d) = \inf_{d' \in D} r(p, d')$

Bayse rule d satisfies this

Why Bayes is good.

① almost all Bayes rules admissible

② every admissible rule is either Bayes or a limit of Bayes rules

③ a decision rule d is minimax if (a) it is a Bayes for some prior P

(b) $\max_\theta R(\theta, d) \le r(P_\theta, d)$

P: prior, s.t. d is minimax

minimize the worst case cost

type 1 error

type 2 error

To minimize $r(p, d)$ sufficient to minimize

$\int_\theta L(\theta, d(x))\, p(\theta|x)\, d\theta$

"expected posterior loss"

Bayes Rules

Risk function:

$R(\theta, d) = E_\theta L(\theta, d)$

$= \int_x L(\theta, d(x))\, f(x;\theta)\, dx$

Bayes risk

$r(p, d) = \int_\theta R(\theta, d)\, p(\theta)\, d\theta$

$= \int_\theta \int_x L(\theta, d(x))\, f(x|\theta)\, p(\theta)\, dx\, d\theta$

$= \int_\theta \int_x L(\theta, d(x))\, f(x)\, p(\theta|x)\, dx\, d\theta$

and $(f(x) = \int f(x|\theta)\, p(\theta)\, d\theta)$

$f(x;\theta)$ → for frequentist concept

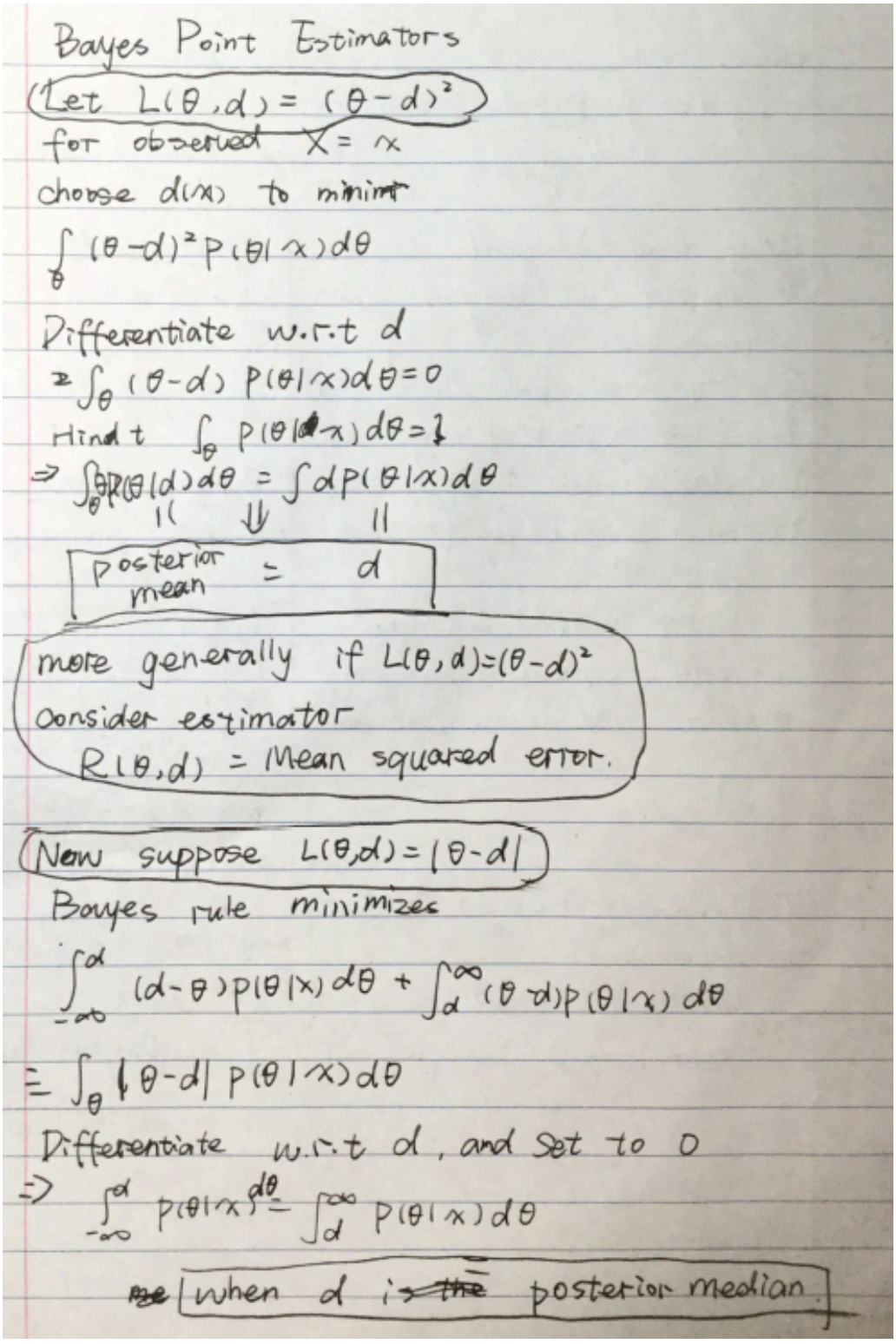
$p(\theta|x)$ → get Bayesian posterior using frequentist concept

↑ key for Bayesian concept

$= \int_x f(x) \{\int_\theta L(\theta, d(x))\, p(\theta|x)\, d\theta\}\, dx$

w.r.t d

2.2.1 Bayesian point estimators



Bayes Point Estimators

Let $L(\theta, d) = (\theta - d)^2$

for observed $X = x$

choose $d(x)$ to minimize

$$\int_\theta (\theta - d)^2 p(\theta | x) d\theta$$

Differentiate w.r.t d

$$= \int_\theta (\theta - d)\, p(\theta | x) d\theta = 0$$

Hint $\int_\theta p(\theta | x) d\theta = 1$

$$\Rightarrow \int_\theta \theta\, p(\theta | x) d\theta = \int d\, p(\theta | x) d\theta$$

$$\Downarrow \qquad\qquad \Downarrow$$

posterior mean $=$ d

more generally if $L(\theta, d) = (\theta - d)^2$

consider estimator

$R(\theta, d)$ = Mean squared error.

Now suppose $L(\theta, d) = |\theta - d|$

Bayes rule minimizes

$$\int_{-\infty}^{d} (d - \theta) p(\theta | x) d\theta + \int_{d}^{\infty} (\theta - d) p(\theta | x) d\theta$$

$$= \int_\theta |\theta - d|\, p(\theta | x) d\theta$$

Differentiate w.r.t d, and set to 0

$$\Rightarrow \int_{-\infty}^{d} p(\theta | x) d\theta = \int_{d}^{\infty} p(\theta | x) d\theta$$

when d is posterior median.

$$\int_{-\infty}^{\infty} L(v,\delta) f_{\theta|x}(v|x)\,dv = \int_{-\infty}^{\delta-\frac{b}{2}} 1 \cdot f_{\theta|x}(v|x)\,dv + \int_{\delta+\frac{b}{2}}^{} 1 \cdot f_{\theta|x}(v|x)\,dx$$

Bayes interval estimator

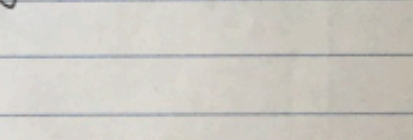
Suppose $L(\theta,d) = \begin{cases} 0 & \text{if } |\theta - d| \le \delta \\ 1 & \text{if } |\theta - d| \ge \delta \end{cases}$

δ is a tolerance level.

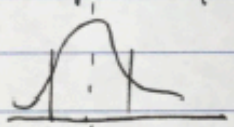
Problem: Find the 'best' interval with ~~respect~~ prespecified length 2δ

$(d-\delta, d+\delta) \Rightarrow$ Lost $= 0$

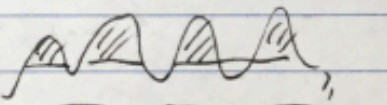
'Best' means the interval maximizes the posterior probability that $\theta \in (d-\delta, d+\delta)$



HPDI (Highest posterior desity interval)

If 

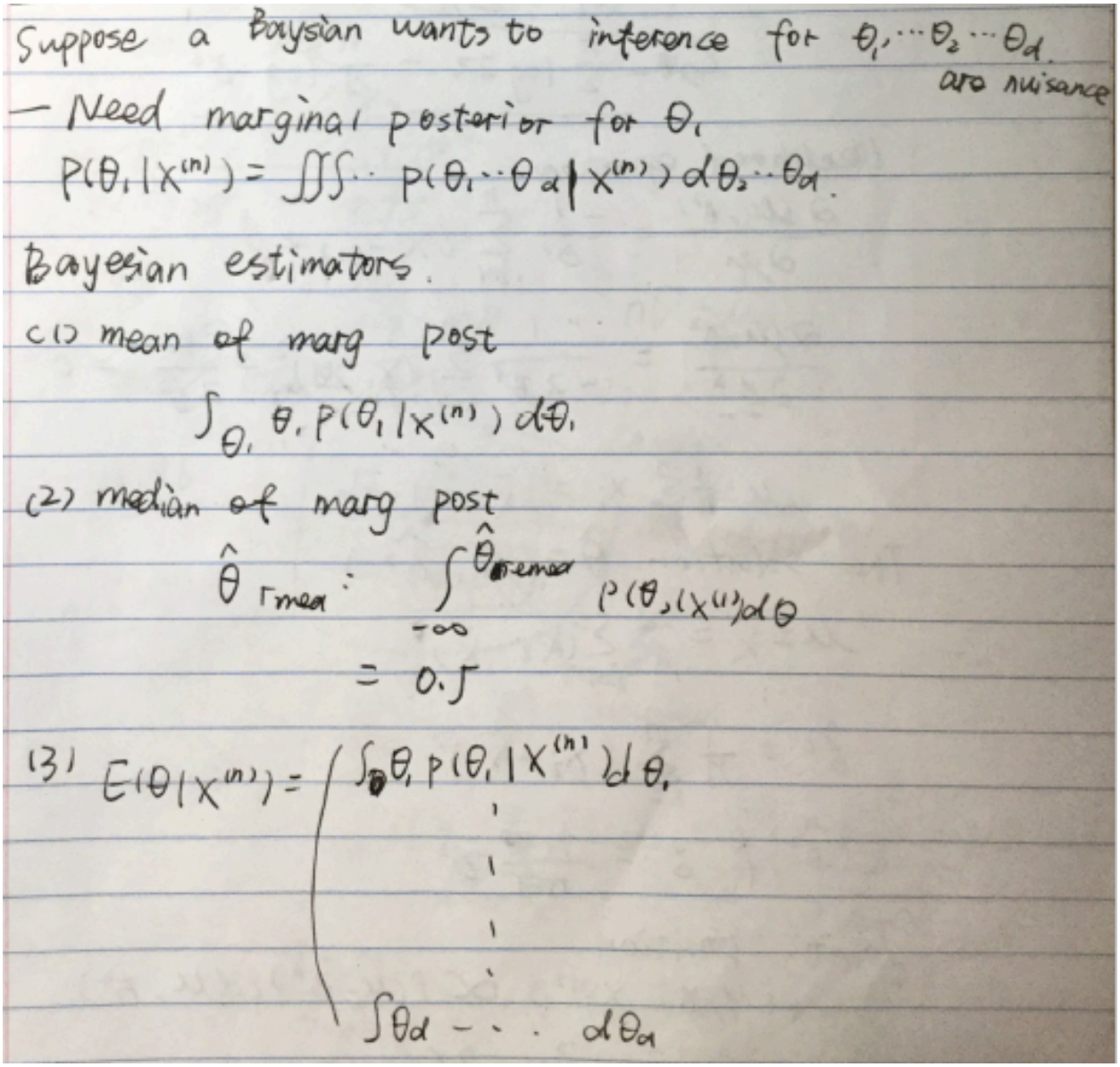
easy to construct

If tolerance ε 

HPD set union of disjoint interval

simplify problem.

2.2.2 properties of posterior estimators



2.2.3 Basics of Parametric Bayesian Asymptotics

consistency convergence to point mass

asymptotic normality Bernstein-von Mises theorems

agreement with frequentist intervals first-order likelihood and Bayesian asymptotics agree

2.2.3.1 consistency

Let $X_1, \ldots, X_n \overset{i.i.d.}{\sim} f(x|\theta_0)$. Denote the posterior by $\Pi(\cdot|X^{(n)})$.

Definition

The sequence of posteriors $\Pi(\cdot|X^{(n)})$ is **consistent** at a point $\theta_0 \in \Theta$ if for every neighborhood U of θ_0, we have that $\Pi(U|X^{(n)}) \to 1$ as $n \to \infty$ almost surely (with respect to the distribution under θ_0).

This implies that the usual estimators such as the posterior mean are consistent in the usual sense.

2.2.3.2 Normality

(Corollary of) BvM theorem

Let $X_1, \ldots, X_n \overset{i.i.d.}{\sim} f(x|\theta)$ and let $\theta \sim \pi(\theta)$, a density function.

Assumptions regularity conditions

- Θ an open set, likelihood function sufficiently smooth
- $0 < I(\theta) < \infty$ (Fisher information)
- prior $\pi(\theta)$ sufficiently smooth in neighborhood of true value θ_0

Theorem Let $\hat{\theta}_n$ be a strongly consistent sequence of roots of the likelihood equation, and let $\Omega(x)$ be the CDF of a normal random variable with mean 0 and variance $I^{-1}(\theta_0)$. Then

$$\sup_{-\infty<x<\infty} |P(\sqrt{n}(\theta - \hat{\theta}_n) \leq x|X^{(n)}) - \Omega(x)| \to_{a.s.} 0$$

2.2.3.3 Agreement of Bayesian and Frequentist Inference

One may guess that since both the posterior and MLE are asymptotically normal, then posterior credible sets and likelihood-based confidence sets might agree asymptotically.

- formally this means that all smooth priors are probability matching to order $O(n^{-1/2})$
- that is, the frequentist repeated sampling probability coverage of a $100(1-\alpha)$-credible set is $1 - \alpha + O(n^{-1/2})$
- philosophically, would a Bayesian care about this?

3 How to choose a prior?

3.1 Background Concept 1: Prior Independence

Consider $\theta = (\theta_1, \ldots, \theta_d)^T$.

- If parameters *a priori* independent, then

$$p(\theta) = p(\theta_1)p(\theta_2)\cdots p(\theta_d).$$

Does not imply *a posteriori* independence.

3.2 Background Concept 2: Fisher Information

Consider $\theta = (\theta_1, \ldots, \theta_d)$. The Fisher information of f, or equivalently, X, is

$$I(\theta) = E_\theta \left[\nabla_\theta \log f(X;\theta) \nabla_\theta \log f(X;\theta)^T \right]$$

which is the covariance of the score function, $\nabla_\theta \log f(X;\theta)$.

The (i,j)th element of the Fisher information matrix is given by

$$I_{ij} = E_\theta \left[\left(\frac{\partial}{\partial \theta_i} \log f(X;\theta) \right) \left(\frac{\partial}{\partial \theta_j} \log f(X;\theta) \right) \right]$$

$$\text{(under regularity)} = -E_\theta \left[\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log f(X;\theta) \right]$$

- Regularity conditions needed to ensure validity of interchanging differentiation and integration.
- Sufficient condition: Lebesgue's dominated convergence theorem, i.e. there exists a function g such that $g(x) \geq \|\nabla_\theta f(x;\theta)\|$ for all θ

3.3 Background Concept 3: Orthogonality

θ_i and θ_j are **orthogonal** if $I_{ij} = 0$.

However, prior independence plus orthogonality **does not imply** posterior independence!

For prior independence to imply posterior independence, also need the likelihood to factor:

$$L(\theta) = L(\theta_1)L(\theta_2)\cdots L(\theta_d)$$

3.4 Selecting Priors by Formal Rules

Definition

A formal rule for selecting a prior is a prescription for how to specify a prior, which is derived from some chosen principle, and this principle is broadly applicable to a wide variety of settings (i.e. the rule is not specific to the particular sampling model).

Let Ψ be a rule for choosing a prior $p(\theta)$, i.e. Ψ could be something like

- principle of insufficient reason
- maximum entropy
- match posterior intervals and frequentist intervals
- invariance under transformations
- maximize the 'information' provided by the data
- decision theory: 'least favorable priors', unbiased decision rules

3.5 Noninformative Priors & Objective Bayes

Prior introduces information into the model.

- Objective Bayesians want priors to have little *influence* on the posterior
- not as easy as it seems!

Historical Approach (Bayes, Laplace): use flat priors

- expresses ignorance
- all values equiprobable

- note: by contrast, a *subjective* Bayesian would first choose a prior, then apply it to the likelihood to derive the posterior

allows data to have maximum effect on the posterior

problem: not invariant under one-to-one reparameterizations!

Before reparameterization, the prior is noninformative, but it is informative in new parameterization.

Problem: Not *invariant*

Consider a prior for a variance parameter σ^2, and the reparameterization $\eta = \log \sigma^2$.

uniform for η $p(\eta) \propto 1$ implies

$$p(\sigma^2) \propto \sigma^{-2}$$

uniform for σ^2 $p(\sigma^2) \propto 1$ implies

$$p(\eta) \propto \exp(\eta)$$

Consider reparameterization using log-odds ratio:

$$\eta = \log \frac{\theta}{1-\theta}$$

- valid reparameterization–natural for mapping θ to the real line
- but the prior $p(\eta) = 1$ is not flat

In original parameterization, $p(\cdot)$ is noninformative, but it is informative in new parameterization.

$\Rightarrow$ not invariant under one-to-one reparameterizations!

3.5.1 Harold Jeffreys rule: formalize what is meant by 'uninformative' prior

Idea: use principle of invariance w.r.t. one-to-one transformations, redefine invariance.

Define a new parameter $\eta = h(\theta)$, $h(\cdot)$ one-to-one. For simplicity, assume θ, η are scalar.

- if we calculate $\pi_J(\theta)$ w.r.t. θ, then transform variables, we will get a prior π on η by change-of-variables formula
- if this prior $\pi(\eta)$ is the same as $\pi_J(\eta)$, that would be computed using η from the beginning, then the Jeffreys rule is invariant under one-to-one transformations

Apply Jeffreys's rule to η and use chain rule to re-express in terms of θ:

$$\begin{aligned} I(\eta) &= -E\left[\frac{d^2 \log f(X;\eta)}{d\eta^2}\right] \\ &= -E\left[\frac{d^2 \log f(X;\theta)}{d\theta^2}\left(\frac{d\theta}{d\eta}\right)^2 + \frac{d \log f(X;\theta)}{d\theta}\frac{d^2\theta}{d\eta^2}\right] \\ &= -E\left[\frac{d^2 \log f(X;\theta)}{d\theta^2}\right]\left(\frac{d\theta}{d\eta}\right)^2 + E\left[\frac{d \log f(X;\theta)}{d\theta}\right]\frac{d^2\theta}{d\eta^2}. \end{aligned}$$

We have exactly what we want, provided that

$$E\left[\frac{d \log f(X;\theta)}{d\theta}\right] = 0$$

For all θ, $\int f(X;\theta)dX = 1$. Assume sufficiently regularity, differentiate w.r.t. θ

$$\begin{aligned} 0 &= \frac{d}{d\theta}\int f(X;\theta)dX \\ &= \int \frac{df(X;\theta)}{d\theta}\frac{f(X;\theta)}{f(X;\theta)}dX \\ &= \int \left[\frac{df(X;\theta)}{d\theta}\frac{1}{f(X;\theta)}\right] f(X;\theta)dX \\ &= \int \left[\frac{d \log f(X;\theta)}{d\theta}\right] f(X;\theta)dX \\ &= E\left[\frac{d \log f(X;\theta)}{d\theta}\right] \end{aligned}$$

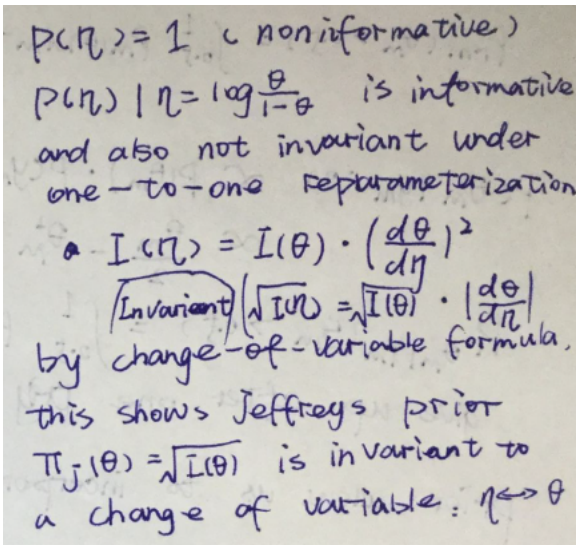
Taking expectation over X with fixed θ is equivalent to taking expectation with η fixed $\Rightarrow$

$$I(\eta) = I(\theta)\left(\frac{d\theta}{d\eta}\right)^2$$

take square root

$$\sqrt{I(\eta)} = \sqrt{I(\theta)}\left|\frac{d\theta}{d\eta}\right|$$

By change-of-variable formula, this shows Jeffreys prior $\pi_J(\theta) = \sqrt{I(\theta)}$ is invariant to a change of variable.



Before apply Jeffreus rule, the prior fine is not invariant. After applying Jeffeys rule, we show that theta is invariant to a change of variable.

E.X. of Jeffeys prior.

1. In this homework you will perform a Bayesian analysis of the gamma distribution using an uninformative prior and MCMC. The density of the gamma distribution is

$$f(x|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}\exp(-\beta x), \quad \alpha,\beta > 0.$$

Derive Jeffreys prior (you must show your work to get credit). Some hints:

(a) To find the Fisher information matrix, you only need the log-likelihood for a sample of size 1.

(b) The derivatives of the natural logarithm of the gamma function are special functions. Note that $\partial \log(\Gamma(\alpha))/\partial\alpha$ is the `digamma` function (also in `R`) and $\partial^2 \log(\Gamma(\alpha))/\partial\alpha^2$ is the `trigamma` function.

Solution.

For Gamma distribution, the density function is

$$f(x|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}\exp(-\beta x), \quad \alpha,\beta > 0.$$

To find the Fisher information matrix, we only need the log-likelihood for a sample of size 1. Then, the likelihood function is

$$L(\alpha,\beta|x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}\exp(-\beta x)$$

and the log-likelihood function is

$$\log L(\alpha,\beta|x) = \alpha\log\beta - \log\Gamma(\alpha) + (\alpha-1)\log x - \beta x$$

First, calculate the first order derivatives for log-likelihood function $\log L(\alpha,\beta|x)$.

$$\frac{\partial \log L}{\partial \alpha} = \log\beta - \frac{\partial \log\Gamma(\alpha)}{\partial\alpha} + \log x$$

$$\frac{\partial \log L}{\partial \beta} = \frac{\alpha}{\beta} - x$$

Then, the second order derivatives are as follows:

$$\frac{\partial^2 \log L}{\partial \alpha^2} = -\frac{\partial^2 \log \Gamma(\alpha)}{\partial \alpha^2}$$
$$\frac{\partial^2 \log L}{\partial \alpha \partial \beta} = \frac{\partial^2 \log L}{\partial \beta \partial \alpha} = \frac{1}{\beta}$$
$$\frac{\partial^2 \log L}{\partial \beta^2} = -\frac{\alpha}{\beta^2}$$

Hence, we obtain the fisher information matrix:

$$I(\alpha, \beta) = \begin{bmatrix} \frac{\partial^2 \log \Gamma(\alpha)}{\partial \alpha^2} & -\frac{1}{\beta} \\ -\frac{1}{\beta} & \frac{\alpha}{\beta^2} \end{bmatrix}$$

Jeffreys' principle leads to defining the non-informative prior density as $\pi_J(\alpha, \beta) \propto \sqrt{\det I(\alpha, \beta)}$.
Therefore, we have Jeffreys prior

$$\pi_J(\alpha, \beta) \propto \frac{1}{\beta} \sqrt{\alpha \frac{\partial^2 \log \Gamma(\alpha)}{\partial \alpha^2} - 1}$$

4 Approximate Bayesian Inference

Exact analytic calculation of posterior quantities often not practical.

4.1 Question: when do we need to approximate an integral?

Ans: if the integral of a function can be computed in closed form.

Question: Is there a simple, fool-proof way to determine if the integral of a function can be computed in closed form?
One approach: let's consider elementary functions.

Definition

A function built using a finite combination of constant functions, algebraic operations (addition, multiplication, division, raising to integer power, root extractions–fractional power), logarithmic, exponential and algebraic functions and their inverses *under repeated compositions* is called an **elementary function**.

Types of elementary functions:

1. algebraic functions (can be expressed as solution of a polynomial equation): polynomials, rational functions, root extraction
2. *(non-algebraic)* transcendental functions: exponentials, logarithms, power functions, periodic functions (e.g. trigonometric: sine, cosine, etc.)

Example

$$\frac{\sin^{-1}(x^4-3)}{\sqrt{\log(6x)+\cos(x^{-2}+9)}}$$

The set of elementary functions is **closed** under *arithmetic operations* (addition, subtraction, multiplication, division) and *differentiation.*

However, it is not closed under integration (**Liouville's theorem**, 1830s)

Implication of Liouville's Theorem

The integrals of certain elementary functions cannot themselves be expressed as elementary functions.

A Non-Elementary Example

The CDF of the standard normal distribution is given by

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt.$$

- you were taught $f(x) = (2\pi)^{-1/2} \exp(-x^2/2)$ is a density, so we must have that $I = \int_{-\infty}^{\infty} f(x)dx = 1$
- but $\int e^{-x^2} dx$ is not an elementary function; there is no closed-form expression
- we *cannot* show that $I = 1$ by computing $\Phi(x)$ as an explicit function of x and then finding $\lim_{x\to\infty} \Phi(x)$

The Gaussian integral $\int e^{-x^2} dx$ is not an elementary function.

Some special functions are the non-elementary antiderivatives of elementary functions (and hence must be approximated).

Example

Exponential integral: $\mathrm{Ei}(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t} dt$

Error function: $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$

Other Non-Elementary Antiderivatives

- $\frac{\sin x}{x}$
- x^x
- $\frac{1}{\log x}$
- $\log(\log x)$
- $\exp(e^x)$
- the integrands of **elliptic integrals**

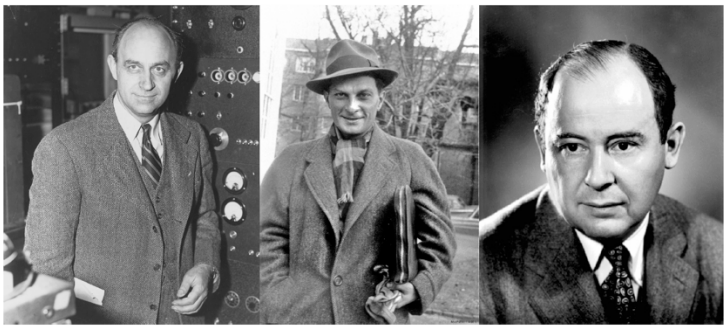
While no elementary antiderivative exists for these functions, some of the integrals can be expressed using special functions.

4.2 Types of Integral Approximations

- asymptotic expansions
- deterministic numerical approximations (Newton-Cotes quadrature, Romberg integration, Gaussian quadrature)
- Monte Carlo integration – simulation-based numerical approximation utilizing randomness
- Monte Carlo integration is slower with approximation error of order $O(n^{-1/2})$ *for any dimension*, but methods may require large samples for high-dimensions (to get an acceptable standard error).

4.3 Monte Carlo Integration Methods

Monte Carlo methods are computational tools characterized by the use of random number generators to obtain a numerical approximation to an unknown quantity.



- Enrico Fermi (1901-1954, physicist)
- Stanslaw Ulam (1909-1984, mathematical physicist)
- John von Neumann (1904-1957, everything)

Monte Carlo integration is a method for approximating integrals (and hence approximate Bayesian inference). Since Monte Carlo methods utilize random sampling, we require a source of randomness .

4.3.1 Random Number Generators

Question: How do we generate random numbers from some set or interval?
More convenient question: how to generate a stream of independent Unif(0; 1) variables

4.3.1.1 Truly random number generators from physical processes :

- coin tosses, dice, etc.
- build a device that delivers a random number from the outcome of some physical process (e.g. radioactive particle emission) which is believed to be truly random

Some disadvantages:

1. cannot re-run a simulation after changing some parameters or encountering an error; a physical random number generator cannot be restarted, so we would have to store the entire sequence of numbers (lots of storage)
2. usually not fast enough

4.3.1.2 Pseudo-random number generators

A computational alternative to physical random number generators is pseudo-random number generators.

- random numbers are **simulated** using an *algorithm*
- such numbers are not truly random; they are *pseudo-random*
- 'pseudo-random' means the sequence behaves *as if* it were random, in that it can 'fool' an arsenal of tests for randomness
- the idea is to generate a sequence of random numbers $x_1, x_2, \ldots$ and convert them to $u_i \in [0, 1]$ and hopefully have that $u_1, u_2, \ldots$ are approximately uniformly distributed

Since pseudo-random number generators dominate statistical practice, we just say RNG and 'pseudo' is understood to be implied.

John von Neumann: *Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.*

Linear Congruential Generators

An RNG based on simple recursions utilizing *modular arithmetic* is the linea congruential generator (LCG):

$$x_i = a_0 + a_1 x_{i-1} \mod M$$

With $a_0 = 0$, this is the multiplicative congruential generator (MCG):

$$x_i = a_1 x_{i-1} \mod M.$$

An LCG must be slower than a MCG, and not really better quality. Thus MCGs are more commonly used.

Generalization of MCG is **multiple recursive generator (MRG)**

$$x_i = a_1 x_{i-1} + a_2 x_{i-2} + \cdots + a_k x_{i-k} \mod M, \quad k \geq 1, \quad a_k \neq 0$$

All of these methods produce sequences of integer values modulo M, i.e.

$x_i \in \{0, 1, \ldots, M-1\}$.

4.3.2 Inverse Transform Sampling Method

- when F^{-1} is available for the target density, then this method is exact

Inverse Transform Sampling Method

Example: Generating Exponential:

Goal: generate $X \sim \text{Exp}(\lambda)$, with CDF $F_X(y) = \int_0^y \lambda e^{-\lambda x} dx = 1 - e^{-\lambda y}$.

- We have $F_X^{-1}(y) = -\frac{\log(1-y)}{\lambda}$.
- Now draw $U \sim \text{Unif}(0, 1)$.
- Then

$$X = F_X^{-1}(U) = -\frac{\log(1-U)}{\lambda} = -\frac{\log U}{\lambda}.$$

In many Bayesian inference problems, there are two frequently-encountered problems which make the above methods difficult or inefficient.

1. Sometimes the problem is very high-dimensional; we can produce multivariate random variates, but it would be useful to have some means of decomposing a high-dimensional problem into a sequence of smaller problems.
2. Often we only know f up to some constant of proportionality, which is non-trivial to compute, and perhaps numerical approximations are costly or inaccurate.

A revolutionary idea: introduce Markov chains, which have some very helpful convergence properties

4.3.3 Generating i.i.d. samples using accept-reject method

Suppose the target $f(x)$ cannot be sampled from directly.

von Neumann's idea: rejection sampling

Suppose there exists some other density $g(x)$ with the same support as $f(x)$ which is easy and fast to sample from, and there exists some constant M such that $f(x) \leq Mg(x)$ for all x. Then $0 \leq f(x)/Mg(x) \leq 1$ for all x.

1. Generate $Y \sim g(y)$ and generate $U \sim \text{Unif}(0, 1)$.
2. Set $X = Y$ (accept) if
$$U \leq \frac{f(Y)}{Mg(Y)},$$
otherwise go back to step 1 (reject).

Rejection sampling also called accept-reject method.

4.4 MCMC(Markov Chain Monte Carlo) method

4.4.1 Idea of Markov Chain

We want to sample from some target probability density f, usually a **posterior**, which we can typically write down as $f \propto$ likelihood $\times$ prior. This is our target.

- Goal: construct a Markov chain in conjunction with likelihood $\times$ prior that has the target f as its stationary distribution, and consider Monte Carlo algorithms which sample from f by utilizing this Markov chain

- a Markov chain is a **random or stochastic process**: a collection of random variables *indexed (and ordered) by time*, where the set of possible states of the process is called the **state space** and each variable takes values on the state space

A Markov chain $\{X^{(t)}\}$ is a sequence of dependent random variables

$$X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(t)}, \ldots$$

such that the probability distribution of $X^{(t)}$ *given the past variables* depends only on $X^{(t-1)}$.

- this conditional probability distribution is the **transition or Markov kernel** K:

$$X^{(t+1)} | X^{(0)}, X^{(1)}, X^{(2)}, \ldots, X^{(t)} \sim K(X^{(t)}, X^{(t+1)}).$$

Example (Simple Random Walk)

$$X^{(t+1)} = X^{(t)} + \epsilon_t,$$

with $\epsilon_t \sim \mathcal{N}(0, 1)$ independently of $X^{(t)}$.
$\Rightarrow K(X^{(t)}, X^{(t+1)}) \equiv \mathcal{N}(X^{(t)}, 1)$

4.4.2 Stationary of Markov chains

We often will work with Markov chains which exhibit a very strong property called **stationarity**. This means that there exists a probability distribution f such that if $X^{(t)} \sim f$, then $X^{(t+1)} \sim f$.

- this means the transition kernel and stationary distribution must satisfy

$$\int_{\mathcal{X}} K(x, y) f(x) dx = f(y).$$

- the existence of a stationary distribution imposes a constraint on the kernel called irreducibility

Definition

A Markov chain is said to be **irreducible** if, regardless of the starting value $X^{(0)}$, there is a positive probability to eventually reach any part of the state space.

Definition

A Markov chain is said to be **recurrent** if the chain returns to any arbitrary part of the state space infinitely many times.

Failure of Convergence

We discuss convergence later (time permitting), but it is important to know when convergence never occurs in Bayesian problems: when the posterior is not proper.

Recall: when we use *improper* priors, there is no assurance that the posterior is proper.

- it is often the case that we have no idea of the posterior is proper or not, but we can still utilize MCMC methods to sample from that posterior
- if we're lucky, the Markov chains will diverge quickly and we can see that there is a problem
- however, it often happens that the chain appears stable even for tens or hundreds of thousands of iterations, and we don't let it run long enough to detect problems

4.4.3 Basic idea of MCMC

Combine Accept-reject method(Monte Carlo) with Markov chain

We like to assume that we can generate independent and identically distributed samples from a density f of interest.

Problem: standard methods for generating i.i.d. samples (inverse transform sampling, importance sampling, etc.) require that we know the target f

Advantages of using Markov chains: (a sequence of *dependent* variables)

- convergence properties of Markov chains can be exploited to make things easier
- minimal requirements on f
- allows for decompositions of high-dimensional sampling problem into sequence of smaller problems

4.5 Metropolis-Hastings Algorithm

4.5.1 Motivating the Metropolis-Hastings Algorithm

Metropolis, Rosenbluth, Rosenbluth, Teller & Teller (1953, *J. Chem. Physics*) sought to simulate random draws from the posterior $p(\theta|y)$ by constructing a Markov chain possessing these attributes:

- should have the same state space as θ
- should be easy to simulate from
- stationary distribution should be the posterior $p(\theta|y)$
- should be able to ignore the normalizing constant in $p(\theta|y)$ to implement the algorithm, which means $p(\theta|y)$ should appear only through ratios of the form

$$\frac{p(\theta|y)}{p(\theta'|y)}$$

so that the normalizing constant cancels

4.5.2 Idea of Metropolis-Hastings

The Metropolis-Hastings algorithm generates correlated variables from a Markov chain.

- the target density f is associated with a 'working' conditional density, $q(y|x)$, that can be easily simulated in practice
- $q(\cdot|x)$ is arbitrary except that it must satisfy:
  1. the ratio $f(y)/q(y|x)$ is known up to a constant *independent of* x
  2. $q(\cdot|x)$ is disperse enough to explore the entire support of f

  Note that the Markov kernel q is not the Markov kernel K of the algorithm.

4.5.3 Magic of M-H:

for every given q , we can construct a M-H kernel K such that f is its stationary distribution.

4.5.4 Generic Metropolis-Hastings Algorithm

Given an initial value $x^{(t)}$ and conditional density $q(y|x^{(t)})$,

1. Generate $Y_t \sim q(y|x^{(t)})$.
2. Set

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \rho(x^{(t)}, Y_t), \\ x^{(t)} & \text{with probability } 1 - \rho(x^{(t)}, Y_t) \end{cases}$$

where

$$\rho(x, y) = \min\left\{ \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)}, 1 \right\}.$$

- q is the **proposal** or **candidate** distribution
- $\rho(x, y)$ is the acceptance probability; Hastings (1970, *Biometrika*) proved this choice gives you the correct stationary distribution for the chain

4.5.5 Independent Metropolis-Hastings

The generic M-H algorithm allows a proposal q which depends only on the current state of the chain.

- if we require q to be independent of the current state, i.e. $q(y|x) = g(y)$, we have a special case

Given initial state $x^{(t)}$

1. Generate $Y_t \sim g(y)$.
2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \min\left\{ \frac{f(Y_t)g(x^{(t)})}{f(x^{(t)})g(Y_t)}, 1 \right\} \\ x^{(t)} & \text{otherwise} \end{cases}.$$

This is like a generalization of rejection sampling.

4.5.6 Comparison of rejection with M-H

- rejection sample is i.i.d.; M-H sample is not–even though Y_t's generated independently–because acceptance probability of Y_t depends on $X^{(t)}$
- M-H involves repeated occurrences of the same value since rejection of Y_t results in repetition of $X^{(t)}$ at time $t + 1$
- rejection sampling requires calculation of the upper bound $\sup_x f(x)/g(x) \leq M$, which is not required for M-H

4.5.7 Generic approach: random walk Metropolis-Hastings

Choose some initial values (e.g. the MLEs), then alternate sampling from these two normal distributions.

- simulate Y_t according to
$$Y_t = X^{(t)} + \varepsilon_t$$
where $\varepsilon_t \sim g$ is random, independent of $X^{(t)}$ (e.g. uniform or normal)
- for instance $g \sim \mathcal{N}(0, \tau^2)$, so that $Y_t \sim \mathcal{N}(X^{(t)}, \tau^2)$
- the proposal $q(y|x)$ is of the form $g(y - x)$
- the Markov chain associated with q is a **random walk** when g is symmetric around zero
- However the M-H Markov chain is not a random walk, due to the acceptance step

RWMH

Given initial $x^{(t)}$

1. Generate $Y_t \sim g(y - x^{(t)})$.
2. Take
$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \min\{1, f(Y_t)/f(x^{(t)})\} \\ x^{(t)} & \text{otherwise.} \end{cases}$$

The acceptance probability does not depend on g, i.e. for a given pair $(x^{(t)}, y_t)$, the probability of acceptance is the same whether g is any symmetric density (student-t, normal, uniform, Cauchy, etc.).

- However, different choices of g will result in different ranges for Y_t and hence different *acceptance rates*

RWMH: univariate gamma density (with coda analysis)

```
library(coda)
mh.gamma <- function(n.sims, start, burnin, cand.sd, shape, rate){
  theta.cur <- start
  draws <- c()
    theta.update <- function(theta.cur, shape, rate){
    theta.can <- rnorm(1, mean=theta.cur, sd=cand.sd)
    accept.prob <- dgamma(theta.can, shape=shape,
      rate=rate)/dgamma(theta.cur, shape=shape, rate=rate)
    if(runif(1) <= accept.prob)
      theta.can
    else
      theta.cur
  }
  for(i in 1:n.sims){
    draws[i] <- theta.cur <- theta.update(theta.cur, shape = shape,
                                          rate = rate)
  }
  res <- mcmc(draws[(burnin + 1):n.sims])
  cat("Acceptance Rate:", 1-rejectionRate(res), "\n")
  return(res)
}
```

4.5.8 2-D RWMH

Consider a single observation (y_1, y_2) from bivariate normal population with mean $\theta = (\theta_1, \theta_2)^T$ and known covariance matrix $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$.

- use uniform prior on θ, then posterior is

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \Big| y \sim \mathcal{N}\left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right).$$

Choose some initial values (e.g. the MLEs), then alternate sampling from these two normal distributions.

Refer to HW3

2. Write your own function to perform random walk Metropolis-Hastings sampling (with $10,000$ samples) from a density which is proportional to the above gamma density times Jeffreys prior. To get full credit, you must add comments to each step of the code to explain what is happening. Some guidelines:

(a) Remember that the default parameters for the `gamma` distribution in `R` are not the same as the usual gamma density above. Make sure you specify the gamma density as above in `R`.

(b) You can use any symmetric proposal density that you want.

(c) Make sure to save the output.

Solution.

See the following R code of random walk Metropolis-Hastings sampling and detailed comments included.

```
# generate sample data from a gamma distribution
set.seed(888)
alpha <- 1
beta <- 1
n <- 20
X <- rgamma(n, shape=alpha, rate=beta)

# define Jeffreys prior derived above
prior <- function(alpha,beta){
return (sqrt(trigamma(alpha)*alpha-1)/beta)
}

# define the target posterior function
target <- function(alpha,beta,X){
gamma <- dgamma(X,shape=alpha,rate=beta)
log_likelihood <- sum(log(gamma))
log_prior <- log(prior(alpha,beta))
return(exp(log_likelihood+log_prior))
}
```

```r
# use maximum-likelihood fitting function in r to generate initial values
require(MASS)

mle_fit <- fitdistr(X, "gamma",start=list(shape = 1, rate = 1))

alpha0 <- mle_fit$estimate[1]
beta0 <- mle_fit$estimate[2]
start <- unname(c(alpha0,beta0))

# spefiсy number of samples
n.sims <- 10000

# use 2-D normal as candidate distribution and sepcify variance-convariance matrix
cand.sd <- unname(diag(mle_fit$sd))

# define the random walk Metropolis-Hastings sampling function
rwmh <- function(target, n.sims, start, burnin, cand.sd, X)
{
require(MASS)
library(coda)
theta.cur <- start
draws <- matrix( NA, nrow = n.sims, ncol = 2)
for(i in 1:n.sims){
theta.can <- mvrnorm(1, theta.cur, cand.sd)
if (theta.can[1]>0 && theta.can[2]>0)
if(runif(1) <= min(1, target(theta.can[1], theta.can[2], X)
/target(start[1], start[2], X)))
theta.cur <- theta.can
draws[i, ] <- theta.cur
}
return(mcmc(unname(draws[(burnin + 1):n.sims, ])))
}

rwmh.draws <- rwmh(target, n.sims, start, burnin=0, cand.sd, X)
```

3. Use the **coda package** to give traceplots, autocorrelation function plots and perform all 4 diagnostic checks in the lecture notes (Gelman & Rubin, Geweke, Raftery & Lewis, and Heidelberg & Welch). Interpret these results.

Solution.

See the following R code for traceplots, autocorrelation plots and 4 different diagnostic checks.

```
# generate samples by rwmh function defined above
library(coda)
set.seed(222)
rwmh.draws <- rwmh(target, n.sims, start, burnin=1000, cand.sd, X)

# Summarizing the Posterior Density
summary(rwmh.draws)

##
## Iterations = 1:9000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 9000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##       Mean     SD Naive SE Time-series SE
## [1,] 1.588 0.4702 0.004956        0.02886
## [2,] 1.428 0.5073 0.005348        0.03394
##
## 2. Quantiles for each variable:
##
##        2.5%   25%   50%   75% 97.5%
## var1 0.7822 1.240 1.552 1.916 2.485
## var2 0.5953 1.065 1.365 1.731 2.587

# traceplots of alpha and beta
```

Also final test Problem 2

```
# RW
TwoDRW <- function(sims, burnin){
  library(coda);
  require(MASS)
```

```
 # generate some data
 n = 25;
 data_X = rnorm(n, mean = 2, sd = 2);
 #posterior probability
 post_prob = function(x){
  prior = x[2]^(-2);
  m=1;
  for(j in data_X){
   m = m * dnorm(j, mean=x[1],sd=x[2])
  }
  prob_pos=m*prior;
  return (prob_pos)
 }

 #initial point
 start =c(2,2);

 # use 2-D normal as candidate distribution and sepcify variance-convariance matrix
 mle_fit <- fitdistr(data_X, "normal")
 cand.sd <- unname(diag(mle_fit$sd))

 # define the random walk Metropolis-Hastings sampling function
 theta.cur <- start
 draws <- matrix( NA, nrow = sims, ncol = 2)
 for(i in 1:sims){
  theta.can <- mvrnorm(1, theta.cur, cand.sd)
  if (theta.can[1]>0 && theta.can[2]>0)
   if(runif(1) <= min(1, post_prob(theta.can)
                  /post_prob(theta.cur)))
    theta.cur <- theta.can
   draws[i, ] <- theta.cur
 }
 return(mcmc(unname(draws[(burnin + 1):sims, ])))
}

# With respect to Two D random walk Metropolis Hastings
rwmh.draws <- TwoDRW(sims  = 13000, burnin = 3000)
cat("Acceptance Rate:", 1-rejectionRate(rwmh.draws), "\n")

# Plot the trace and the marginal posterior density
plot(rwmh.draws)
hist(rwmh.draws[,1])
hist(rwmh.draws[,2])

rwmh.draws1 <- TwoDRW(sims  = 13000, burnin = 3000)
rwmh.draws2 <- TwoDRW(sims  = 13000, burnin = 3000)
rwmh.draws3 <- TwoDRW(sims  = 13000, burnin = 3000)
rwmh.draws4 <- TwoDRW(sims  = 13000, burnin = 3000)
```

```
rwmh.draws5 <- TwoDRW(sims  = 13000, burnin = 3000)
rwmh.list <- list(rwmh.draws1, rwmh.draws2, rwmh.draws3, rwmh.draws4, rwmh.draws5)

# Plotting how PSRF Changes through Iteration
gelman.diag(rwmh.list)
gelman.plot(rwmh.list)
```

4.5.9 Gibbs Sampling

Suppose the parameter vector θ can be divided into d subvectors $\theta = (\theta_1, \ldots, \theta_d)^T$.

- an iteration of the Gibbs sampler draws values of each subvector, conditional on the values of all the other subvectors, i.e. there are d steps in iteration t
- this is possible when we can explicitly write down the conditional posterior distribution of each subvector, i.e.

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

with $\theta_{(-j)}^{t-1}$ all the components except for θ_j, at their current values:

$$\theta_{(-j)}^{t-1} = (\theta_1^t, \ldots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \ldots, \theta_d^{t-1})^T$$

which are the iteration t values for those subvectors *already* updated and the iteration $(t-1)$ values for those subvectors not yet updated

- most often applied when the the conditional distributions are conjugate distributions which are easy to simulate from

- conditional posterior distributions are given by

$$\theta_1 | \theta_2, y \sim \mathcal{N}(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2),$$

$$\theta_2 | \theta_1, y \sim \mathcal{N}(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2).$$

Choose some initial values (e.g. the MLEs), then alternate sampling from these two normal distributions.

Problem1 in final test

```
Gibbs <- function(sims, burnin){
 library(coda);
 # generate some data
 n = 25;
 data_X = rnorm(n, mean = 2, sd = 2);
 #posterior probability
 post_prob = function(x){
  prior = x[2]^(-2);
  m=1;
```

```
   for(j in data_X){
     m = m * dnorm(j, mean=x[1],sd=x[2])
   }
   prob_pos=m*prior;
   return (prob_pos)
  }
  #initial point
  theta.current <- c(2, 2)
  #define RWMH function
  theta.mh <- matrix(NA, nrow = sims, ncol = 2)
  theta.update <- function(index,theta.current) {
   if (index == 1){
     theta.can <- rnorm(1, theta.current[1], 1);
     theta.can =c(theta.can,theta.current[2]);
   }
   else{
     theta.can <- rgamma(1, theta.current[2], 1);
     theta.can =c(theta.current[1],theta.can);
   }

   accept.prob <- min(1,post_prob(theta.can)/post_prob(theta.current))
   if(runif(1) <= accept.prob)
     theta.can
   else
     theta.current
  }

  for(i in 1:sims){
   theta.current <- theta.update(1, theta.current);
   theta.mh[i,1] = theta.current[1];
   theta.current<- theta.update(2, theta.current);
   theta.mh[i,2] = theta.current[2]
  }
  res <- mcmc(theta.mh[(1+burnin):sims,])
  cat("Acceptance Rate:", 1-rejectionRate(res), "\n")
  return(res)
}

# With respect to gibbs sampling
gibbs.draws <- Gibbs(sims  = 11000, burnin = 1000)

# Plot the trace and the marginal posterior density
plot(gibbs.draws)
hist(gibbs.draws[,1])
hist(gibbs.draws[,2])

gibbs.draws1 <- Gibbs(sims  = 11000, burnin = 1000)
gibbs.draws2 <- Gibbs(sims  = 11000, burnin = 1000)
```

```
gibbs.draws3 <- Gibbs(sims  = 11000, burnin = 1000)
gibbs.draws4 <- Gibbs(sims  = 11000, burnin = 1000)
gibbs.draws5 <- Gibbs(sims  = 11000, burnin = 1000)
gibbs.list <- list(gibbs.draws1, gibbs.draws2, gibbs.draws3, gibbs.draws4, gibbs.draws5)

# Plotting how PSRF Changes through Iteration
gelman.diag(gibbs.list)
gelman.plot(gibbs.list)
```

4.5.10 Comments on Mixing of Chain

The mixing time of a Markov chain is the number of steps (amount of time) until the chain is close to the stationary distribution. Mixing measures how well the chain moves around around the parameter space.

Fast mixing means the the chain will reach the stationary distribution quickly, regardless of the starting values.

With perfect mixing, MC samples can move from one region of the state space to any other in one step.

Formal definition requires measure theory.

- we can do some visual inspections in practice, and these must be done for *every parameter* (e.g. traceplots, running means plots)

4.5.11 Comments on Autocorrelation

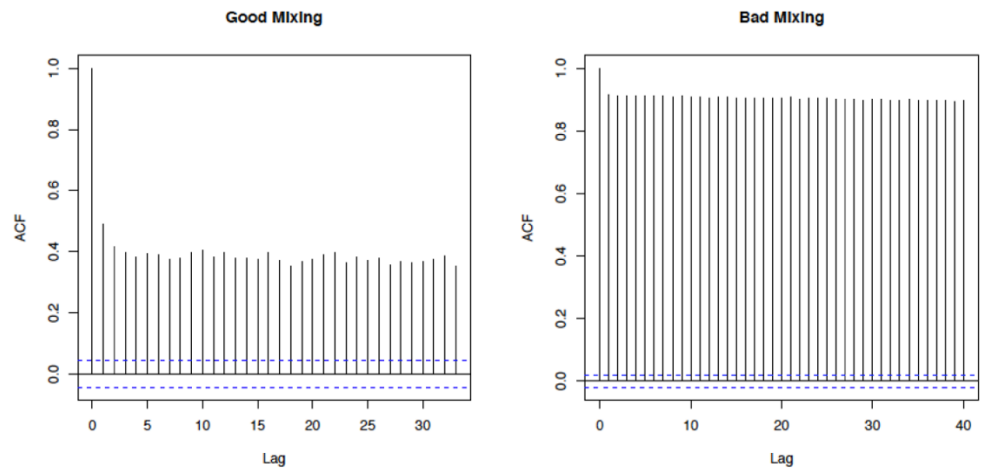
The lag k autocorrelation, denoted ρ_k, is the correlation between each draw and its kth lag:

$$\rho_k = \frac{\sum_{i=1}^{n-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

We expect the kth lag autocorrelation to be smaller as k increases (i.e. the 3rd and 25th draws should be less correlated than the 3rd and 4th draws).

- if autocorrelation is high for large values of k, this indicates high correlation between the draws and slow mixing.

Any Markov chain will have autocorrelation, since the current value depends on the previous one.
The Ergodic Theorem tells us that inference from correlated samples will still be valid, i.e. the Monte Carlo approximation of the posterior quantity of interest will converge to the true value (in probability).
Autocorrelation can still be present in chains that have converged.
Monte Carlo approximation still works, but you may need a very large sample (long chain) for the approximation to be accurate.



4.5.12 Tuning MCMC algorithms

You may come across optimal acceptance rates, e.g. 23.4%: while these arise from great ideas, they are not helpful in practical problems, since they assume posterior independence of the parameters (i.e. the target factors into a product of marginal densities).

Generally you want to tune the algorithms to get better performance in terms of:

- reducing correlation within each chain, i.e. autocorrelation or serial correlation
- reducing correlation between each chain
- acceptance rate

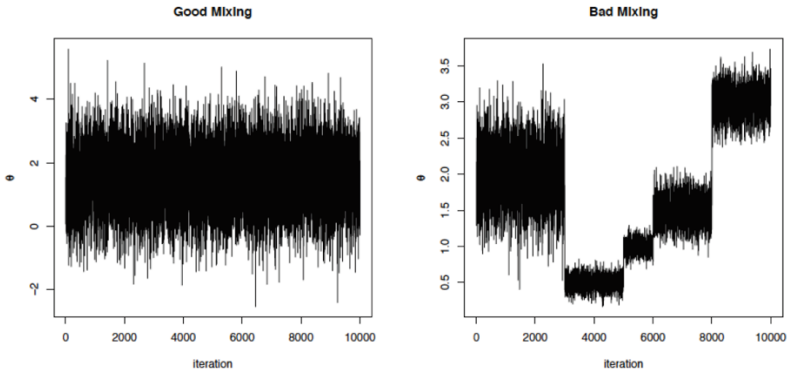
  too high can increase correlation

  too low takes forever to explore the state space (the support of the posterior)

Traceplots

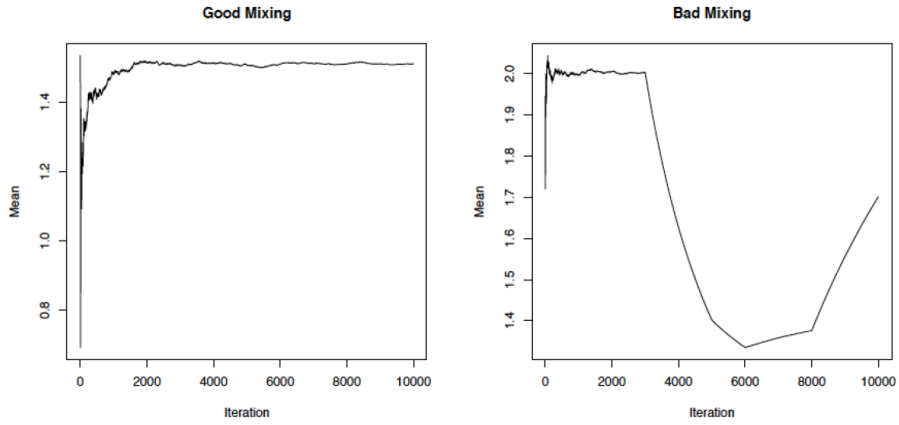
Plots iteration number against the value of the draw of the parameter for each iteration.

- helps us see if the chain gets stuck in a particular region of the parameter space, which indicates bad mixing



Running Means Plot with `ggmcmc`

Plot of iterations against mean of the draws up to that iteration.



5 Linear Regression Model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{k-1} X_{i,k-1} + \varepsilon_i$$

- response variable Y related to predictors/covariates/explanatory variables $X_1, \ldots, X_{k-1}$
- observe n predictor-response pairs $\{X_{ij}, Y_i\}$, $i = 1, \ldots, n$, $j = 0, 1, \ldots, k-1$
- $\varepsilon_i \overset{iid}{\sim} g(\cdot)$; usually $E(\varepsilon_i) = 0$ and $\text{Var}(\varepsilon_i) = \sigma^2$

Model in Matrix Form

$$Y = X\beta + \varepsilon$$

with

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & X_{11} & \cdots & X_{1,k-1} \\ 1 & X_{21} & \cdots & X_{2,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & \cdots & X_{n,k-1} \end{pmatrix},$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-1} \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

5.1 Frequentist Estimation by Least Squares

The least squares estimator is the solution to

$$\min_{\beta}(Y - X\beta)^T(Y - X\beta)$$

and this is equal to

$$\hat{\beta} = (X^T X)^{-1} X^T Y, \quad \hat{\sigma}^2 = \frac{(Y - X\hat{\beta})^T(Y - X\hat{\beta})}{n - k}$$

provided $(X^T X)^{-1}$ exists.

- if we assume $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_{n\times n})$, the likelihood function is

$$L(\beta, \sigma^2; Y, X) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left(-\frac{1}{2\sigma^2}(Y - X\beta)^T(Y - X\beta)\right)$$

- maximizing this is equivalent to the least squares problem

5.2 Towards a Bayesian Analysis

When a Bayesian encounters this model, she sees X and Y and immediately thinks there must be some sampling model for each of them:

$$p(X|\psi), \quad p(Y|\theta)$$

but in fact we have a joint density $f(x, y|\psi, \theta)$ and hence a joint likelihood $L(\psi, \theta)$.

- we need a joint prior $p(\psi, \theta)$
- Bayesians like to assume the distribution of X, $p(X|\psi)$, and hence the parameter ψ, provides no information about $p(Y|X, \theta)$
- i.e. prior independence of ψ and θ, $p(\psi, \theta) = p(\psi)p(\theta)$

If we assume that $p(\psi, \theta) = p(\psi)p(\theta)$ and, if the likelihood factors, then the posterior distribution factors

$$p(\psi, \theta|X, y) = p(\psi|X)p(\theta|X, y)$$

and the second factor (i.e. the regression model) can be studied by itself without information loss:

$$p(\theta|X, y) \propto p(\theta)p(y|X, \theta)$$

In a fixed design, the Xs are not random, and then $p(X)$ is known (there are no parameters ψ).

The joint posterior $p(\beta, \sigma^2|X, y)$ is proper provided that

1. $n > k$
2. $\text{rank}(X) = k$

5.2.1 Bayesian Linear Regression with Noninformative Prior

Common to use uniform prior on $(\beta, \log \sigma)$, which is the same as

$$p(\beta, \sigma^2|X) \propto \sigma^{-2}$$

Based on the above justification, and to facilitate the use of Gibbs sampling, we factor the joint posterior as

$$p(\beta, \sigma^2|X, y) = p(\beta|\sigma^2, X, y)p(\sigma^2|X, y)$$

Start by finding the posterior for β, conditional on σ, then find marginal distribution of σ^2.

5.2.1.1 *Conditional posterior $\boldsymbol{\beta|\sigma}$*

$\beta|\sigma$ is the exponential of a quadratic form in $\beta \Rightarrow$ conditional posterior

$$\beta|\sigma, X, y \sim \mathcal{N}(\hat{\beta}, (X^T X)^{-1}\sigma^2)$$

which can be seen noting that $\hat{\beta} = (X^T X)^{-1} X^T Y$

5.2.1.2 Marginal posterior of σ^2

Written as

$$p(\sigma^2|y) = \frac{p(\beta, \sigma^2|X, y)}{p(\beta|\sigma^2, X, y)}$$

which is a scaled inverse-χ^2 so that

$$\sigma^2 \sim \text{Inv-}\chi^2(n-k, s^2)$$

with

$$s^2 = \frac{1}{n-k}(Y - X\hat{\beta})^T(Y - X\hat{\beta})$$

- marginal posterior of $\beta|X, y$, found by *averaging over* σ, is multivariate t with $(n-k)$ degrees of freedom (and some covariance matrix)
- in practice we use MCMC by drawing from σ and then drawing $\beta|\sigma$, so we don't really need to use $\beta|X, y$ explicitly

5.2.1.3 Sampling from the posterior(joint)

To sample from $p(\beta, \sigma^2|X, y)$,

1. Compute $\hat{\beta}$ and $(X^T X)^{-1}$.
2. Compute s^2.
3. Draw σ^2 from scaled inverse-χ^2 distribution.
4. Draw β from multivariate normal distribution above.

If we know the full conditional distributions $p(\beta|\sigma^2, X, Y)$ and $p(\sigma^2|\beta, X, Y)$, we can sample from the joint posterior $p(\beta, \sigma^2|X, Y)$ using the Gibbs sampler:

Initialize $\beta_{(1)}$, $\sigma^2_{(1)}$

For $t = 1 : T$

$$\beta_{(t+1)} \sim p(\beta_{(t)}|\sigma^2_{(t)}, X, Y)$$
$$\sigma_{(t+1)} \sim p(\sigma^2_{(t)}|\beta_{(t+1)}, X, Y)$$

END

5.3 generalized linear models (GLMs):

Motivation of generalized linear models (GLMs): unify different approaches to regression modeling for responses which are not necessarily normal

A GLM has 3 components:

Distribution for response (often exponential family).

Linear predictor $\eta = X\beta$.

Link function $g(\cdot)$ such that $E(y) = \mu = g^{-1}(\eta)$.

Exponential families are easier to work with (can make variance calculation easier).

- GLMs fit by iteratively reweighted least squares
- posterior usually not available in closed form; typically fit by Laplace approximation or MCMC

5.3.1 GML example

Example: Linear Model

The classical linear model is also a GLM.

$$y_i = \beta_0 + \beta x_i + \varepsilon_i,$$

$$E(y_i) = \beta_0 + \beta x_i$$

Distribution $y_i \sim \mathcal{N}(\mu_i, \sigma^2)$

Linear predictor (continuous or discrete) and linear in parameters; predictors can also be transformed (still linear in transformed predictors)

Link function identity: $\eta = g(E(y_i)) = g(\mu_i) = E(y_i)$: modeling the mean directly, the simplest link function

Logit and Logistic Functions

The logit function of a number $q \in (0, 1)$ is

$$\text{logit}(q) = \log\left(\frac{q}{1-q}\right) = \log(q) - \log(1-q).$$

Hence when q is a probability, with $q/(1-q)$ the **odds**, $\text{logit}(q)$ is the log odds.

The logistic function of any number $d \in \mathbb{R}$ is the inverse logit function

$$\text{logit}^{-1}(d) = \text{logistic}(d) = \frac{1}{1+e^{-d}} = \frac{e^d}{1+e^d}.$$

The logit maps from $(0, 1) \mapsto \mathbb{R}$; the inverse logit (logistic) maps from $\mathbb{R} \mapsto (0, 1)$.

Example: Binary Logistic Regression

Models binary response y as function of k explanatory variables $X = (X_1, \ldots, X_k)$ (note we need values between 0 and 1)

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta x_i,$$

where $\pi_i = E(y_i) = \Pr(y_i = 1)$.

Distribution $y_i \sim \text{Binomial}(n, \pi)$ with π the probability of success

Linear predictor same as above; predictors can be transformed as usual

Link function logit link

$$\eta = \text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$$

More generally, the logit link is a model for the log odds of the mean, and the mean here is π

Example: Log-Linear Poisson Model for Count Data Response

Distribution $y_i \sim \text{Poisson}(\lambda)$

Linear predictor same as above

Link function $\eta = \log \lambda = E(y)$

Connects the rate $\lambda_i = E(y_i)$ of Poisson distribution with linear predictor $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$ using

$$\lambda_i = \exp(\eta_i) = \exp(\beta_0)\exp(\beta_1 x_{i1}) \cdots \exp(\beta_k x_{ik})$$

or in log-linear form through

$$\log(\lambda_i) = \eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$$

Effect of covariates on the rate λ is exponentially multiplicative.

5.3.2 Frequentist's approach(R code)

```
# initiate date
icecream <- data.frame(
  temp = c(11.9,14.2,15.2,16.4,17.2,18.1,
           18.5,19.4,22.1,22.6,23.4,25.1),
  units=c(185L, 215L, 332L, 325L, 408L, 421L,
          406L, 412L, 522L, 445L, 544L, 614L)
);
basicPlot <- function(...){
  plot(units ~ temp, data=icecream, bty="n", lwd=2,
       main="Number of ice creams sold", col="#00526D",
       xlab="Temperature (Celsius)",
       ylab="Units sold", ...)
  axis(side = 1, col="grey")
  axis(side = 2, col="grey")
}
basicPlot()

# Gaussian fit

lin.mod <- glm(units ~ temp, data=icecream,
               family=gaussian(link="identity"))
lin.sig <- summary(lin.mod)$dispersion
lin.pred <- predict(lin.mod)

library(arm) # for 'display' function only
display(lin.mod)
# Log-linear Fit
log.lin.mod <- glm(log(units) ~ temp, data=icecream,
                   family=gaussian(link="identity"))
display(log.lin.mod)
log.lin.sig <- summary(log.lin.mod)$dispersion
```

```
log.lin.pred <- exp(predict(log.lin.mod) + 0.5*log.lin.sig)
basicPlot()
From Ice_cream.R
# plot line for Gaussian
lines(icecream$temp, log.lin.pred, col="red", lwd=2)
lines(icecream$temp, lin.pred, col="blue", lwd=2)
legend(x="topleft", bty="n", lwd=c(2,2), lty=c(NA,1),
      legend=c("observation", "log-transformed LM"),
      col=c("#00526D","red"), pch=c(1,NA))




# Poission fit

pois.mod <- glm(units ~ temp, data=icecream,
              family=poisson(link="log"))
display(pois.mod)
pois.pred <- predict(pois.mod, type="response")
predict(pois.mod,newdata=data.frame(temp=32),
       type="response")
# Plot for poisson
basicPlot()
lines(icecream$temp, pois.pred, col="blue", lwd=2)
legend(x="topleft", bty="n", lwd=c(2,2), lty=c(NA,1),
      legend=c("observation", "Poisson (log) GLM"),
      col=c("#00526D","blue"), pch=c(1,NA))



# Binomial regression fit

market.size <- 800
icecream$opportunity <- market.size - icecream$units
bin.glm <- glm(cbind(units, opportunity) ~ temp, data=icecream,
             family=binomial(link = "logit"))
display(bin.glm)
# Plot Binomial fit
bin.pred <- predict(bin.glm, type="response")*market.size
basicPlot()
lines(icecream$temp, bin.pred, col="purple", lwd=2)
legend(x="topleft", bty="n", lwd=c(2,2), lty=c(NA,1),
      legend=c("observation", "Binomial (logit) GLM"),
      col=c("#00526D","purple"), pch=c(1,NA))
# Sales at 0 Celsius
plogis(coef(bin.glm)[1])*market.size
# Sales at 35 Celsius
plogis(coef(bin.glm)[1]+coef(bin.glm)[2]*35)*market.size



## plot together

temp <- 0:35
p.lm <- predict(lin.mod, data.frame(temp=temp), type="response")
p.log.lm <- exp(predict(log.lin.mod, data.frame(temp=0:35), type="response") +
```

```
                0.5 * summary(log.lin.mod)$dispersion)
p.pois <- predict(pois.mod, data.frame(temp=temp), type="response")
p.bin <- predict(bin.glm, data.frame(temp=temp), type="response")*market.size
basicPlot(xlim=range(temp), ylim=c(-20,market.size))
lines(temp, p.lm, type="l", col="orange", lwd=2)
lines(temp, p.log.lm, type="l", col="red", lwd=2)
lines(temp, p.pois, type="l", col="blue", lwd=2)
lines(temp, p.bin, type="l", col="purple", lwd=2)
legend(x="topleft",
       legend=c("observation",
                "linear model",
                "log-transformed LM",
                "Poisson (log) GLM",
                "Binomial (logit) GLM"),
       col=c("#00526D","orange", "red",
             "blue", "purple"),
       bty="n", lwd=rep(2,5),
       lty=c(NA,rep(1,4)),
       pch=c(1,rep(NA,4)))


##Visualizing GLMs (function to produce 3d plots)
  glmModelPlot <- function(x, y, xlim,ylim, meanPred, LwPred, UpPred,
                           plotData, main=NULL){
    ## Based on code by Arthur Charpentier:
    ## http://freakonometrics.hypotheses.org/9593
    par(mfrow=c(1,1))
    n <- 2
    N <- length(meanPred)
    zMax <- max(unlist(sapply(plotData, "[[", "z")))*1.5
    mat <- persp(xlim, ylim, matrix(0, n, n), main=main,
                 zlim=c(0, zMax), theta=-30,
                 ticktype="detailed",box=FALSE)
    C <- trans3d(x, UpPred, rep(0, N),mat)
    lines(C, lty=2)
    C <- trans3d(x, LwPred, rep(0, N), mat)
    lines(C, lty=2)
    C <- trans3d(c(x, rev(x)), c(UpPred, rev(LwPred)),
                 rep(0, 2*N), mat)
    polygon(C, border=NA, col=adjustcolor("yellow", alpha.f = 0.5))
    C <- trans3d(x, meanPred, rep(0, N), mat)
    lines(C, lwd=2, col="grey")
    C <- trans3d(x, y, rep(0,N), mat)
    points(C, lwd=2, col="#00526D")
    for(j in N:1){
      xp <- plotData[[j]]$x
      yp <- plotData[[j]]$y
      z0 <- plotData[[j]]$z0
      zp <- plotData[[j]]$z
      C <- trans3d(c(xp, xp), c(yp, rev(yp)), c(zp, z0), mat)
      polygon(C, border=NA, col="light blue", density=40)
      C <- trans3d(xp, yp, z0, mat)
      lines(C, lty=2)
      C <- trans3d(xp, yp, zp, mat)
      lines(C, col=adjustcolor("blue", alpha.f = 0.5))
    }
```

```
  }

## 3D for "Linear regression"
  xlim <- c(min(icecream$temp)*0.95, max(icecream$temp)*1.05)
  ylim <- c(floor(min(icecream$units)*0.95),
            ceiling(max(icecream$units)*1.05))
    #lin.mod <- glm(units ~ temp, data=icecream,
    #           family=gaussian(link="identity"))
  par(mfrow=c(2,2))
  plot(lin.mod)
  title(outer=TRUE, line = -1,
        main = list("Linear regression",
                    cex=1.25,col="black", font=2))
  meanPred <- predict(lin.mod, type="response")
  sdgig <- sqrt(summary(lin.mod)$dispersion)
  UpPred <- qnorm(.95, meanPred, sdgig)
  LwPred <- qnorm(.05, meanPred, sdgig)
  plotData <- lapply(
    seq(along=icecream$temp),
    function(i){
      stp <- 251
      x = rep(icecream$temp[i], stp)
      y = seq(ylim[1], ylim[2], length=stp)
      z0 = rep(0, stp)
      z = dnorm(y, meanPred[i], sdgig)
      return(list(x=x, y=y, z0=z0, z=z))
    }
  )
  glmModelPlot(x = icecream$temp, y=icecream$units,
               xlim=xlim, ylim=ylim,
               meanPred = meanPred, LwPred = LwPred,
               UpPred = UpPred, plotData = plotData,
               main = "Linear regression")

  #### 3D for "log-Linear regression"
  xlim <- c(min(icecream$temp)*0.95, max(icecream$temp)*1.05)
  ylim <- c(floor(min(icecream$units)*0.95),
            ceiling(max(icecream$units)*1.05))
  #lin.mod <- glm(log(units) ~ temp, data=icecream,
  #           family=gaussian(link="identity"))
  par(mfrow=c(2,2))
  plot(log.lin.mod)
  plot(log.lin.mod)
  title(outer=TRUE, line = -1,
        main = list("log Linear regression",
                    cex=1.25,col="black", font=2))

  meanPred <- log.lin.pred
  sdgig <- sqrt(exp(mean(log.lin.sig*meanPred)));
  UpPred <- qnorm(.95, meanPred, sdgig)
  LwPred <- qnorm(.05, meanPred, sdgig)
  plotData <- lapply(
    seq(along=icecream$temp),
    function(i){
      stp <- 251
      x = rep(icecream$temp[i], stp)
```

```
    y = seq(ylim[1], ylim[2], length=stp)
    z0 = rep(0, stp)
    z = dnorm(y, meanPred[i], sdgig)
    return(list(x=x, y=y, z0=z0, z=z))
  }
)
glmModelPlot(x = icecream$temp, y=icecream$units,
            xlim=xlim, ylim=ylim,
            meanPred = meanPred, LwPred = LwPred,
            UpPred = UpPred, plotData = plotData,
            main = "log Linear regression")

## 3D for "poisson regression"
xlim <- c(min(icecream$temp)*0.95, max(icecream$temp)*1.05)
ylim <- c(floor(min(icecream$units)*0.95),
          ceiling(max(icecream$units)*1.05))
#lin.mod <- glm(units ~ temp, data=icecream,
#           family=gaussian(link="identity"))
par(mfrow=c(2,2))
plot(pois.mod)
title(outer=TRUE, line = -1,
      main = list("Linear regression",
                  cex=1.25,col="black", font=2))
meanPred <- pois.pred
sdgig <- sqrt(summary(pois.mod)$dispersion)
UpPred <- qnorm(.95, meanPred, sdgig)
LwPred <- qnorm(.05, meanPred, sdgig)
plotData <- lapply(
  seq(along=icecream$temp),
  function(i){
    stp <- 251
    x = rep(icecream$temp[i], stp)
    y = seq(ylim[1], ylim[2], length=stp)
    z0 = rep(0, stp)
    z = dnorm(y, meanPred[i], sdgig)
    return(list(x=x, y=y, z0=z0, z=z))
  }
)
glmModelPlot(x = icecream$temp, y=icecream$units,
            xlim=xlim, ylim=ylim,
            meanPred = meanPred, LwPred = LwPred,
            UpPred = UpPred, plotData = plotData,
            main = "Poisson regression")
```

5.3.3 Bayesian GLMs(R code)

We use `brms` package to create prediction intervals for the four GLM models discussed above. Want to predict how much ice cream should be kept in stock when temperature is 35, such that you only run out of ice cream with posterior probability 2.5%.

Here the prior for σ is Cauchy, whereas in the coding above it was inverse gamma.

- should imply a small difference in σ as compared to above

Now we give traceplots and density plots for the MCMC samples.

```
##We use brms package to create prediction intervals for the four
#GLM models
temp <- c(11.9,14.2,15.2,16.4,17.2,18.1,18.5,19.4,22.1,22.6,23.4,25.1)
units <- c(185L,215L,332L,325L,408L,421L,406L,412L,522L,445L,544L, 614L)
library(brms)
# Linear model
lin.brm.mod <- brm(units ~ temp, family="gaussian")
# Log-transformed LM
log.brm.mod <- brm(log(units) ~ temp, family="gaussian")
# Poisson (log)
pois.brm.mod <- brm(units ~ temp, data=icecream,
            family=poisson(link="log"))
#Binomial (logit)
bin.brm.mod <- brm(units ~ temp, data=icecream,
           family=binomial(link = "logit"))
summary(lin.brm.mod)
summary(log.brm.mod)
summary(pois.brm.mod)
summary(bin.brm.mod)


# plot posterior distribution
plot(lin.brm.mod)
plot(log.brm.mod)
plot(pois.brm.mod)
plot(bin.brm.mod)
## Need 97.5% percentile of posterior predictive MCMC samples.
A <- function(samples){
as.matrix(samples[,c("b_Intercept" ,"b_temp")])
}
x <- c(1, 35)
prob <- 0.975
#lin reg
lin.samples <- posterior_samples(lin.brm.mod)
```

```
summary(lin.samples)
n <- nrow(lin.samples)
mu <- A(lin.samples) %*% x
sigma <- lin.samples[,"sigma_units"]
lin_pred <- rnorm(n, mu, sigma);
(lin.q <- quantile( lin_pred, prob))
hist(lin_pred)

#log-lin reg
log.lin.samples <- posterior_samples(log.brm.mod)
mu <- A(log.lin.samples) %*% x
sigma <- log.lin.samples[,"sigma_units"]
log.lin_pred <- exp(rnorm(n, mu + 0.5*sigma^2, sigma))
(log.lin.q <- quantile(log.lin_pred, prob))
hist(log.lin_pred)

#poisson reg
pois.samples <- posterior_samples(pois.brm.mod)
mu <- exp(A(pois.samples) %*% x)
pois_pred <- rpois(n, mu)
(pois.q <- quantile(pois_pred , prob))
hist(pois_pred)

#bin reg
bin.samples <- posterior_samples(bin.brm.mod)
mu <- (800 - exp(A(bin.samples) %*% x))/800
bin_pred <- rbinom(n,800, mu)
(bin.q <- quantile(bin_pred , prob))
hist(bin_pred)

percentiles <- c(lin.q, log.lin.q, pois.q, bin.q)
b <- barplot(percentiles,
        names.arg = c("Linear", "Log-transformed",
                "Poisson", "Binomial"),
        ylab="Predicted ice cream units",
        main="Predicted 97.5%ile at 35??C")
text(b, percentiles-75, round(percentiles))
```

5.4 hierarchical linear models

5.4.1 Basic idea

Basic idea of hierarchical linear models

Let j denote group, $j = 1, \ldots, g$ and i is observation, $i = 1, \ldots, n$.

Level 1 model:

$$Y_{ij} = \alpha_j + \beta_j X_{ij} + e_{ij}$$

Level 2 model:

$$\alpha_j = \gamma_\alpha + \delta_\alpha Z_j + u_{\alpha j},$$

$$\beta_j = \gamma_\beta + \delta_\beta + Z_j + u_{\beta j}$$

Thus the coefficients (intercept and slopes) in Level 1 model are modeled as response variables in Level 2 model. Can allow for random incercepts, random slopes, etc.

Assume $y \sim N(\tilde{X}\tilde{\beta}, \Lambda)$. A Bayesian analysis proceeds by assigning prior distributions to $\tilde{\beta}$ and Λ. In constructing the prior for $\tilde{\beta}$, consider the components β and u separately. Assume

$$\beta \sim N(\beta_0, \Sigma_\beta), \quad \text{and} \quad u \sim N(0, \Omega)$$

independently.

For the

- **fixed** effects β, we select β_0 and Σ_β while for the
- **random** effects u, we assign a prior for Ω.

Therefore we have created a hierarchical model for the random effects and thus refer to this as a *hierarchical linear model*.

Summary

These models are referred to as

- mixed-effect models,
- hierarchical linear models, or
- multi-level models.

The parameters for the prior distribution for the

- fixed effects are not learned and
- random effects are learned.

This corresponds to a non-Bayesian analysis learning a variance parameter for random effects.

5.4.2 Implement

Demo23

Initially, we could consider the model

$$y_{st} \overset{ind}{\sim} N(\beta_{s,0} + x_{st}\beta_{s,1}, \sigma_s^2)$$

where

- y_{st} is the mean log count $(+1)$ for species s at time t
- x_{st} is the year (minus 2005) for species s at time t

This model treats each species completely independently.

Random intercept, random slope model

A reasonable assumption is to treat these species exchangeably and put a distribution on the intercept and slope.

Then a *random intercept, random slope model* is

$$\begin{aligned} y_{st} &\overset{ind}{\sim} N(\beta_{s,0} + x_{st}\beta_{s,1}, \sigma^2) \\ \beta_s &\overset{ind}{\sim} N(\mu_\beta, \Sigma_\beta) \end{aligned}$$

where $\beta_s = (\beta_{s,0}, \beta_{s,1})'$ and σ^2, μ_β, and Σ_β are parameters to be estimated.

Notice that there is now a common variance for all species.
Equivalently,

$$y_j \overset{ind}{\sim} N(X_j\beta_j, \sigma_y^2 I_{n_j}), \quad \beta_j \overset{ind}{\sim} N(\mu_\beta, \Sigma_\beta)$$

where $y_j = (y_{1j}, \ldots, y_{n_j j})'$ and the $i^{(th)}$ row of X_j is $(1, x_{st})$.

Bayesian random intercept, random slope model

The model

$$\begin{aligned} y_{st} &\overset{ind}{\sim} N(\beta_{s,0} + x_{st}\beta_{s,1}, \sigma^2) \\ \beta_s &\overset{ind}{\sim} N(\mu_\beta, \Sigma_\beta) \end{aligned}$$

and a prior

$$p(\sigma, \mu_\beta, \Sigma_\beta) \propto p(\sigma)p(\mu_\beta)p(\Sigma_\beta)$$

and

- $\sigma \sim Ca^+(0, 1)$,
- $p(\mu_\beta) \propto 1$, and
- $\Sigma_\beta \sim$?

Conjugate prior for a covariance matrix

The natural conjugate prior for a covariance matrix is the *inverse-Wishart* distribution, which has density

$$p(\Sigma) \propto |\Sigma|^{-(\nu+d+1)/2} \exp\left(-\frac{1}{2}\text{tr}\left(S\Sigma^{-1}\right)\right)$$

with $\nu > d-1$ and S is a positive definite matrix. The expected value is

$$E[\Sigma] = \frac{S}{\nu - d - 1}$$

for $\nu > d+1$. We write $\Sigma \sim IW(\nu, S^{-1})$.

Special cases:

- If $\nu = d+1$ and S is diagonal, then each of the correlations in Σ has a marginal uniform prior.
- Jeffreys prior
$$p(\Sigma) = |\Sigma|^{-(d+1)/2}$$

Hierarchical model for the variances

The model

$$\begin{aligned} y_{st} &\overset{ind}{\sim} N(\beta_{s,0} + x_{st}\beta_{s,1}, \sigma_s^2) \\ \beta_s &\overset{ind}{\sim} N(\mu_\beta, \Sigma_\beta) \\ \sigma_s &\overset{ind}{\sim} LN(\mu_\sigma, \tau_\sigma) \end{aligned}$$

and a prior

$$p(\mu_\sigma, \tau_\sigma, \mu_\beta, \Sigma_\beta) \propto p(\mu_\sigma)p(\tau_\sigma)p(\mu_\beta)p(\Sigma_\beta)$$

and

- $p(\mu_\sigma) \propto 1$,
- $\tau_\sigma \sim Ca^+(0, 1)$,
- $p(\mu_\beta) \propto 1$, and
- Σ_β as before

6 The Bayes Factor

The **Bayes factor** is the ratio of the posterior probabilities of the null and alternative hypotheses *over* the ratio of the prior probabilities of the null and alternative hypotheses,

$$B_{01}^{\pi}(x) = \frac{Pr(\theta \in \Theta_0|x)}{Pr(\theta \in \Theta_1|x)} \Big/ \frac{\pi(\theta \in \Theta_0)}{\pi(\theta \in \Theta_1)}.$$

This measures how the odds of Θ_0 against Θ_1 change due to the observed data.

- simple interpretation is to compare B_{01} to 1 to assess the evidence against each model; in practice the comparison scale depends on the loss function

The Bayes factor for comparing M_k to M_l is

$$BF_{kl} = \frac{m(y|M_k)}{m(y|M_l)}$$

- observe that the crucial thing for computing Bayes factors is to be able to compute the **marginal likelihood** in each model
- recall this is a messy integral; until the late 1980s, we had trouble computing this, or even finding a decent estimate, in complex models (Laplace approximation)
- mid-1990s was the real breakthrough (MCMC estimation of marginal likelihood)

To estimate the marginal likelihood, we can use Laplace's method or the method of Chib (1995), with corresponding function notations, marginal.likelihood = c("Laplace","Chib95"), developed in R package MCMCpack.

6.1 computing Bayes factors and marginal likelihoods via Laplace approximation

Consider a model $M_\alpha = \{\mathcal{F}_\alpha, \Pi_i, \lambda_\alpha\}$ for some α in a countable index set A.

- usual way a Bayesian assesses evidence for or against some model is by computing the Bayes factor for the model of interest and some alternative model under consideration.

$$BF_{kl} = \frac{m(y|M_k)}{m(y|M_l)}$$

where the marginal likelihood for model M_i is

$$m(y|M_i) = \int f(y|\theta_i, M_i)\pi(\theta_i|M_i)d\theta_i,$$

where $f(y|\theta_i, M_i)$ is the likelihood under model i and $\pi(\theta_i|M_i)$ is the prior under model i.

Key issue: how to compute or accurately estimate the marginal likelihoods

problem:

Often the integral

$$m(y|M_i) = \int f(y|\theta_i, M_i)\pi(\theta_i|M_i)d\theta_i,$$

cannot be evaluated analytically.

- some common numerical methods are inefficient because when sample sizes are moderate or large, the integrand becomes highly peaked around its maximum, which *can be found by other methods*
- for example, quadrature methods which are initialized without knowing the maximum can encounter difficulty finding the region where the integrand mass is accumulating
- moreover in high dimensions, MCMC is often more efficient (see below)

In fact, the commonly-encountered examples for which the marginal likelihood can be evaluated analytically are restricted to exponential family models with conjugate priors (e.g. normal linear models)

Proposal 1: Laplace's method
Laplace approximation of integrals

Suppose we have an integral of the form

$$I(x) = \int_a^b e^{xg(t)} f(t) dt$$

for large x (i.e. as $x \to \infty$).

- we want an accurate approximation $\hat{I}(x)$ s.t. the ratio converges to 1 as $x \to \infty$
- clearly $\hat{I}(x)$ will depend on the two functions f, g, e.g. whether g is monotone or not, the boundary behaviors of f, g, etc.

Suppose g is strictly monotone and has nonzero derivative in the interval (a, b), then *integration by parts* yields

$$I(x) = \frac{1}{x} \frac{f(t)}{g'(t)} e^{xg(t)} \Big|_a^b - \frac{1}{x} \int_a^b \frac{d}{dt} \frac{f(t)}{g'(t)} e^{xg(t)} dt$$

- if one of $f(a)$, $f(b)$ is nonzero, the first term will dominate and then

$$I(x) \sim \frac{1}{x} \frac{f(b)}{g'(b)} e^{xg(b)} - \frac{1}{x} \frac{f(a)}{g'(a)} e^{xg(a)}$$

This is the simplest situation.

In practical situations, g may not be strictly monotone and may have zero derivative at one or more points in the interval (a, b).

- then we can't integrate by parts

Idea of Laplace's method: if g has a maximum at some unique c in (a, b), and if $f(c) \neq 0$, $g'' \neq 0$, then due to the large magnitude of the parameter x, the *dominant part* of the integral will come from a *neighborhood of* c.

Taylor expansion of g around c up to a quadratic term yields a normal kernel (density).

- the integral will not change much if the range of integration is changed from (a, b) to the real line
- upon normalizing the normal density, the factor $\sqrt{2\pi}$ appears and we have the approximation

$$\hat{I}(x) = \frac{\sqrt{2\pi} f(c) e^{x g(c)}}{\sqrt{-x g''(c)}}$$

  and as $x \to \infty$, $\hat{I}(x) \sim I(x)$
- intermediate steps of this derivation require that a, b are not stationary points of g, that $g'(c) = 0$ and that $g''(c) < 0$
- if either of the two boundary points a, b is a stationary point of g, then the approximating function will change to accomodate contributions from the boundary stationary points

If the interior local maximum of g is *not unique*, then $I(x)$ must be partitioned into subintervals separating the different maxima and summing over the terms to obtain a final approximation to $I(x)$.

Example 1

Let

$$I(x) = \int_{-\infty}^{\infty} e^{x(t - e^t)} dt,$$

(which equals $\Gamma(x)/x^x$ for a suitable change of variable in $I(x)$).

- try an asymptotic approximation for $I(x)$ as $x \to \infty$ by Laplace's method with $f(t) = 1$ and $g(t) = t - e^t$
- the only saddlepoint of g is $t = 0$, and $g''(t) = -e^t$

Therefore, the Laplace approximation is

$$\hat{I}(x) = \frac{\sqrt{2\pi} e^{-x}}{x}$$

6.2 computing Bayes factors and marginal likelihoods via MCMC

Simple Monte Carlo estimation of marginal likelihood

Basic idea: since

$$m(y) = \int f(y|\theta)\pi(\theta)d\theta,$$

draw $\theta_1, \ldots, \theta_m$ i.i.d. from $\pi(\theta)$.

- a Monte Carlo estimator of $m(y)$ is then

$$\frac{1}{m}\sum_{i=1}^{m} f(y|\theta^{(i)})$$

Common problem is large variance of this estimator.

Marginal likelihood from the Gibbs Sampler Output

- Chib (1995) Marginal likelihood from the Gibbs output. *JASA* **90**(432), 1313–1321.

Note the **basic marginal likelihood identity** (rearranging definition of posterior):

$$m(y) = \frac{f(y|\theta)\pi(\theta)}{\pi(\theta|y)}$$

- an identity since it holds *for any* θ (in the parameter space)
- easy to evaluate $f(y|\theta)$ and $\pi(\theta)$
- so to estimate $m(y)$ we only need to estimate the posterior $\pi(\theta|y)$

Decompose θ into two blocks (θ_1, θ_2) such that $\pi(\theta_1|\theta_2, y)$ and $\pi(\theta_2|\theta_1, y)$ are known (completely specified).

$$\pi(\theta_1, \theta_2|y) = \pi(\theta_2|\theta_1, y)\pi(\theta_1|y)$$

- Gibbs sampler gives dependent draws from joint posterior $\pi(\theta_1, \theta_2|y)$ and therefore marginally from $\pi(\theta_2|y)$; find the marginal posterior for θ_1 by

$$\begin{aligned}\pi(\theta_1|y) &= \int \pi(\theta_1|\theta_2, y)\pi(\theta_2|y)d\theta_2 \\ &\approx \frac{1}{G}\sum_{g=1}^{G} \pi(\theta_1|\theta_2^{(g)}, y)\end{aligned}$$

Such a marginal posterior estimator is (simulation) consistent; under regularity conditions $\hat{\pi}(\theta|y) \to \pi(\theta|y)$ almost surely as $G \to \infty$ (due to ergodic theorem)

General case (arbitrary number of blocks)

Decompose posterior at the point θ as

$$\pi(\theta|y) = \pi(\theta_1|y) \times \pi(\theta_2|\theta_1, y) \times \cdots \times \pi(\theta_B|\theta_1, \ldots, \theta_{B-1}, y)$$

where the last term is the marginal ordinate and can be estimated from the draws of the initial Gibbs run

- the other terms are the reduced conditional ordinates $\pi(\theta_r|\theta_1, \ldots, \theta_{r-1}, y)$ given by

$$\int \pi(\theta_r|\theta_1, \ldots, \theta_{r-1}, \theta_l(l > r), y) d\pi(\theta_{r+1}, \ldots, \theta_B|\theta_1, \ldots, \theta_{r-1}, y)$$

Estimate this by

$$\hat{\pi}(\theta_r|\theta_s(s < r), y) = G^{-1} \sum_{j=1}^{G} \pi(\theta_r|\theta_1, \ldots, \theta_{r-1}, \theta_l^{(j)}(l > r), y)$$

and estimate the joint density by $\prod_{r=1}^{B} \hat{\pi}(\theta_r|\theta_s(s < r), y)$

Bayes factor estimate

Typically the log of the marginal likelihood is estimated using the above method, yielding the estimate of B_{kl}

$$\hat{B}_{kl} = \exp\{\log \hat{m}(y|M_k) - \log \hat{m}(y|M_l)\}$$

Homework 4

Use the `heart` data in the `ncvreg` package. This dataset contains 462 observations on 10 variables. The response variable is `chd`, which indicates whether or not coronary heart disease was present at the time of observation.

(a) Using a package of your choosing, fit Bayesian probit and logistic regression models. Explain which package and priors are used.

(b) Using a method (Laplace and/or MCMC) of your choosing, estimate the marginal likelihood for each model and hence estimate the Bayes factor. Explain how the marginal likelihood is estimated according to the method you have chosen. Give an interpretation of the computed value of the Bayes factor.

(c) Remove one data point from the original dataset. Fit the probit and logistic models again. Provide a HPD interval for the predicted value of the response using the predictor values for the one observation left out from the original dataset. This means you need to simulate from the posterior predictive distribution. Give a 95% HPD interval, as well as an estimate of the posterior predictive mean. Compare this estimate to the actual observed value of that response variable.