Chapter 8: Markov Chain Monte Carlo 2

ST4234: Bayesian Statistics

Semester 2, AY 2019/2020

Department of Statistics and Applied Probability

National University of Singapore

LI Cheng

stalic@nus.edu.sg

Introduction

- In Chapter 7, we have introduced the Metropolis-Hastings algorithm, which is a generic Markov chain Monte Carlo algorithm that can be applied to almost any Bayesian models.

- In this chapter, we introduce the Gibbs sampler (or sometimes called Gibbs sampling). Gibbs sampler is more convenient to implement if the posterior distribution satisfies certain structure. The Gibbs sampler is very useful when the parameter is of high dimensions.

- We will then introduce some composite sampling methods using both MH algorithm and Gibbs sampler.

Outline

Gibbs Sampler

Example 1: Normal Model

Example 2: Student's t Model

8.1 Gibbs Sampler

- A Markov chain algorithm that has been found useful in many high-dimensional problems is the Gibbs sampler.
- Suppose that we partition the parameter vector of interest into p components or subvectors, $\theta = (\theta_1, \ldots, \theta_p)$.
- Let us consider the set of conditional distributions:

$$
\begin{gathered}
p(\theta_1|\theta_2, \ldots, \theta_p, \boldsymbol{y}) \\
p(\theta_2|\theta_1, \theta_3, \ldots, \theta_p, \boldsymbol{y}), \\
\vdots \\
p(\theta_p|\theta_1, \ldots, \theta_{p-1}, \boldsymbol{y}),
\end{gathered}
$$

where each component is conditional on the data and all other components.

8.1 Gibbs Sampler

- The idea behind Gibbs sampler is that we can set up a Markov chain simulation algorithm from the joint posterior distribution by sampling in turn from these p conditional posterior distributions.

- At each iteration, an ordering of the p components of θ is chosen and each is sampled in turn from the conditional distribution given the current value of all other components of θ. Thus, there are p steps in each iteration.

- Under general regularity conditions, draws from this simulation algorithm will converge to the target joint posterior distribution $p(\theta|\boldsymbol{y})$ of interest.

8.1 Gibbs Sampler

- Suppose we use the ordering $\theta = (\theta_1, \ldots, \theta_p)$. In other words, $\theta_1, \ldots, \theta_p$ are p components or subvectors of the parameter vector θ. Then the Gibbs sampler algorithm proceeds as follows.

Gibbs sampler

1. Let $\{\theta_1^{(0)}, \ldots, \theta_p^{(0)}\}$ denote an arbitrary set of starting values.
2. For $t = 1, 2, \ldots, T$,
    - Draw $\theta_1^{(t)} \sim p(\theta_1 | \theta_2^{(t-1)}, \ldots, \theta_p^{(t-1)}, \boldsymbol{y})$,
    - Draw $\theta_2^{(t)} \sim p(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_p^{(t-1)}, \boldsymbol{y})$,

      $\vdots$
    - Draw $\theta_p^{(t)} \sim p(\theta_p | \theta_1^{(t)}, \ldots, \theta_{p-1}^{(t)}, \boldsymbol{y})$.
3. Output $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(T)}$.

- Thus, each component θ_j is updated conditional on the latest values of all other components of θ.

8.1 Gibbs Sampler

- For a large group of inference problems, Gibbs sampler is very attractive in that all conditional posterior distributions are available or easy to simulate using standard probability distributions.
- Unlike the more general Metropolis-Hastings algorithms, there are no tuning constants or proposal densities to define in Gibbs sampler.
- In fact, it can be shown that Gibbs sampler is a special case of Metropolis-Hastings algorithm, with acceptance probability always equal to 1.

Outline

Gibbs Sampler

Example 1: Normal Model

Example 2: Student's t Model

8.2 Example 1: Normal Model

- Suppose that $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ are i.i.d. from $\mathrm{N}(\mu, \sigma^2)$.
- The prior on $\theta = (\mu, \sigma^2)$ is

$$\mu|\sigma^2 \sim \mathrm{N}\left(\mu_0, \frac{\sigma^2}{n_0}\right), \quad \sigma^2 \sim \text{Inv-Gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right).$$

- The posterior density is (see Chapter 3 notes page 30):

$$p(\mu, \sigma^2|\boldsymbol{y}) \propto (\sigma^2)^{-1/2} \exp\left\{-\frac{(\mu - \frac{n\bar{y}+n_0\mu_0}{n+n_0})^2}{2\sigma^2/(n+n_0)}\right\}$$
$$\times (\sigma^2)^{-\frac{\nu_0+n}{2}-1} \exp\left\{-\frac{(n-1)s^2 + \nu_0\sigma_0^2 + \frac{nn_0(\bar{y}-\mu_0)^2}{n+n_0}}{2\sigma^2}\right\}.$$

- Here $\bar{y} = n^{-1}\sum_{i=1}^{n} y_i$, $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$.

8.2 Example 1: Normal Model

- From Chapter 3, we know that this posterior is the so-called normal-inverse gamma distribution. We can use Gibbs sampler to draw samples of (μ, σ^2) from this posterior.

- We can derive that

$$\mu \mid \sigma^2, \boldsymbol{y} \sim \mathrm{N}\left(\frac{n\bar{y} + n_0\mu_0}{n + n_0}, \frac{\sigma^2}{n + n_0}\right),$$

$$\sigma^2 \mid \mu, \boldsymbol{y} \sim \text{Inv-Gamma}\left(\frac{\nu_0 + n + 1}{2}, \frac{S}{2}\right),$$

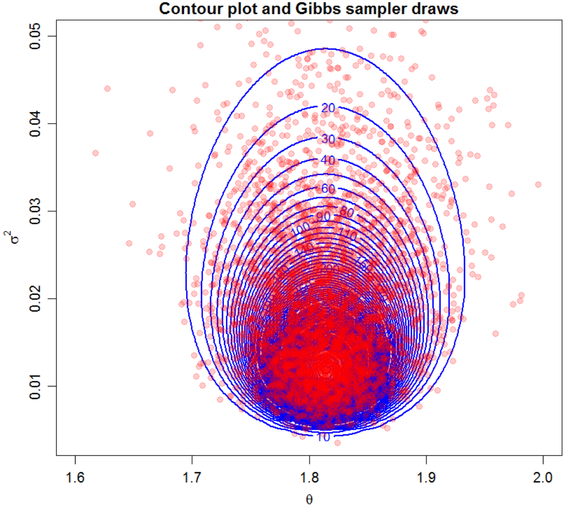
$$\text{where } S = (n + n_0)\left(\mu - \frac{n\bar{y} + n_0\mu_0}{n + n_0}\right)^2 + (n-1)s^2 + \nu_0\sigma_0^2 + \frac{nn_0(\bar{y} - \mu_0)^2}{n + n_0}.$$

- The Gibbs sampler alternates between the sampling of μ and σ^2 from these two conditional distributions, using the parameter values from the previous iteration.

- We can compare the posterior means and standard deviations of μ and σ^2 computed from Gibbs sampler, with the "true" values obtained from the direct Monte Carlo simulation using normal-inverse gamma posterior derived in Chapter 3. The R codes of Gibbs sampler are in the file "Chapter8_Rcodes.r".

	Posterior Mean		Posterior SDs	
	Gibbs	direct MC	Gibbs	direct MC
μ	1.8137	1.8129	0.0436	0.0433
σ^2	0.0193	0.0191	0.0111	0.0108

- The draws from Gibbs sampler are plotted in the contour plot of the posterior $p(\mu, \sigma^2|\boldsymbol{y})$.



Outline

Gibbs Sampler

Example 1: Normal Model

Example 2: Student's t Model

8.3 Example 2: Student's t Model

- When modeling data with a possibility of outliers, a good strategy is to assume the observations are distributed from a population with tails that are heavier than the normal form.

- An example of a heavy-tailed distribution is the t family with a small number of degrees of freedom. Suppose that $\boldsymbol{y} = (y_1, \ldots, y_n)$ are a sample from a t distribution with location μ, scale σ^2 and known degrees of freedom ν.

- If we assign a noninformative prior on (μ, σ), $p(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$, the posterior density is given by

$$p(\mu, \sigma^2 | \boldsymbol{y}) \propto \frac{1}{\sigma^2} \prod_{i=1}^{n} \left[\frac{1}{\sigma} \left(1 + \frac{(y_i - \mu)^2}{\nu \sigma^2} \right)^{-(\nu+1)/2} \right].$$

- If we use the following mixture representation and introduce a scale parameter λ_i for each y_i, then we can write our model as

$$\begin{aligned} y_i|\lambda_i, \mu, \sigma^2 &\sim \mathrm{N}\left(\mu, \frac{\sigma^2}{\lambda_i}\right), \qquad i = 1, \dots, n, \\ \lambda_i &\sim \mathrm{Gamma}(\nu/2, \nu/2), \quad i = 1, \dots, n, \\ (\mu, \sigma^2) &\sim p(\mu, \sigma^2) \propto \frac{1}{\sigma^2}. \end{aligned}$$

- Derivation on board: If y_i is modeled as above, then $y_i|\mu, \sigma^2$ follows the t_ν distribution with location μ and scale σ^2.

- Idea: Heavy-tailed distributions such as t distributions can be viewed as an (infinite) mixture of thin-tailed distributions such as normal.

- Now we introduce the additional parameters $\lambda = (\lambda_1, \ldots, \lambda_n)$ into the model. They are called **latent variables**. They can be viewed as either parameters, or missing data.

- Such an approach of deliberately introducing additional parameters into the model is usually called **data augmentation**. The purpose is to make the posterior distribution more tractable.

- For example, t distributions do not belong to the exponential family distribution and do not have nice conjugate priors. But normal distributions and gamma distributions are both exponential family distributions, and they allow convenient specification of conjugate priors. This makes the application of Gibbs sampler possible.

- The joint posterior of $\theta = (\mu, \sigma^2, \lambda)$ is

$$\begin{aligned}
p(\mu, \sigma^2, \lambda | \boldsymbol{y}) &\propto p(\boldsymbol{y}|\mu, \sigma^2, \lambda) p(\mu, \sigma^2, \lambda) \\
&= \prod_{i=1}^{n} [p(y_i|\mu, \sigma^2, \lambda_i) p(\lambda_i)] p(\mu, \sigma^2) \\
&\propto \prod_{i=1}^{n} \left[\frac{\lambda_i}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{\lambda_i}{2} \left(\frac{(y_i - \mu)^2}{\sigma^2} \right) \right\} \cdot \lambda_i^{\frac{\nu}{2}-1} e^{-\frac{\nu\lambda_i}{2}} \right] \cdot \frac{1}{\sigma^2} \\
&\propto \prod_{i=1}^{n} \left[\frac{1}{\sqrt{\sigma^2}} \lambda_i^{\frac{\nu+1}{2}-1} \exp\left\{ -\frac{\lambda_i}{2} \left[\frac{(y_i - \mu)^2}{\sigma^2} + \nu \right] \right\} \right] \cdot \frac{1}{\sigma^2} \\
&= (\sigma^2)^{-\frac{n}{2}-1} \prod_{i=1}^{n} \left[\lambda_i^{\frac{\nu+1}{2}-1} \exp\left\{ -\frac{\lambda_i}{2} \left[\frac{(y_i - \mu)^2}{\sigma^2} + \nu \right] \right\} \right]
\end{aligned}$$

- To implement Gibbs sampler, we need to find the full conditional posteriors for each of $\mu, \sigma^2, \lambda_1, \ldots, \lambda_n$:

$$\begin{aligned}&p(\mu|\sigma^2, \lambda, \boldsymbol{y}),\\&p(\sigma^2|\mu, \lambda, \boldsymbol{y}),\\&p(\lambda_i|\mu, \sigma^2, \lambda_{-i}, \boldsymbol{y}) \text{ for } \ i = 1, \ldots, n,\end{aligned}$$

  where $\lambda_{-i} = (\lambda_1, \ldots, \lambda_{i-1}, \lambda_{i+1}, \ldots, \lambda_n)$, i.e. the vector λ with the ith component, λ_i, excluded.

- We will see that all these conditional distributions have simple forms, because we have introduced the additional latent variables in λ.

- The conditional posterior of σ^2 is

$$p(\sigma^2|\mu,\lambda,\boldsymbol{y}) = \frac{p(\mu,\sigma^2,\lambda|\boldsymbol{y})}{p(\mu,\lambda|\boldsymbol{y})} \overset{\text{why?}}{\propto} p(\mu,\sigma^2,\lambda|\boldsymbol{y}) \propto (\sigma^2)^{-n/2-1}\exp\left\{-\frac{\sum_{i=1}^n \lambda_i (y_i-\mu)^2}{2\sigma^2}\right\}$$

- Therefore,

$$\sigma^2|\mu,\lambda,\boldsymbol{y} \sim \text{Inv-Gamma}\left(\frac{n}{2},\ \frac{\sum_{i=1}^n \lambda_i (y_i-\mu)^2}{2}\right).$$

- General Idea: To find the conditional posterior of a parameter (e.g. σ^2 here) given all the other parameters (μ,λ), we only need to extract all the parts that involve this parameter (σ^2) from the full joint posterior, treat all the other parameters (μ,λ) as fixed constants, simplify the formula, and identify it as a familiar distribution.

8.3 Example 2: Student's t Model

Find conditional posteriors

- The conditional posterior of μ is

$$
\begin{aligned}
p(\mu|\sigma^2, \lambda, \boldsymbol{y}) &\propto p(\mu, \sigma^2, \lambda|\boldsymbol{y}) \\
&\propto \prod_{i=1}^{n} \left[\exp\left\{ -\frac{\lambda_i}{2} \frac{(y_i-\mu)^2}{\sigma^2} \right\} \right] \\
&\propto \exp\left[\sum_{i=1}^{n} -\left\{ \frac{\mu^2\lambda_i - 2\mu y_i \lambda_i}{2\sigma^2} \right\} \right] \\
&= \exp\left[-\frac{\mu^2 \sum_{i=1}^{n} \lambda_i - 2\mu \sum_{i=1}^{n} y_i\lambda_i}{2\sigma^2} \right] \\
&\propto \exp\left[-\frac{\left(\mu - \frac{\sum_{i=1}^{n} y_i\lambda_i}{\sum_{i=1}^{n} \lambda_i} \right)^2}{2\frac{\sigma^2}{\sum_{i=1}^{n} \lambda_i}} \right].
\end{aligned}
$$

- Therefore,

$$
\mu|\sigma^2, \lambda, \boldsymbol{y} \sim \mathrm{N}\left(\frac{\sum_{i=1}^{n} y_i\lambda_i}{\sum_{i=1}^{n} \lambda_i}, \frac{\sigma^2}{\sum_{i=1}^{n} \lambda_i} \right).
$$

- For $i = 1, \ldots, n$,

$$
\begin{aligned}
p(\lambda_i|\mu, \sigma^2, \lambda_{-i}, \boldsymbol{y}) &\propto p(\mu, \sigma^2, \lambda_i, \lambda_{-i}|\boldsymbol{y}) \\
&\propto \prod_{i=1}^{n} \left[\lambda_i^{(\nu+1)/2-1} \exp\left\{ -\frac{\lambda_i}{2} \left[\frac{(y_i-\mu)^2}{\sigma^2} + \nu \right] \right\} \right] \\
&\propto \lambda_i^{(\nu+1)/2-1} \exp\left\{ -\frac{\lambda_i}{2} \left(\frac{(y_i-\mu)^2}{\sigma^2} + \nu \right) \right\}
\end{aligned}
$$

- We can see that $\lambda_1, \ldots, \lambda_n$ are conditionally independent given μ, σ^2 and $\boldsymbol{y}$, and

$$
\lambda_i|\mu, \sigma^2, \boldsymbol{y} \sim \text{Gamma}\left(\frac{\nu+1}{2}, \frac{(y_i-\mu)^2}{2\sigma^2} + \frac{\nu}{2} \right).
$$

8.3 Example 2: Student's t Model

- The Gibbs sampling algorithm can thus proceed as follows.

Gibbs sampler

1. Let $\{\mu^{(0)}, \sigma^{2(0)}\}$ be the starting values.
2. For $t = 1, 2, \ldots, T$:
   - Draw $\lambda_i^{(t)} \sim \text{Gamma}\left(\frac{\nu+1}{2}, \frac{(y_i - \mu^{(t-1)})^2}{2\sigma^{2(t-1)}} + \frac{\nu}{2}\right)$ for $i = 1, \ldots, n$,
   - Draw $\mu^{(t)} \sim \text{N}\left(\frac{\sum_{i=1}^n y_i \lambda_i^{(t)}}{\sum_{i=1}^n \lambda_i^{(t)}}, \frac{\sigma^{2(t-1)}}{\sum_{i=1}^n \lambda_i^{(t)}}\right)$,
   - Draw $\sigma^{2(t)} \sim \text{Inv-Gamma}\left(\frac{n}{2}, \frac{\sum_{i=1}^n \lambda_i^{(t)}(y_i - \mu^{(t)})^2}{2}\right)$.

- We do not need to initialize $\lambda_1, \ldots, \lambda_n$, since they are the first to be sampled in Gibbs sampler.

- We apply this Student's t model to Darwin's famous dataset concerning 15 differences of the heights of cross- and self-fertilized plants quoted by Fisher (1960). This dataset can be found in the `LearnBayes` library with the name `darwin`.
- We set the number of degrees of freedom $\nu = 4$ and run the Gibbs sampling algorithm for 10^4 iterations.

```
require(invgamma)
require(LearnBayes)     # load dataset from LearnBayes
data(darwin)
y <- darwin$difference  # observations
n <- length(y)
nu <- 4                 # degrees of freedom
T <- 10^4              # no. of iterations in Gibbs sampling
```

- The Gibbs sampling algorithm can be implemented using the R-code below.
- The arguments of `Gibbs_t` are the data vector `y`, the degrees of freedom `nu`, and the number of iterations of the Gibbs sampler `T`.

```
Gibbs_t <- function(y, nu, T){
    n <- length(y)                    # no. of observations
    lambda_draws <- matrix(0,T,n) # save draws of lambda
    mu_draws <- rep(0,T)              # save draws of mu
    sigma2_draws <- rep(0,T)          # save draws of sigma2
    mu <- 0                           # initial value of mu
    sigma2 <- 1                       # initial value of sigma2
```

(codes continue on next page...)

```
 for (t in 1:T){
  # random generation
  lambda <- rgamma(n, shape=(nu+1)/2,
                   rate=(y-mu)^2/(2*sigma2) + nu/2)
  mu <- rnorm(1, mean=sum(lambda*y)/sum(lambda),
              sd=sqrt(sigma2/sum(lambda)))
  sigma2 <- rigamma(1, n/2, sum(lambda*(y-mu)^2)/2)

  # save values of parameters at t iteration
  lambda_draws[t,] <- lambda
  mu_draws[t] <- mu
  sigma2_draws[t] <- sigma2
 }
 list(lambda_draws=lambda_draws, mu_draws=mu_draws,
     sigma2_draws=sigma2_draws)
}
```

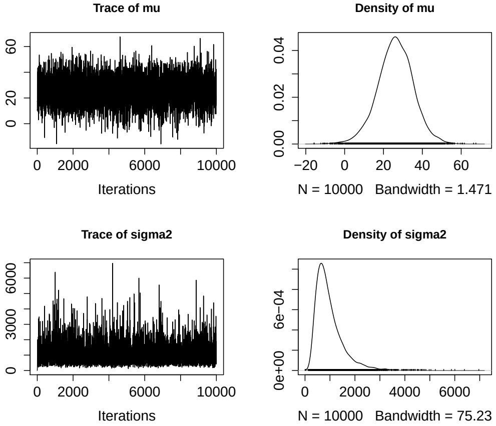
8.3 Example 2: Student's t Model

Darwin's plant data

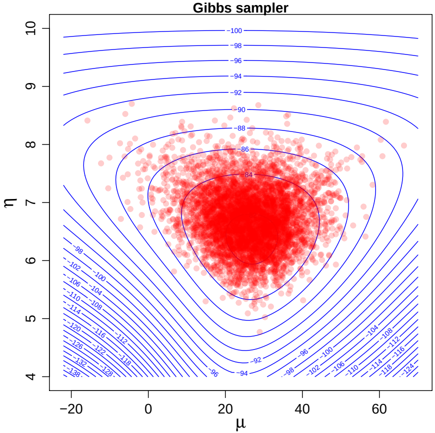
- Note that we are using `rgamma` with a vector rate parameter. Due to the conditional independence property, $\lambda_1, \ldots, \lambda_n$, can be simulated simultaneously.
- The output is a list with three components: `mu_draws` is a vector of simulated draws of μ, `sigma2_draws` is a vector of simulated draws of σ^2, and `lambda_draws` is a matrix of simulated draws of λ, where each row corresponds to a single draw.
- In the R-code below, we run the Gibbs sampling algorithm.

```
set.seed(1)
fit <- Gibbs_t(y, nu, T)
require(coda)
mcmc.gibbs <- mcmc(cbind(fit$mu_draws,fit$sigma2_draws))
colnames(mcm.gibbs) <- c("mu","sigma2")
plot(mcmc.gibbs)
```

- Then we use the `coda` package to obtain trace plots and density estimates from the simulated draws of μ and σ^2. These are shown in the figure below.



- We can also plot the contour plot of the log posterior, and overlay the posterior samples drawn from Gibbs sampler. We plot the parameters (μ, η), where $\eta = \log(\sigma^2)$. Later we will apply the MH algorithm to this model and do some comparison.

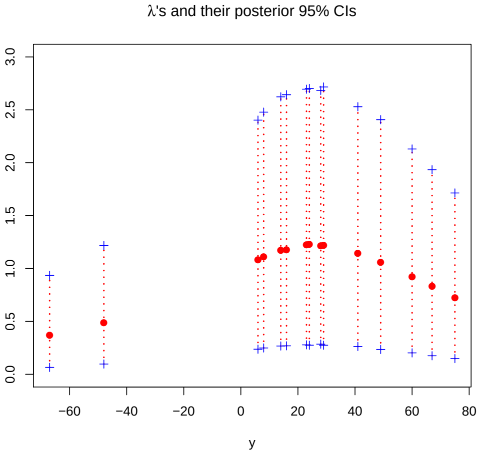


- Note that each λ_i represents the weight of the observation y_i in the estimation of the location and scale of the t population.
- In the R-code below, we compute the posterior mean and the 5th and 95th percentiles of each λ_i. Then we plot the posterior means of the $\{\lambda_i\}$ against the observations $\{y_i\}$, and overlay lines that represent 90% confidence intervals for $\{\lambda_i\}$.

```
plot(y,post.mean.lambda,ylab="",ylim=c(0,3),
     pch=16,cex=1.2,col="red",
     main=expression(paste(lambda,
     "'s and their posterior 95% CIs")))
for (i in 1:n){
  lines(rep(y[i],2), CI.lambda[,i],lwd=2,lty=3,col="red")
  points(y[i],CI.lambda[1,i],pch=3,col="blue")
  points(y[i],CI.lambda[2,i],pch=3,col="blue")
}
```

8.3 Example 2: Student's t Model

- The location of the posterior density of λ_i tends to be small for outlying observations; these particular observations are downweighted in the estimation of the location and scale parameters.



8.3 Example 2: Student's t Model

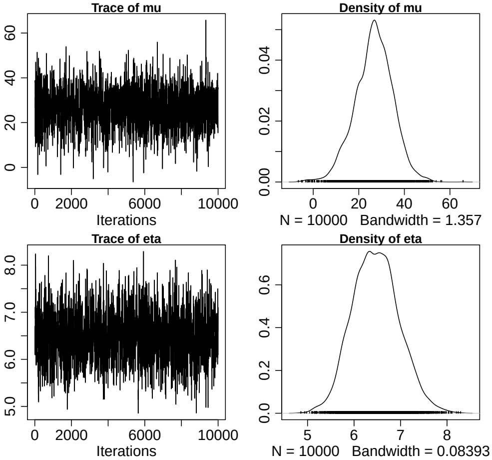
Comparison with MH algorithm

- For the Student's t model, we can also directly implement the Metropolis-Hastings algorithm, by using `rwmetrop()` and `indepmetrop`.
- To implement the MH algorithm, we first find the mode and the variance in normal approximation as before.
- For the random walk Metropolis algorithm, we use the option `scale=2` and set the variance `var` to be the variance in normal approximation.

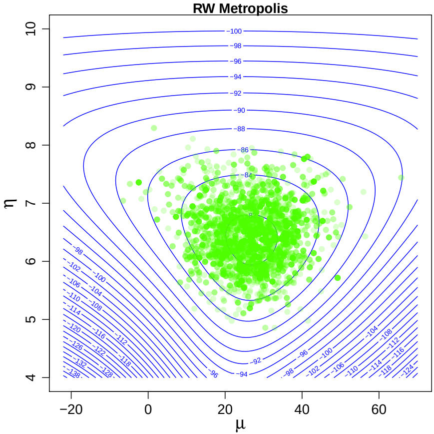
```
start <- c(25,6.5)
out <- optim(par=start,fn=logpost,hessian=TRUE,
             control=list(fnscale=-1),y=y,nu=nu)
(post.mode <- out$par)
(post.cov <- -solve(out$hessian))

proposal1 <- list(var=post.cov,scale=2)
fit1 <- rwmetrop(logpost,proposal1,start=post.mode,m=T,y=y,nu=nu)
```

- We can also use the coda package to obtain trace plots and density estimates from the simulated draws of μ and η from random walk Metropolis.



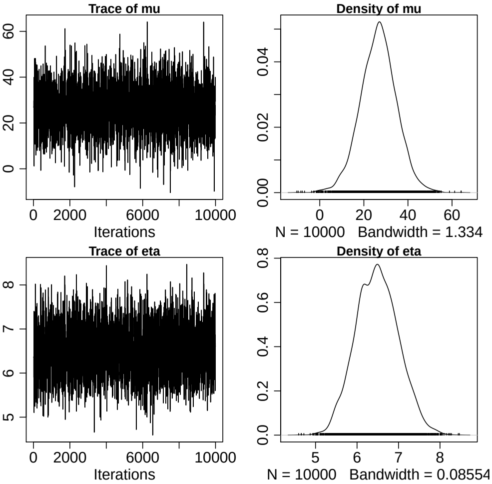
- We can also plot the contour plot of the log posterior, and overlay the posterior samples drawn from random walk Metropolis. We plot the parameters (μ, η).



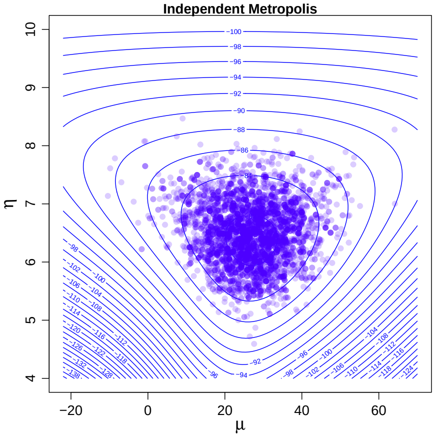
- For the independence Metropolis algorithm, in the `indepmetrop` function, inside the `proposal` argument, we set `mu` to be the posterior mode, and the variance `var` to be 4 times the variance in normal approximation.
- The acceptance rate of independent Metropolis algorithm with this configuration is 0.426.

```
> proposal2 <- list(mu=post.mode,var=4*post.cov)
> fit2 <- indepmetrop(logpost,proposal2,
                      start=post.mode,m=T,y=y,nu=nu)
> fit2$accept
       [,1]
[1,] 0.8661
```

- We can also use the `coda` package to obtain trace plots and density estimates from the simulated draws of μ and η from independent Metropolis.



- We can also plot the contour plot of the log posterior, and overlay the posterior samples drawn from independent Metropolis. We plot the parameters (μ, η).



- We can also use other methods for sampling from this posterior.
- In the file "Chapter8_Rcodes.r", I have included rejection sampling and SIR to sample from $p(\mu, \eta|\boldsymbol{y})$, using a t_3 distribution as the proposal.
- The normal approximation is another method to approximate the posterior.

- The table below displays the estimated 5%, 50%, and 95% quantiles of the marginal posteriors of μ and η, respectively. In addition, the acceptance rates for the Metropolis-Hastings algorithms are shown.

Method	Acceptance rate	μ	η
Gibbs sampler	(1)	(10.99, 26.32, 40.66)	(5.91, 6.71, 7.64)
Random walk MH	0.30	(12.77, 26.38, 39.66)	(5.61, 6.46, 7.32)
Independence MH	0.43	(12.85, 26.67, 40.14)	(5.67, 6.49, 7.34)
Rejection sampling		(12.74, 26.75, 40.46)	(5.68, 6.47, 7.31)
SIR		(12.39, 26.94, 39.90)	(5.64, 6.45, 7.26)
Normal approximation		(14.58, 26.92, 39.25)	(5.56, 6.36, 7.17)
Metropolis within Gibbs	0.45,0.43	(13.13, 26.64, 39.21)	(5.64, 6.45, 7.29)

- In the previous table, the last method is called Metropolis within Gibbs, which uses a Metropolis move within a Gibbs sampling framework.
- In situations where it is not convenient to sample directly from the conditional distributions, one can use a Metropolis algorithm such as the random walk type to simulate from each distribution.
- A Metropolis within Gibbs algorithm of this type is programmed in the function `gibbs` in the `LearnBayes` package.

8.3 Example 2: Student's t Model

- Suppose that $\theta_i^{(t)}$ represents the current value of θ_i in the simulation. Then a candidate value for θ_i is given by

$$\theta_i^* = \theta_i^{(t)} + c_i Z,$$

where $Z \sim \mathrm{N}(0, I)$, c_i is a fixed scale parameter, and I is the identity matrix (if θ_i is multivariate). The next simulated value of θ_i,

$$\theta_i^{(t+1)} = \begin{cases} \theta_i^* & \text{with probability } \min\left\{ \dfrac{p(\theta_i^* | \theta_{-i}^{(t)}, y)}{p(\theta_i^{(t)} | \theta_{-i}^{(t)}, y)}, 1 \right\} \\ \theta_i^{(t)} & \text{otherwise.} \end{cases}$$

8.3 Example 2: Student's t Model

- The usage of `gibbs` is as follows:

```
gibbs(logpost, start, m, scale, ...)
```

  One inputs `logpost` to be the function defining the log posterior, `start` to be the starting value of the simulation, `m` as the number of Gibbs cycles, and `scale` as a vector of scale parameters containing $(c_1, \ldots, c_p)$.

- The output of gibbs is a list; the component `par` is a matrix of simulated draws and `accept` is a vector, the ith element of which represents the acceptance rates for the Metropolis steps for θ_i.

```
post.sd <- sqrt(diag(post.cov)) # sd of each parameter
# post.cov is obtained from optim() before
# see detailed codes in Chapter8_Rcodes.r
set.seed(4234)
fit.mwg <- gibbs(logpost,start,m=T,scale=2.5*post.sd,y=y,nu=nu)
> fit.mwg$accept
       [,1]   [,2]
[1,] 0.4488 0.4334
```

- The following is the contour and scatter plot for Metropolis within Gibbs.