Bayesian Models

STAT8810, Fall 2017

M.T. Pratola

October 1, 2017

Today

The Bayesian Approach to Model Building;
Conjugacy

So Far. . .

- Our approach to modeling our "expensive black-box functions"?
  - treat them as a realization from a GP model.
  - Estimate the covariance parameters from the data.
  - Predict and quantify uncertainties using the BLUP.
- The BLUP uses plug-in estimates $\widehat{\boldsymbol{\rho}}$. Does the uncertainty in $\boldsymbol{\rho}$ matter?
- In more complicated models (e.g. treed models) we will have *a lot* more parameters. Does the uncertainty in those parameters matter?
- The Bayesian approach (loosely): treat all parameters as random variables, change maximization problem into integration problem.

Predictive Distribution

- Suppose $\boldsymbol{\rho}, \sigma^2$ are known. In this case, our BLUP was the solution $\hat{y} = c^T \mathbf{y}$ s.t. $\hat{y}$ was unbiased and minimum variance.
  - i.e. an optimization problem.
- Consider the following instead (which we have briefly seen before):

$$\left(\begin{array}{c} y(\mathbf{x}) \\ \hline y(\mathbf{x}_1) \\ \vdots \\ y(\mathbf{x}_n) \end{array} \right) \sim N\left(\left(\begin{array}{c} 0 \\ \mathbf{0}_n \end{array} \right), \left[\begin{array}{cc} 1 & \mathbf{r}^T \\ \mathbf{r} & \mathbf{R} \end{array} \right] \right)$$

- In other words, we *know* that any finite collection of data must have the appropriate Normal distribution under our GP modeling framework.

Predictive Distribution

- But we *observe* $y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n)$! The only thing we don't know is $y(\mathbf{x})$ for some $\mathbf{x} \notin \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.
- *Conditioning* on the quantities *we know*, the resulting conditional Normal distribution is:

$$y(\mathbf{x})|\mathbf{y} \sim N\left(\mathbf{r}^T\mathbf{R}^{-1}\mathbf{y}, \sigma^2(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})\right).$$

- The mean of the conditional Normal is just our BLUP! And it's variance is just the variance of the BLUP!!
- This conditional distribution is called the *predictive distribution*.

Conditional Distributions

- In general, we can write a conditional distribution in many equivalent ways by *Bayes rule*

$$\pi(A|B) = \frac{\pi(A,B)}{\pi(B)} = \frac{\pi(B|A)\pi(A)}{\pi(B)} = \frac{\pi(A|B)\pi(B)}{\pi(B)}$$

$$= \frac{\pi(A,B)}{\int_A \pi(A,B)dA} = \ldots$$

- The normalizing constant in the denominator is the *marginal* distribution, $\pi(B) = \int_A \pi(A,B)dA = \int_A \pi(B|A)\pi(A)dA = \int_A \pi(B|A)d\pi(A)$.
- In general, arriving at closed-form solutions for these distributions is not possible, but some cases are fortunately tractable.

A Brief Aside on Bayes Rule

- Think inversion in the face of uncertainty, or probabilistic inversion.
- That is, if we had the deterministic math problem $y = g(\theta)$ one might solve for θ by

$$\theta = g^{-1}(y).$$

- With uncertainty, the Bayes rule is saying if we have $y = g(\theta) + \epsilon$ then one might "solve" for θ by

$$\pi(\theta|y) = \frac{L(y|\theta)\pi(\theta)}{\pi(y)}.$$

Gaussian Distributions

- For Gaussian distributions, the results are well known. If

$$\begin{pmatrix} \mathbf{y}_a \\ \mathbf{y}_b \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \begin{pmatrix} \Sigma_a & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_b \end{pmatrix}\right)$$

  then the conditional distribution of $\mathbf{y}_a|\mathbf{y}_b$ is

$$\mathbf{y}_a|\mathbf{y}_b \sim N\left(\mu_a + \Sigma_{ab}\Sigma_b^{-1}(\mathbf{y}_b - \mu_b), \Sigma_a - \Sigma_{ab}\Sigma_b^{-1}\Sigma_{ba}\right)$$

  and the marginal distribution is

$$\mathbf{y}_a \sim N\left(\mu_a, \Sigma_a\right)$$

  and similarly for $\mathbf{y}_b|\mathbf{y}_a$ and $\mathbf{y}_b$.

Gaussian Process Interpretation

- Our GP is, of course, nothing but a Normal distribution with a particular covariance function,

  $$\mathbf{z} \sim GP\left(0, c(\mathbf{x}, \mathbf{x}'; \sigma^2, \boldsymbol{\rho})\right)$$

  with corresponding density function f.

- In the Bayesian perspective, this modelling assumption is viewed as a *prior* distribution on the space of functions from which our data arise.
- For example, we can specify that the function space is continuous and differentiable using an Gaussian correlation model, or continuous but nowhere differentiable using an Exponential correlation model, etc.

Gaussian Process Interpretation

- Our observed data, in the emulation context, is observed without error,

$$y(\mathbf{x}_i) = z(\mathbf{x}_i).$$

- As such, the likelihood function for our data is the same GP density function,

$$L(;\mathbf{y}) = f(\mathbf{y}; 0, c(\mathbf{x}, \mathbf{x}'; \sigma^2, \boldsymbol{\rho})).$$

- The predictive distribution for $y(\mathbf{x}) \equiv z(\mathbf{x})$ is simply the conditional GP specified by the corresponding conditional Normal distribution,

$$\pi(z(\mathbf{x})|y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n)) \sim GP(m(\mathbf{x}), C(\mathbf{x}))$$

Example

```r
source("dace.sim.r")
set.seed(88)
n=5
x1=runif(n)
l1=list(m1=abs(outer(x1,x1,"-")))
l.dez=list(l1=l1)
rho=c(0.0001)
s2=1
# sim.field defaults to Gaussian correlation
z=sim.field(l.dez,rho,s2)
```

Example

```
# Now we assume that the z's are observed error-free:
y=z

# Let's write down the predictive distribution
# over a fine grid
ng=100
xg=seq(0,1,length=ng)
X=c(x1,xg)
l1=list(m1=abs(outer(X,X,"-")))
l.dez=list(l1=l1)
Rall=rhogeodacecormat(l.dez,rho)$R
```

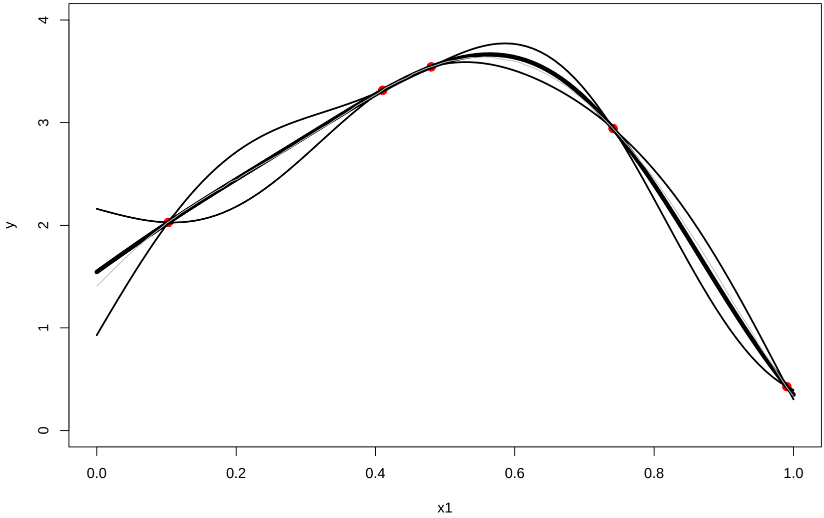
Example

```r
# Extract the sub-matrices we need
Ryy=Rall[1:n,1:n]
Rgg=Rall[(n+1):(n+ng),(n+1):(n+ng)]
Rgy=Rall[(n+1):(n+ng),1:n]
Ryy.inv=chol2inv(chol(Ryy))

# Mean of conditional distribution:
m.cond=Rgy%*%Ryy.inv%*%y

# Covariance of conditional distribution:
E.cond=s2*(Rgg-Rgy%*%Ryy.inv%*%t(Rgy))
```

Example

```
# Now the predictive distn is N(m.cond,E.cond).
# Let's generate a realization!
L=t(chol(E.cond+diag(ng)*1e-5))
u=rnorm(ng)
z.cond=m.cond + L%*%u

# And make a plot
plot(x1,y,pch=20,col="red",cex=2,xlim=c(0,1),ylim=c(0,4))
lines(xg,m.cond,lwd=5,col="black")
lines(xg,m.cond-1.96*sqrt(diag(E.cond)),lwd=2,col="black")
lines(xg,m.cond+1.96*sqrt(diag(E.cond)),lwd=2,col="black")
lines(xg,z.cond,col="grey")
```

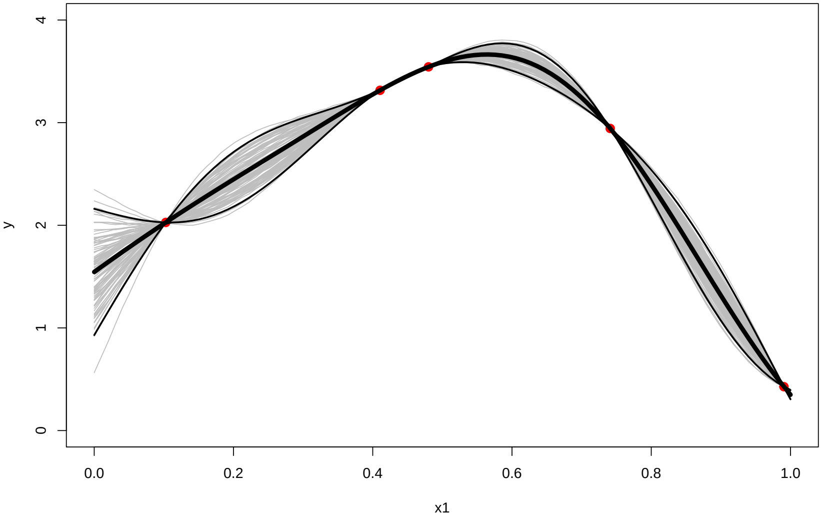
Example



Example

```
# Generate some more realizations
plot(x1,y,pch=20,col="red",cex=2,xlim=c(0,1),ylim=c(0,4))
for(i in 1:100){
  u=rnorm(ng)
  z.cond=m.cond + L%*%u
  lines(xg,z.cond,col="grey")

}
lines(xg,m.cond,lwd=5,col="black")
lines(xg,m.cond-1.96*sqrt(diag(E.cond)),lwd=2,col="black")
lines(xg,m.cond+1.96*sqrt(diag(E.cond)),lwd=2,col="black")
```

Example



What about those parameters?

- We are going to treat those as random variables as well by assigning the parameters *prior distributions*
- Much like our “function-space prior” for the z’s, these distributions are also called priors because they don’t depend on the data.
- Once we observe our data, we will update our information about these parameters using the conditional distribution.
    - Except this conditional distribution is not called the predictive distribution.
    - It is usually called the *posterior distribution*.

The Posterior Distribution

- Let's update our earlier model formulation, but now we will explicitly include the parameters.
- The likelihood function for our data is

$$L(\sigma^2, \boldsymbol{\rho}; \mathbf{y}) = f(\mathbf{y}; 0, c(\mathbf{x}, \mathbf{x}'; \sigma^2, \boldsymbol{\rho})) \equiv f(\mathbf{y}|\sigma^2, \boldsymbol{\rho}).$$

- The prior distributions we will specify are $\pi(\sigma^2)$ and $\pi(\boldsymbol{\rho})$.
- We want to know the updated distribution of the parameters after we observe data. We apply Bayes rule:

$$\begin{aligned} \pi(\sigma^2, \boldsymbol{\rho}|\mathbf{y}) &= \frac{f(\mathbf{y}|\sigma^2, \boldsymbol{\rho})\pi(\sigma^2)\pi(\boldsymbol{\rho})}{\int_{\sigma^2, \boldsymbol{\rho}} f(\mathbf{y}|\sigma^2, \boldsymbol{\rho})\pi(\sigma^2)\pi(\boldsymbol{\rho})d\sigma^2 d\boldsymbol{\rho}} \qquad (1) \\ &\propto f(\mathbf{y}|\sigma^2, \boldsymbol{\rho})\pi(\sigma^2)\pi(\boldsymbol{\rho}) \end{aligned}$$

The Posterior Distribution

- Can we write down the conditional distribution $\pi(\sigma^2, \boldsymbol{\rho}|\mathbf{y})$ like we did for the predictive distribution earlier?
  - Generally no. Closed-forms are only available in very simple cases.
  - Instead, we will have to approximate it numerically.
- And what about the predictions? Previously we wrote

  $$\pi(z(\mathbf{x})|\mathbf{y}), C(\mathbf{x}))$$

  but what we actually had was

  $$\pi(z(\mathbf{x})|\mathbf{y}, \sigma^2, \boldsymbol{\rho}) \sim GP(m(\mathbf{x}), C(\mathbf{x}))$$

- What do we do about the $\sigma^2, \boldsymbol{\rho}$ in this predictive distribution?

The Posterior Predictive Distribution

- If we (somehow) knew (1), then we can *marginalize* our usual predictive distribution *with respect to the posterior*,

$$\pi(z(\mathbf{x})|\mathbf{y}) \quad = \quad \int_{\sigma^2,\boldsymbol{\rho}} \pi(z(\mathbf{x})|\mathbf{y},\sigma^2\boldsymbol{\rho})\pi(\sigma^2,\boldsymbol{\rho}|\mathbf{y})d\sigma^2 d\boldsymbol{\rho}. \quad (2)$$

- In this way we incorporate the uncertainty in the parameters $\sigma^2, \boldsymbol{\rho}$ by marginalizing over the posterior distribution.
    - e.g. this distribution will be more disperse than if we just plugged-in point estimates for $\sigma^2, \boldsymbol{\rho}$ into the usual predictive distribution.
- The resulting distribution (2) is known as the *posterior predictive* distribution.

The Bayesian Conundrum

- So everything depends on somehow getting a handle on the posterior distribution.
- Generally, not easy. In fact, until the 90's, Bayesian methods often avoided because of this problem.
- Three breakthroughs changed things:
  - The Gibbs sampler.
  - The Metropolis-Hastings algorithm.
  - Cheap powerful computers.
- But first, lets look at really simple models we can solve in closed-form.

The Simplest Model Ever

- Consider n noisy independent measurements of a scalar quantity with unknown mean μ and measurement error $e_i \sim N(0, \sigma^2)$ where σ^2 is a known constant.
- The model for the data is

$$y_i = \mu + \epsilon_i, \quad i = 1, \ldots, n$$

- The joint density of the data is

$$f(\mathbf{y}|\mu) = \prod_{i=1}^{n} f(y_i|\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right)$$

The Simplest Model Ever

- Since we have 1 parameter (μ), we will place a prior on it. Suppose the prior distribution is $\mu \sim N(\mu_0, \tau_0^2)$
  - here, μ_0 and τ_0^2 are called *hyperparameters*. We will *not* place priors on these but treat them as known constants that the modeler specifies.
- We want to derive the posterior distribution,

$$\pi(\mu|\mathbf{y}).$$

The Simplest Model Ever

1. Rewrite the model likelihood as

$$
\begin{aligned}
L(\mu|\mathbf{y}) &\equiv f(\mathbf{y}|\mu) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right) \\
&= \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2\right) \\
&= \frac{1}{\sqrt{2\pi}\sigma} exp\left(\frac{-n(\bar{y}-\mu)^2}{2\sigma^2}\right) exp\left(\frac{-(\sum y_i^2 - n\bar{y}^2)}{2\sigma^2}\right)
\end{aligned}
$$

The Simplest Model Ever

2. Write down the posterior up to proportionality as

$$\begin{aligned}\pi(\mu|\mathbf{y}) &\propto L(\mu|\mathbf{y})\pi(\mu)\\ &\propto exp\left(-\frac{1}{2\sigma^2/n}(\bar{y}-\mu)^2\right) exp\left(-\frac{1}{2\tau_0^2}(\mu-\mu_0)^2\right)\end{aligned}$$

3. Rearrange to factor into a term involving μ and everything else

$$\pi(\mu|\mathbf{y}) \propto exp\left(-\frac{1}{2}\left[\frac{(\mu-\mu_1)^2}{\tau_1^2}+\frac{(\bar{y}-\mu_0)^2}{\sigma^2/n+\tau_0^2}\right]\right)$$

where

$$(\tau_1^2)^{-1} = n(\sigma^2)^{-1} + (\tau_0^2)^{-1}$$

and

$$\mu_1 = \tau_1^2(n(\sigma^2)^{-1}\bar{y} + (\tau_0^2)^{-1}\mu_0)$$

The Simplest Model Ever

4. Recognize the term involving only μ as the kernel of the posterior distribution of interest

$$
\begin{aligned}
\pi(\mu|\mathbf{y}) &\propto exp\left(-\frac{1}{2}\left[\frac{\mu-\mu_1)^2}{\tau_1^2}+\frac{(\bar{y}-\mu_0)^2}{\sigma^2/n+\tau_0^2}\right]\right)\\
&\propto exp\left(-\frac{(\mu-\mu_1)^2}{2\tau_1^2}\right)\\
&\Rightarrow \mu|\mathbf{y}\sim N(\mu_1,\tau_1^2)
\end{aligned}
$$

The Simplest Model Ever

- One could work out everything explicitly as a tedious exercise that doesn't matter.
- For Gaussians, the trick is in completing the square.
- Interpretation for this model?
    - precision (reciprocal of variance) of posterior is additive.
    - posterior mean can be written as a weighted combination of the sample mean and prior mean,

$$\begin{aligned}
\mu_1 &= \tau_1^2 \left(\frac{n\bar{y}}{\sigma^2} + \frac{\mu_0}{\tau_0^2} \right) \\
&= \frac{\sigma^2 \tau_0^2}{n\tau_0^2 + \sigma^2} \left(\frac{n\bar{y}}{\sigma^2} + \frac{\mu_0}{\tau_0^2} \right) \\
&= \bar{y} \left(\frac{n\tau_0^2}{n\tau_0^2 + \sigma^2} \right) + \mu_0 \left(\frac{\sigma^2}{n\tau_0^2 + \sigma^2} \right) \\
&= \bar{y}\alpha + \mu_0(1-\alpha), \quad \alpha \in (0,1)
\end{aligned}$$

The Simplest Model Ever

- This provides a nice interpretation, namely that if the prior variance is large (or prior precision is small), little weight is given to the prior mean and more weight is given to the sample mean.
- While a large prior variance (small prior precision) has little effect on the posterior variance (or precision).
- And vice-versa.
- Gives us some ideas of how we might logically specify the prior hyperparameters μ_0, τ_0^2.

Conjugate Distributions

- A family of distributions $\mathcal{P}$ is conjugate to an observational model $\mathcal{F}$ if for every prior $p \in \mathcal{P}$ and for any observational distribution $f \in \mathcal{F}$ the posterior distribution $\pi \in \mathcal{P}$.
- For example, in our simple model we had a (scalar) Normal likelihood and a Normal prior on μ leading to a Normal posterior for μ.
- Advantages of conjugacy is the posterior is always available in closed form.
- Disadvantage may be the restriction to a conjugate form limiting flexibility in the type of prior we can specify.
- Some popular conjugacies for continuous distributions are linked next to this slide set.

Conjugate Distributions

- What about our simple problem – what if σ^2 is also unknown?
- What about GP regression? Then we need a higher dimensional prior involving σ^2 and $\boldsymbol{\rho}$ that is also conjugate to the likelihood?
- Generally conjugacy becomes harder to take advantage of as the dimensionality of model parameters increases.
- However, it turns out that having conjugacy with subsets of parameters is still very useful.

Conditional Conjugacy

- Let us refer to model parameters more generally as $\theta_j, \quad j = 1, \ldots, d$.
- A general form of conditionining is

$$\pi(\theta_i | \theta_j, j \in C) = \frac{\pi(\theta_i, \theta_j, j \in C)}{\pi(\theta_j, j \in C)}$$

  for all $\forall C \subset \{1, \ldots, i-1, i+1, \ldots, d\}$.
- The most important of these is the *full conditional*,

$$\pi(\theta_i | \theta_{-i}).$$

Conditional Conjugacy

- In the case of a parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^T$, a (possibly vector) component $\theta_i, i = 1, \ldots, d$ exhibits conditional conjugacy if the prior $\pi(\theta_i|\theta_{-i})$ (often simply $\pi(\theta_i|\theta_{-i}) \equiv \pi(\theta_i)$ by assuming prior independence) and the full conditional $\pi(\theta_i|\theta_{-i}, \mathbf{y})$ belong to the same family of distributions.

The Simplest Model, Revisited

- To make things easier, I will reparameterize our model in terms of the precision, $\lambda = (\sigma^2)^{-1}$.
- Then, our likelihood was

$$L(\mu, \lambda|\mathbf{y}) = \prod_{i=1}^{n} \frac{\sqrt{\lambda}}{\sqrt{2\pi}} exp\left(-\frac{\lambda}{2}(y_i - \mu)^2\right)$$

- Our prior on μ was

$$\mu \sim N(\mu_0, \lambda_0^{-1})$$

- We will consider λ to also be random now. Let's say our prior is

$$\lambda \sim \text{Gamma}(\alpha, \beta)$$

- Our posterior from earlier now becomes the full-conditional for μ:

$$\mu|\mathbf{y}, \lambda \sim N(\mu_1, \lambda_1^{-1})$$

where $\mu_1 = \lambda_1^{-1}(n\lambda\bar{y} + \lambda_0\mu_0)$ and $\lambda_1 = n\lambda + \lambda_0$.

The Simplest Model, Revisited

- Next, we need the full conditional for λ.

5. The form of the prior is

$$\pi(\lambda) \propto \lambda^{\alpha-1} exp\left(-\lambda\beta\right)$$

6. Write down the full conditional up to proportionality

$$\pi(\lambda|\mu, \mathbf{y}) \quad \propto \quad \lambda^{n/2} exp\left(-\frac{\lambda}{2}\sum_{i=1}^{n}(y_i-\mu)^2\right)\lambda^{\alpha-1} exp\left(-\lambda\beta\right)$$

7. Rearrange into a factor involve λ and everything else

$$\pi(\lambda|\mu, \mathbf{y}) \quad \propto \quad \lambda^{\alpha+n/2-1} exp\left(-\lambda\left[\frac{1}{2}\sum_{i=1}^{n}(y_i-\mu)^2+\beta\right]\right)$$

The Simplest Model, Revisited

8. Recognize the term involving only λ as the kernel of the posterior distribution of interest

$$\begin{aligned}\pi(\lambda|\mu,\mathbf{y}) &\propto \lambda^{\alpha+n/2-1} exp\left(-\lambda\left[\frac{1}{2}\sum_{i=1}^{n}(y_i-\mu)^2+\beta\right]\right)\\ &\Rightarrow \lambda|\mu,\mathbf{y}\sim \text{Gamma}(\alpha_n,\beta_n)\end{aligned}$$

where $\alpha_n = \alpha + \frac{n}{2}$ and $\beta_n = \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i-\mu)^2$.

The Gibbs Sampler†

- So we have $\pi(\mu|\lambda, \mathbf{y})$ and $\pi(\lambda|\mu, \mathbf{y})$ in closed form. Is this the same as the posterior $\pi(\mu, \lambda|\mathbf{y})$?
- No. But it turns out we can use our full conditionals to get approximate samples from the true posterior by using an algorithm called the Gibbs Sampler†.

† Geman and Geman: *Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.6, pp.721–741 (1984).

Gelfand and Smith: *Sampling-based approaches to calculating marginal densities*, Journal of the American Statistical Association, vol.85, pp.398–409 (1990).

The Gibbs Sampler

- Allows us to simulate draws from the joint posterior $\pi(\theta_1, \ldots, \theta_d | \mathbf{y})$ when its only possible to directly simulate from full conditional distributions, $\pi(\theta_i | \theta_{-i}, \mathbf{y})$.

1. Initialize counter $j = 1$ and $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_d^{(0)})$
2. Successively obtain new values $\boldsymbol{\theta}^{(j)} = (\theta_1^{(j)}, \ldots, \theta_d^{(j)})$ via the full conditionals as:

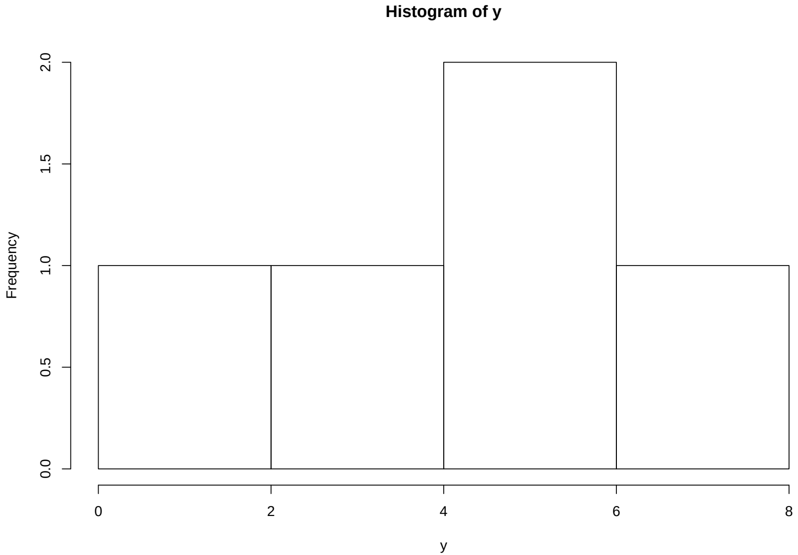
$$\begin{aligned} \theta_1^{(j)} &\sim \pi(\theta_1 | \theta_2^{(j-1)}, \ldots, \theta_d^{(j-1)}) \\ \theta_2^{(j)} &\sim \pi(\theta_2 | \theta_1^{(j)}, \theta_3^{(j-1)}, \ldots, \theta_d^{(j-1)}) \\ &\vdots \\ \theta_d^{(j)} &\sim \pi(\theta_d | \theta_1^{(j)}, \ldots, \theta_{d-1}^{(j)}) \end{aligned}$$

3. Increment j to $j + 1$ and return to step (2) until convergence.

Example: λ known

```
# Simple example with known precision, unknown mean.
set.seed(88) # just to replicate this example
lambda=0.5
mu=3.5
n=5
y=rnorm(n,mean=mu,sd=sqrt(1/lambda))
hist(y)
```

Example: λ known

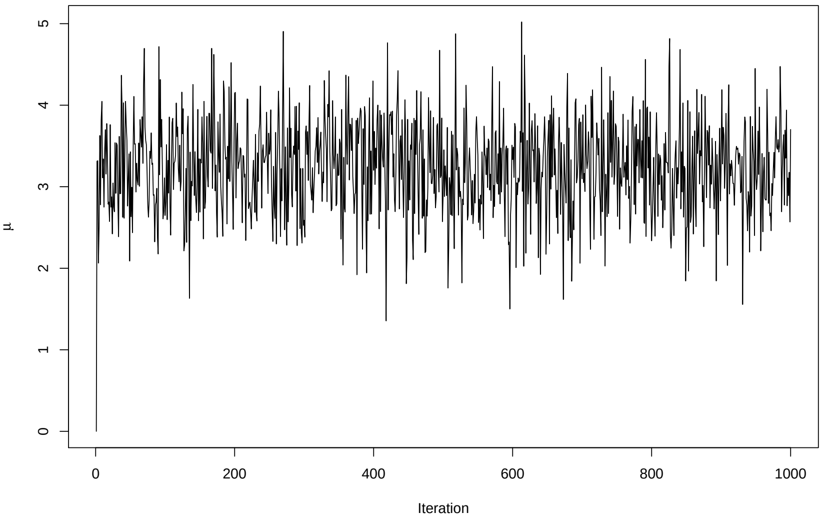


Example: λ known

```
N=1000
ybar=mean(y)
mu0=0
lambda0=0.5
draw.mu=rep(0,N)
lambda1=n*lambda+lambda0
mu1=1/lambda1*(n*lambda*ybar+lambda0*mu0)
for(i in 2:N) {
  draw.mu[i]=rnorm(1,mean=mu1,sd=sqrt(1/lambda1))
}
```

Example: λ known

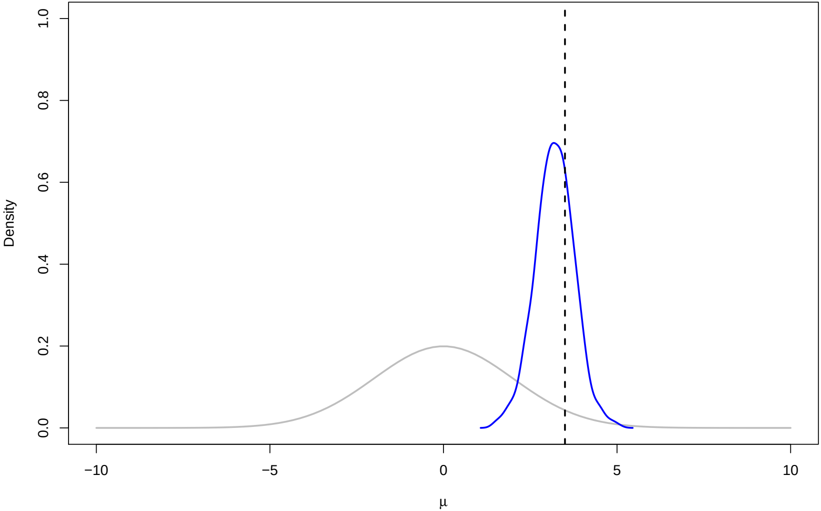
```r
plot(draw.mu,type='l',xlab="Iteration",ylab=expression(mu))
```



Example: λ known

```r
# Drop "burn-in"
draw.mu=draw.mu[(N/2):N]

# Plot the prior, posterior and the truth
x=seq(-10,10,length=100)
dens.prior=dnorm(x,mean=mu0,sd=1/lambda0)
plot(x,dens.prior,type='l',col="grey",lwd=2,xlim=c(-10,10),
lines(density(draw.mu),lwd=2,col="blue")
abline(v=mu,lty=2,lwd=2)
```

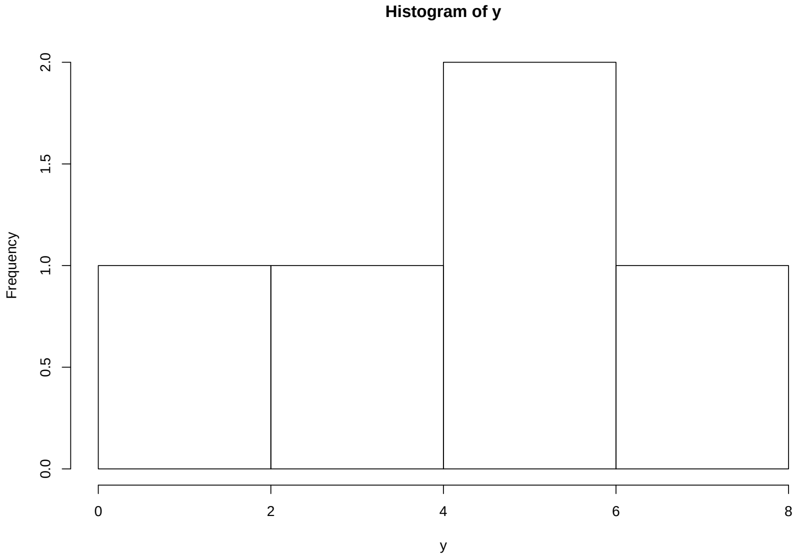
Example: λ known



Example: λ unknown

```
# Simple example with unknown precision, unknown mean.
set.seed(88) # just to replicate this example
lambda=0.5
mu=3.5
n=5
y=rnorm(n,mean=mu,sd=sqrt(1/lambda))
hist(y,xlim=c(0,7))
```

Example: λ unknown



Example: λ unknown

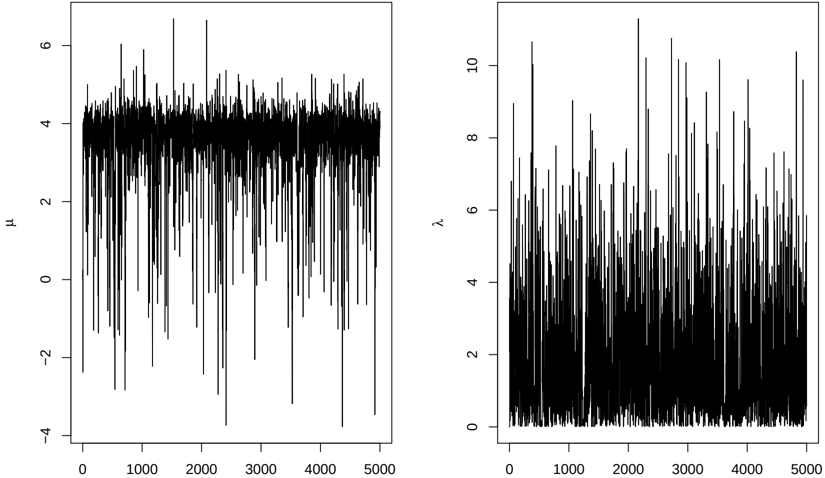
```
N=5000
ybar=mean(y)
mu0=0
lambda0=0.25
alpha=1
beta=1
draw.mu=rep(0,N)
draw.lambda=rep(0,N)
```

Example: λ unknown

```
for(i in 2:N) {
  # Gibbs step for mu
  lambda1=n*draw.lambda[i-1]+lambda0
  mu1=1/lambda1*(n*draw.lambda[i-1]*ybar+lambda0*mu0)
  draw.mu[i]=rnorm(1,mean=mu1,sd=sqrt(1/lambda1))

  # Gibbs step for lambda
  alpha.n=alpha+n/2
  beta.n=beta+0.5*sum(y-draw.mu[i])^2
  draw.lambda[i]=rgamma(1,shape=alpha.n,rate=beta.n)
}
```

Example: λ unknown

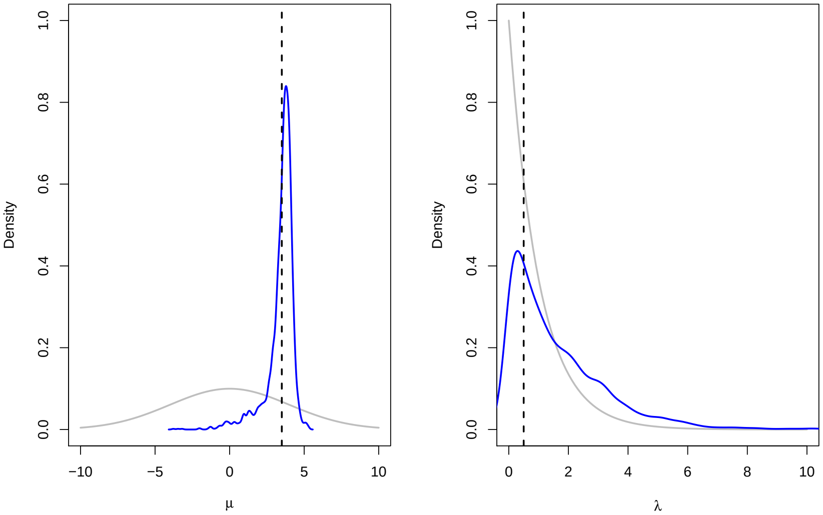
```
par(mfrow=c(1,2))
plot(draw.mu,type='l',xlab="Iteration",ylab=expression(mu)
plot(draw.lambda,type='l',xlab="Iteration",ylab=expression
```



Example: λ unknown

```r
# Drop "burn-in"
draw.mu=draw.mu[(N/2):N]
draw.lambda=draw.lambda[(N/2):N]

# Plot the prior, posterior and the truth
x=seq(-10,10,length=100)
dens.prior=dnorm(x,mean=mu0,sd=1/lambda0)
plot(x,dens.prior,type='l',col="grey",lwd=2,xlim=c(-10,10),
lines(density(draw.mu),lwd=2,col="blue")
abline(v=mu,lty=2,lwd=2)
x=seq(0,10,length=100)
dens.prior=dgamma(x,shape=alpha,rate=beta)
plot(x,dens.prior,type='l',col="grey",lwd=2,xlim=c(0,10),y
lines(density(draw.lambda),lwd=2,col="blue")
abline(v=lambda,lty=2,lwd=2)
```

Example: λ unknown



The Gibbs Sampler

- Under mild conditions, the draws $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(n)}$ form an ergodic sequence of random variables (a Markov Chain) with stationary distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$.
- In practice, we need to allow the chain to "burn-in" for *m* iterations and then take an additional *n* samples as draws from the posterior distribution.
- In some situations, even full conditionals are not available in closed form. In this case we can use another algorithm to get approximate samples from the full conditionals - the Metropolis-Hastings algorithm.

Metropolis-Hastings (MH) Algorithm

- MH defines the *transition density*, $p(\theta, \theta')$, of a Markov Chain such that the posterior of interest, $\pi(\theta|\mathbf{y})$ is the stationary distribution of the chain.
- First, define a *proposal distribution*, $q(\theta'; \theta)$. This is also called the transition kernel, since it defines the proposed probability from being in state θ and moving to state θ': $q(\theta'; \theta) \equiv q(\theta \to \theta')$.
- Next, define the acceptance probability of a proposed state θ' as
$$\alpha = min\left\{1, \frac{\pi(\theta'|\mathbf{y})q(\theta' \to \theta)}{\pi(\theta|\mathbf{y})q(\theta \to \theta')}\right\}$$
- Accept θ' as a draw from $\pi(\theta|\mathbf{y})$ with probability α, otherwise reject.
- Following this procedure for a sequence of proposed θ's simulates draws from the posterior distribution of interest.

Metropolis-Hastings (MH) Algorithm

- Selecting $q()$ is important - it is a flexible tool to help construct an efficient sampling algorithm.
- Roberts and Smith† show that if $q()$ is irreducible and aperiodic and $\alpha > 0$ for every possible value of (θ, θ') then the algorithm defines an irreducible and aperiodic Markove Chain with limiting distribution $\pi(\theta|\mathbf{y})$
  - In particular, it is ergodic which means we can estimate quantites of interest from a single realization of the sample path – recall we saw this earlier!

† G.O. Roberts and A.F.M. Smith: *Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms*, Stochastic processes and their applications, vol.49, pp.207–216 (1994).

Metropolis-Hastings (MH) Algorithm

- If $q()$ is symmetric, then $q(\theta;\theta') = q(\theta';\theta)$ for every (θ,θ'), so α reduces to
$$\alpha = min\left\{1, \frac{\pi(\theta'|\mathbf{y})}{\pi(\theta|\mathbf{y})}\right\}$$
- Since the normalizing constant for $\pi(\theta|\mathbf{y})$ is the same for both numerator and denominator, to calculate α we only require $L(\theta|\mathbf{y})$ and $\pi(\theta)$.
- We can incorporate *MH* steps to generate approximate samples from the full conditionals so that we can generate samples from the joint posterior using a Gibbs-like algorithm.

Metropolis-Hastings (MH) Algorithm

So our final algorithm is:

1. Initialize $j = 1$ and set arbitrary initial value for $\boldsymbol{\theta}^{(0)}$.
2. Set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$.
3. For $k = 1, \ldots, d$

   3.1 Draw proposal $\theta'_k \sim q_k(\theta'_k; \theta_k^{(j)})$

   3.2 Calculate

   $$\alpha = min\left\{1, \frac{L(\theta'_k, \boldsymbol{\theta}_{-k}^{(j)}|\mathbf{y})\pi(\theta'_k, \boldsymbol{\theta}_{-k}^{(j)})q(\theta'_k; \theta_k^{(j)})}{L(\boldsymbol{\theta}^{(j)}|\mathbf{y})\pi(\boldsymbol{\theta}^{(j)})q(\theta_k^{(j)}; \theta'_k)}\right\}$$

   3.3 Generage $u \sim \text{Unif}(0, 1)$. If $u \leq \alpha$ then accept move to $\theta_k^{(j)} = \theta'_k$; otherwise reject move and keep $\theta_k^{(j)} = \theta_k^{(j-1)}$.
4. Increment j to $j + 1$ and return to step (2) until convergence.

Metropolis-Hastings (MH) Algorithm

- Success of MH depends on maintaining a reasonably high acceptance rate (α), which depends on a good proposal distribution q.
- For continuous scalar parameters, a good acceptance rate is $\sim 44\%$. For higher-dimensional parameters, a good acceptance rate is smaller. Generally we are reasonably happy if $23\% < \alpha < 49\%$.
- Random-walk proposal distribution, q:

  $$\theta' = \theta + \omega, \quad \omega \sim N(0, \sigma_q^2)$$

  where σ_q^2 is a user-specified constant chosen to tune α to a desired rate.
- Uniform proposal distribution, q:

  $$\theta' = \theta + u, \quad u \sim Unif(-a_q, a_q)$$

  where a_q is a user-specified constant chosen to tune α to a desired rate.

Metropolis-within-Gibbs

- If we have full conditionals in closed-form for $\theta_1, \dots, \theta_l$ then we can combine Gibbs steps for these with MH steps for $\theta_{l+1}, \dots, \theta_d$:

1. Initialize $j = 1$ and set arbitrary initial value for $\boldsymbol{\theta}^{(0)}$.
2. Set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$.
3. Sample $\theta_1, \dots, \theta_l$ using the Gibbs algorithm
4. Sample $\theta_{l+1}, \dots, \theta_d$ using the Metropolis-Hastings algorithm (i.e. steps 3.1-3.3).
5. Increment j to $j + 1$ and return to step (2) until convergence.