

Chapter 14

Gibbs sampling

14.1 The sensitivity and specificity of a test for a disease

Suppose that for a particular tropical disease no gold standard clinical test exists. Instead we have a test that is known to be imperfect; not always identifying a disease if the patient has the disease, and sometimes yielding false positives (patients that do not have the disease but test positive). However, by using this test in a clinical trial it is hoped that we can obtain better estimates for the disease sensitivity (S , the proportion of disease positive individuals who test positive) and specificity (C , the proportion of individuals who don't have the disease who test negative).

To do this we can construct a table of the observed and latent data for the test outcomes (see Table 14.1). In the table a and b are the number of observed positive and negative results respectively. Y_1 and Y_2 are latent variables that represent the gold standard – the true number of positive individuals out of a and b respectively.

Problem 14.1.1. Write down an expression for the likelihood, supposing that the prevalence for the disease is π . (Hint: multiply together the likelihoods corresponding to each of the cells in Table 14.1).

The likelihood of the top-left cell in Table 14.1 is $[\pi S]^{Y_1}$. This is the proportion of individuals with the disease (π) times the test specificity (S), giving the probability that we correctly identify one person as having the disease. This is raised to the power Y_1 since there are this many individuals with the disease in our sample.

Determining similar expressions for the other cells we obtain an overall likelihood,

		Truth		
		+	-	
Test	+	Y_1	$a - Y_1$	a
	-	Y_2	$b - Y_2$	b
		$Y_1 + Y_2$	$N - (Y_1 + Y_2)$	N

Table 14.1: Test outcomes versus true outcomes.

$$L(a, b | Y_1, Y_2, \pi, S, C) = \overbrace{[\pi S]^{Y_1}}^{\text{top-left: true +}} \times \overbrace{[\pi(1-S)]^{Y_2}}^{\text{bottom-left: false -}} \times \quad (14.1)$$

$$\underbrace{[(1-\pi)(1-C)]^{a-Y_2}}_{\text{top-right: false +}} \times \underbrace{[(1-\pi)C]^{b-Y_2}}_{\text{bottom-right: true -}} \quad (14.2)$$

Problem 14.1.2. Assuming priors of the form: $\pi \sim \text{beta}(\alpha_\pi, \beta_\pi)$, $S \sim \text{beta}(\alpha_S, \beta_S)$ and $C \sim \text{beta}(\alpha_C, \beta_C)$, it is possible to code up a Gibbs sampler for this problem [2] of the form,

$$Y_1 | a, \pi, S, C \sim \mathcal{B}\left(a, \frac{\pi S}{\pi S + (1-\pi)(1-C)}\right) \quad (14.3)$$

$$Y_2 | b, \pi, S, C \sim \mathcal{B}\left(b, \frac{\pi(1-S)}{\pi(1-S) + (1-\pi)C}\right) \quad (14.4)$$

$$\pi | a, b, Y_1, Y_2 \sim \text{beta}(Y_1 + Y_2 + \alpha_\pi, a + b - Y_1 - Y_2 + \beta_\pi) \quad (14.5)$$

$$S | Y_1, Y_2 \sim \text{beta}(Y_1 + \alpha_S, Y_2 + \beta_S) \quad (14.6)$$

$$C | a, b, Y_1, Y_2 \sim \text{beta}(b - Y_2 + \alpha_C, a - Y_1 + \beta_C) \quad (14.7)$$

Using the above expressions code up a working Gibbs sampler.

Problem 14.1.3. Suppose that out of a sample of 100 people, 20 of those tested negative and 80 positive. Assuming uniform priors on π , S and C , use Gibbs sampling to generate posterior samples for π . What do you conclude?

Since we have no idea what the true disease prevalence is, nor what the sensitivity or specificity of our test is, we cannot infer much here. So unsurprisingly, the posterior here is completely uninformative for π (left hand panel of Figure 14.1).

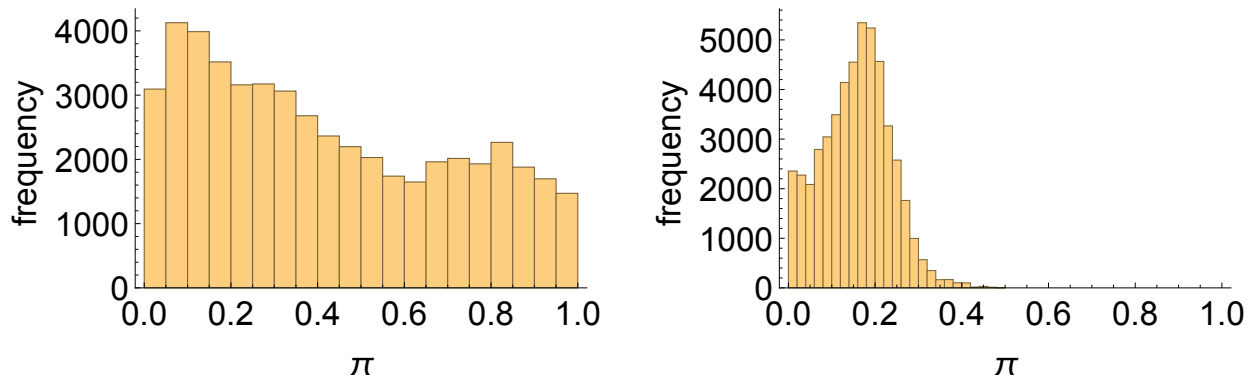


Figure 14.1: Posterior samples for left: uninformative, and right: informative priors being used for S and C .

Problem 14.1.4. Suppose that a previous study that compares the clinical test with a laboratory gold standard concludes that $S \sim \text{beta}(10, 1)$ and $C \sim \text{beta}(10, 1)$. Use Gibbs sampling to estimate the new posterior for π . Why does this look different to your previously-estimated distribution?

Since we have a reasonable knowledge of what the test sensitivity and specificity are, this means that we can now get a more concentrated estimate of what the disease prevalence is (right hand panel of Figure 14.1). This is the benefit of Bayesian statistics – we can put prior knowledge into a system and use that to determine better estimates of quantities of interest!

Problem 14.1.5. Suppose a previous analysis concluded that $\pi \sim \text{beta}(1, 10)$. Using this distribution as a prior, together with uniform priors on S and C , determine the posterior distributions for the test sensitivity and specificity respectively. Why does the test appear to be quite specific, although it is unclear how sensitive it is?

The posterior for S is fairly uniform between 0 and 1 (left hand panel of Figure 14.2). This is because using our prior for π we would assume that the true disease prevalence is somewhere around 10%,

```
hist(rbeta(100000, 1, 10))
```

So out of our sample of 100 individuals we would expect that only 10 of them have the disease. This is only a really small sample size, and even though it is potentially the case that we have identified all of these correctly, it is also possible that we have identified *none* of these correctly (if none of these fell within our 20 that tested positive). So intuitively, we still have no idea whether the people who tested positive are actually positive.

However, for C we expect that 90 people will not have the disease and we have predicted that 80 of them don't have the disease. This means regardless of who those people are, our test is pretty good at minimising false positives, and so we get a reasonably informative posterior for the specificity.

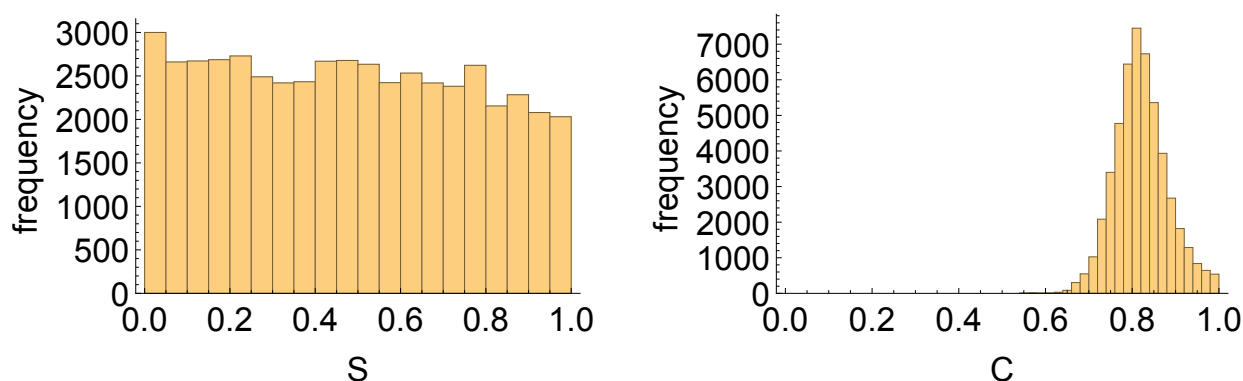


Figure 14.2: Posterior samples for left: test sensitivity (S), and right: specificity (C) assuming uniform priors on each of these parameters and $\pi \sim \text{beta}(1, 10)$.

Problem 14.1.6. Suppose that based on lab results you suppose that the test specificity $C \sim \text{beta}(10, 1)$, and $\pi \sim \text{beta}(1, 10)$, but the prior for S is still uniform. Explain the shape of the posterior for S now.

It is still uninformative (data not shown). Regardless of how specific our test is – there are still a large number of ways we could find 20/100 people having the disease. Only some of those ways actually include the 10 people that likely have the disease out of the positively-testing sample!

Problem 14.1.7. Now suppose that the sample size was 1000 people of which 200 tested positive. Using the same priors as the previous question, determine the posterior for S . What do you conclude about your test's sensitivity?

Still uninformative (data not shown). Nothing has changed from the previous question – we still have no idea whether the people who tested positive are actually positive. Collecting more data here won't help!

Problem 14.1.8. What do the previous results suggest is necessary to assess the sensitivity of a clinical test for a disease?

We need a gold standard to compare our test against! Intuitively, unless we know who the true positives are, we cannot hope to determine whether our test can detect the disease. Remember statistics will never give you a way to calculate something that can't be seen from a given experiment/data!

14.2 Coal mining disasters in the UK

The data in `gibbs_coal.csv` contains time series of the annual number of coal mining disasters in the UK from 1851-1961 [1]. In this question we are going to use Gibbs sampling to estimate the point in time when legislative and societal changes caused a reduction in coal mining disasters in the UK.

A model for the number of disasters D_t in a particular year t is,

$$D_t \sim \begin{cases} \text{Poisson}(\lambda_1), & \text{if } t \leq n \\ \text{Poisson}(\lambda_2), & \text{if } t > n \end{cases}$$

where λ_1 and λ_2 are the early and late mean disaster rates in the UK, and n is the time where the change in rates occurred.

Problem 14.2.1. Graph the data over time. Around what year (n) does it appear the disaster rate occurred?

This appears to occur roughly from 1880-1897.

Problem 14.2.2. Assuming the same $\lambda_i \sim \Gamma(a, b)$ priors for $i = \{1, 2\}$, and a discrete uniform prior for n between 1851-1861, determine out an expression for the full (un-normalised) posterior density.

The full density,

$$p(\lambda_1, \lambda_2, n | x_{1:N}, a, b) \propto p(x_{1:n} | \lambda_1) p(x_{n+1:N} | \lambda_2) p(\lambda_1 | a, b) p(\lambda_2 | a, b) p(n) \quad (14.8)$$

which can be explicitly calculated,

$$\begin{aligned}
 p(x_{1:n}|\lambda_1)p(x_{n+1:N}|\lambda_2)p(\lambda_1|a,b)p(\lambda_2|a,b)p(n) = & Poisson(x_{1:n}|\lambda_1) \times \\
 & Poisson(x_{n+1:N}|\lambda_2) \times \\
 & \Gamma(\lambda_1|a,b)\Gamma(\lambda_2|a,b) \times \\
 & p(n)
 \end{aligned} \tag{14.9}$$

Problem 14.2.3. Determine the conditional distribution for λ_1 (i.e. $p(\lambda_1|x_{1:n}, n, a, b)$) by finding all those terms in the density that include λ_1 , and removing the rest as constants of proportionality. (Hint: remember that a gamma prior is conjugate to a Poisson likelihood.)

We have that,

$$p(\lambda_1|x_{1:n}, n, a, b) \propto Poisson(x_{1:n}|\lambda_1)\Gamma(\lambda_1|a, b) \tag{14.10}$$

But we know that the gamma distribution is the conjugate prior to the Poisson likelihood, and so obtain,

$$\lambda_1|x_{1:n}, n, a, b \sim \Gamma(a + \sum_{t=1}^n x_t, b + n) \tag{14.11}$$

Problem 14.2.4. Using your answer to the previous problem write down the conditional distribution for λ_2 .

By examining the previous answer we obtain,

$$\lambda_2|x_{n+1:N}, n, a, b \sim \text{gamma}(a + \sum_{t=n+1}^N x_t, b + N - n) \tag{14.12}$$

Problem 14.2.5. By collecting the terms that depend on n show that its conditional density can be written as,

$$p(n|x_{1:N}, \lambda_1, \lambda_2) \propto \lambda_1^{\sum_{t=1}^n x_t} e^{-n\lambda_1} \times \lambda_2^{\sum_{t=n+1}^N x_t} e^{-(N-n)\lambda_2} \tag{14.13}$$

Taking only those terms that depend on n ,

$$p(n|x_{1:N}, \lambda_1, \lambda_2) \propto Poisson(x_{1:n}|\lambda_1) \times Poisson(x_{n+1:N}|\lambda_2) \tag{14.14}$$

Note that the discrete uniform prior for n does not appear in the above because this is only a constant ($1/N$). Now writing out each term explicitly, we have,

$$p(n|x_{1:N}, \lambda_1, \lambda_2) \propto \left(\prod_{t=1}^n \frac{\lambda_1^{x_t} e^{-\lambda_1}}{x_t!} \right) \times \left(\prod_{t=n+1}^N \frac{\lambda_2^{x_t} e^{-\lambda_2}}{x_t!} \right) \quad (14.15)$$

$$\propto \lambda_1^{\sum_{t=1}^n x_t} e^{-n\lambda_1} \times \lambda_2^{\sum_{t=n+1}^N x_t} e^{-(N-n)\lambda_2} \quad (14.16)$$

where we note that we have dropped the $x_t!$ terms because these are constants.

Problem 14.2.6. Write a function in R that calculates the un-normalised expression for $p(n|x_{1:N}, \lambda_1, \lambda_2)$ for a single value of n . Hint: remember that the change point cannot occur at the last data point, and so return 0 for this case!

This can be done using,

```
fPointnConditional <- function(n, X, lambda.1, lambda.2){
  N <- length(X)
  aSum.1 <- sum(X[1:n])
  aSum.2 <- sum(X[(n+1):N])
  if(n == 111){
    return(0)
  }else{
    return(lambda.1 ^ aSum.1 * exp(-n * lambda.1) * lambda.2 ^ aSum.2 *
           exp(-(N - n) * lambda.2))
  }
}
```

Problem 14.2.7. Create a function that calculates the discrete probability distribution across all values of n . (Hint: remember this must be a valid probability distribution.)

All you need to do is calculate the point-wise un-normalised density, and then normalise it.

```
fAllnConditional <- function(X, lambda.1, lambda.2){
  lProb <- sapply(seq_along(X), function(n)
                 fPointnConditional(n, X, lambda.1, lambda.2))
  return(lProb / sum(lProb))
}
```

Problem 14.2.8. Create a function that independently samples from the discrete distribution for n you calculate. (Hint use the “sample” function.)

Such a function is,

```
fSampleN <- function(lambda.1, lambda.2, X){
  sample(seq_along(X), 1, prob=fAllnConditional(X, lambda.1, lambda.2))
}
```

For $\lambda_1 = 3$ and $\lambda_2 = 1$ we can draw a histogram of 1,000 draws from the distribution for n ,

```
hist(sapply(1:1000, function(i) fSampleN(3, 1, X)), xlab='n')
```

which should be peaked around $n = 40$ which corresponds to the change-point being 1891 since we are using reasonable values for λ .

Problem 14.2.9. Write functions to independently sample from the conditional distributions for λ_1 and λ_2 that you previously determined.

Two such functions are,

```
fSampleLambda1 <- function(X, a, b, n){
  aSum <- sum(X[1:n])
  return(rgamma(1, (a + aSum), (b + n)))
}

fSampleLambda2 <- function(X, a, b, n){
  N <- length(X)
  aSum <- sum(X[(n + 1):N])
  return(rgamma(1, (a + aSum), (b + N - n)))
}
```

Problem 14.2.10. Combine all three previously created sampling functions to create a working Gibbs sampler. Hence estimate the change-point and its 95% central credible intervals.

I first create a function that does one iteration for the Gibbs sampler,

```
fGibbsSingle <- function(n, lambda.1, lambda.2, X, a, b){
  n <- fSampleN(lambda.1, lambda.2, X)
  lambda.1 <- fSampleLambda1(X, a, b, n)
  lambda.2 <- fSampleLambda2(X, a, b, n)
  return(list(n=n, lambda.1=lambda.1, lambda.2=lambda.2))
}
```

And then iterate using the above to create a Gibbs sampler,

```
fGibbs <- function(numIterations, n, lambda.1, lambda.2, X, a, b){
  mSamples <- matrix(nrow=numIterations, ncol=3)
  mSamples[1,] <- c(n, lambda.1, lambda.2)
  for(i in 2:numIterations){
    nPrev <- mSamples[(i - 1), 1]
    lambda.1Prev <- mSamples[(i - 1), 2]
    lambda.2Prev <- mSamples[(i - 1), 3]
  }
```

```

lParams <- fGibbsSingle(nPrev, lambda.1Prev, lambda.2Prev, X, a, b)
mSamples[i,] <- c(lParams$n, lParams$lambda.1, lParams$lambda.2)
}
colnames(mSamples) <- c("n", "lambda.1", "lambda.2")
return(as.data.frame(mSamples))
}

```

I start the Gibbs sampler at non-random values here, but to do this properly I should pick these using some sampling distribution,

```

mTest <- fGibbs(10000, 40, 3, 1, X, 1, 1)
hist(mTest$n)

```

The median estimate I obtain for n is 40/41, with its 2.5% and 97.5% quantiles being 36 and 46 respectively.

Problem 14.2.11. Using your sampler determine posterior median estimates for λ_1 and λ_2 .

The posterior medians are roughly 3.06, and 0.93 respectively.

14.3 Bayesian networks

Suppose that when you leave your house in the morning you notice that the grass is wet. However you do not know whether the grass is wet because of last night's rain, or because the sprinkler went on in the night. You want to determine the cause of the wet grass, because this affects whether you need to water your plants in your windowsill. The causal pathway of the grass being wet is shown in Figure 14.3, along with the probabilities of the states. Here “cloudy” means that the night was completely overcast.

Problem 14.3.1. Show that the probability that it was cloudy last night conditional on the sprinkler being on, that it rained, and the grass being wet is approximately 0.444.

This can be done with the following computation,

$$p(c|r, s, w) = p(c|r, s) = \frac{p(r, s|c)p(c)}{p(r, s)} \quad (14.17)$$

$$= \frac{p(r|c)p(s|c)p(c)}{p(r|c)p(s|c)p(c) + p(r|\neg c)p(s|\neg c)p(\neg c)} \quad (14.18)$$

$$= \frac{0.8 \times 0.1 \times 0.5}{0.8 \times 0.1 \times 0.5 + 0.2 \times 0.5 \times 0.5} \quad (14.19)$$

$$\approx 0.444 \quad (14.20)$$

Problem 14.3.2. Show that the probability that it rained given that it was cloudy, the sprinkler was on, and the grass is wet is approximately 0.815.

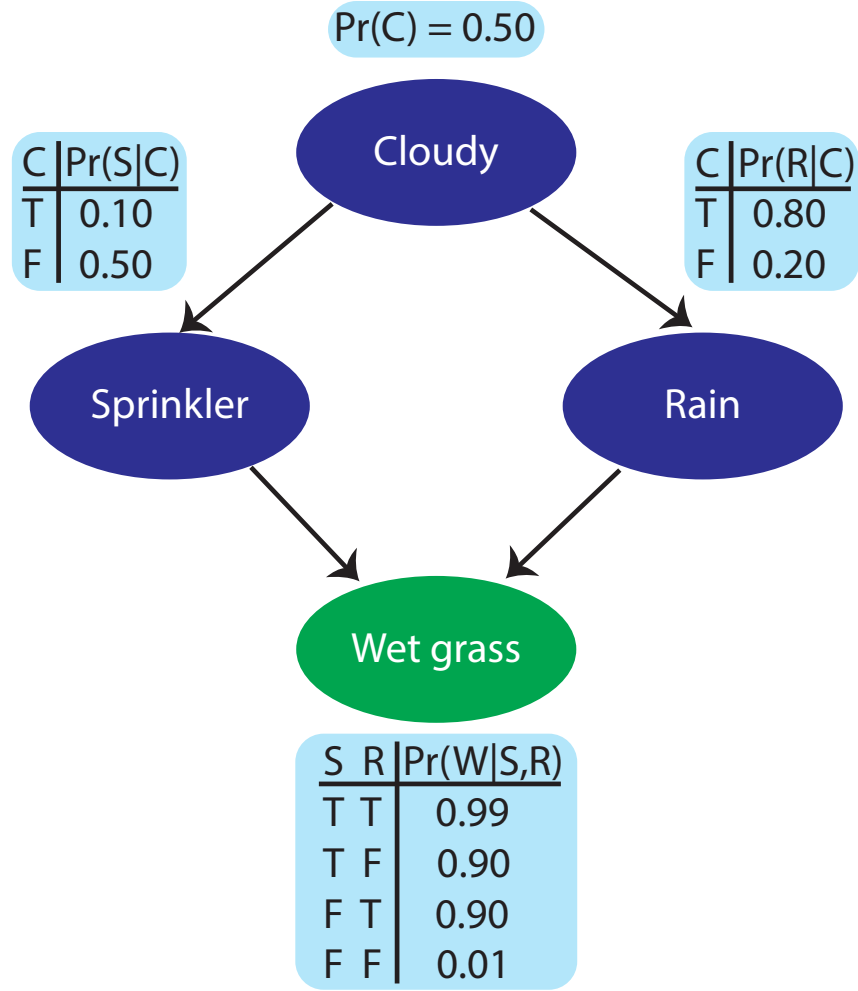


Figure 14.3: A Bayesian network for the wet grass example.

This can be done as follows,

$$p(r|c, s, w) = \frac{p(w|r, c, s)p(r|c, s)}{p(w|c, s)} \quad (14.21)$$

$$= \frac{p(w|r, s)p(r|c)}{p(w|c, s)} \quad (14.22)$$

$$= \frac{p(w|r, s)p(r|c)}{p(w|r, c, s)p(r|c) + p(w|\neg r, c, s)p(\neg r|c)} \quad (14.23)$$

$$= \frac{p(w|r, s)p(r|c)}{p(w|r, s)p(r|c) + p(w|\neg r, s)p(\neg r|c)} \quad (14.24)$$

$$= \frac{0.99 \times 0.8}{0.99 \times 0.8 + 0.90 \times 0.2} \quad (14.25)$$

$$\approx 0.815 \quad (14.26)$$

Problem 14.3.3. The remaining (non-trivial) conditional probabilities are $p(c|\neg r, s, w) = 0.048$

and $p(r|\neg c, s, w) = 0.216$. Suppose that when we walk outside we see that the grass is wet, and we also know that the sprinkler went on last night (we were woken by its noise). Create a Gibbs sampler to determine the unconditional probability that it was cloudy. Then find the unconditional probability as to whether it rained.

The conditional probabilities that we require to do this problem. These are shown in Figure 14.4. I implemented this sampler in steps. The first thing I did was create a transition function that accepts a probability and returns a binary outcome,

```
fTransition <- function(aProb){
  indicator <- rbinom(1, 1, aProb)
  return(indicator)
}
```

Then we need a function that calculates the probabilities of the various outcomes,

```
fProbCalculator <- function(cloud_or_rain, cloud, rain){
  if(cloud_or_rain){
    if(rain){
      aProb <- 0.444
    }else{
      aProb <- 0.048
    }
  }else{
    if(cloud){
      aProb <- 0.815
    }else{
      aProb <- 0.216
    }
  }
  return(aProb)
}
```

where `cloud_or_rain` is a boolean that indicates whether we are updating cloud (if true) or rain (if false). I then created two functions that update the state, and choose which state to update respectively,

```
fGibbsStep <- function(cloud_or_rain, cloud, rain){
  aProb <- fProbCalculator(cloud_or_rain, cloud, rain)
  aNewState <- ifelse(fTransition(aProb) == 1, T, F)
  if(cloud_or_rain){
    cloud <- aNewState
  }else{
    rain <- aNewState
  }
  return(list(cloud=cloud, rain=rain))
}
```

```
fGibbsSelector <- function(cloud, rain){
  cloud_or_rain <- ifelse(rbinom(1, 1, 0.5), T, F)
  return(fGibbsStep(cloud_or_rain, cloud, rain))
}
```

Finally we iterate over the latter function to create our working Gibbs sampler,

```
fGibbsComplete <- function(numIterations, cloud, rain){
  lCloud <- vector(length=numIterations)
  lRain <- vector(length=numIterations)
  lCloud[1] <- cloud
  lRain[1] <- rain
  for(i in 2:numIterations){
    lState <- fGibbsSelector(lCloud[i - 1], lRain[i - 1])
    lCloud[i] <- lState$cloud
    lRain[i] <- lState$rain
  }
  return(list(cloud=lCloud, rain=lRain))
}
```

Now running the sampler for 10,000 iterations we obtain unconditional probabilities of cloud and rain,

```
lTest <- fGibbsComplete(10000, T, T)
hist(ifelse(lTest$cloud, 1, 0))
hist(ifelse(lTest$rain, 1, 0))
mean(mean(ifelse(lTest$cloud, 1, 0)))
mean(mean(ifelse(lTest$rain, 1, 0)))
```

so that $Pr(cloud) \approx 0.17$ and $Pr(rain) \approx 0.32$.

Problem 14.3.4. Using your Gibbs sampler determine the joint probability that it was cloudy and it rained.

This is straightforward to obtain from our sampler,

```
lTest <- fGibbsComplete(100000, T, T)
mean(ifelse(lTest$cloud, 1, 0)==1 & ifelse(lTest$rain, 1, 0)==1)
```

which should be roughly 0.14.

Problem 14.3.5. Visualise the path of your Gibbs sampler by creating a network graph showing the frequency of transitions between the four states. Hint 1: first create an adjacency matrix by calculating the frequency of transitions between all states. Hint 2: to visualise the graph in R use the iGraph package.

First create indicator variables representing the four states,

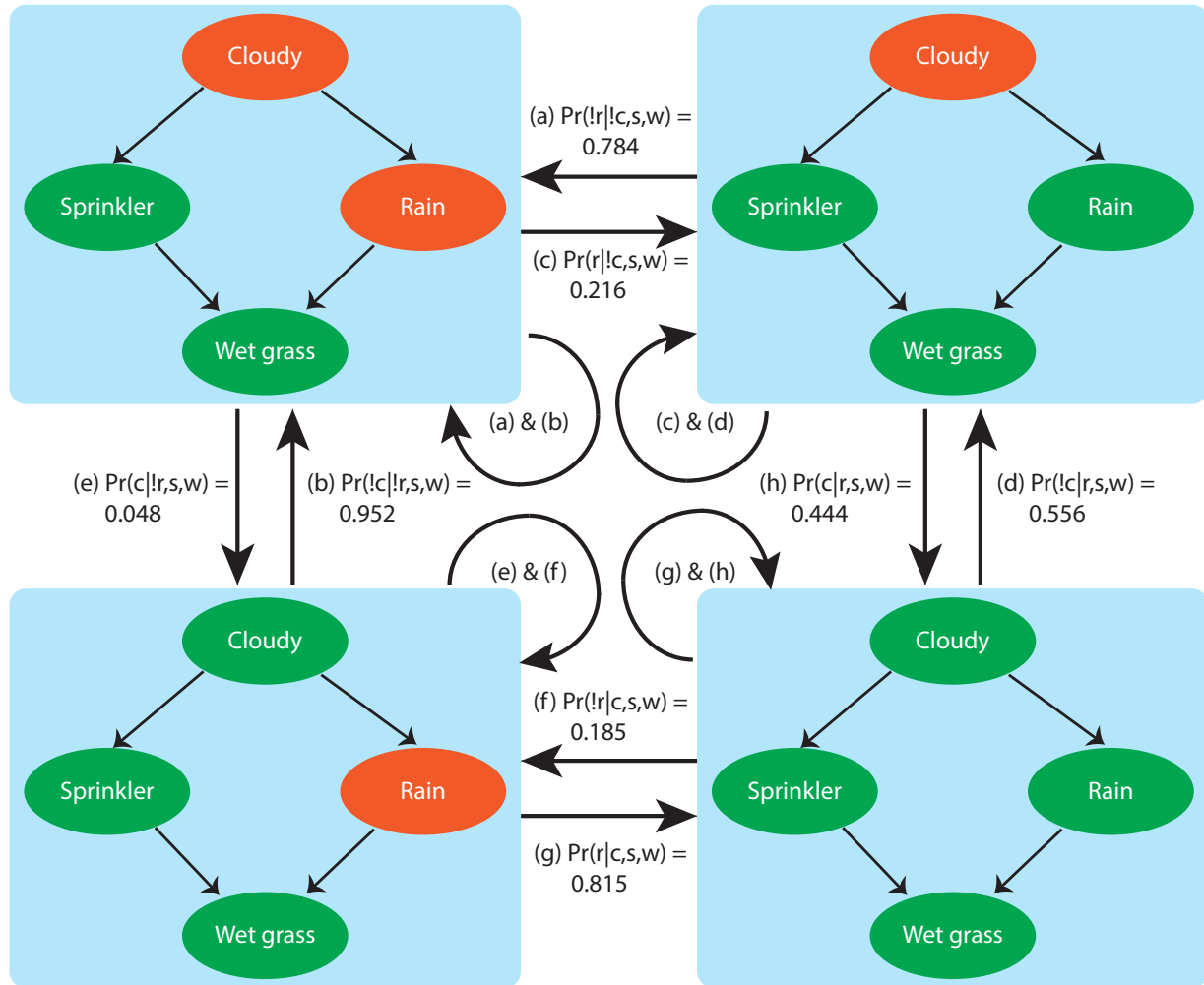


Figure 14.4: A schematic of a Gibbs sampler for the wet grass example. Green means that the event occurred, and orange that it didn't. **Replace ! with \neg for consistency.**

```
aDF <- data.frame(cloud=1Test$cloud, rain=1Test$rain,
                  A=(!1Test$cloud) & (!1Test$rain),
                  B=(1Test$cloud) & (!1Test$rain),
                  C=(!1Test$cloud) & (1Test$rain),
                  D=(1Test$cloud) & (1Test$rain))
```

Then create a function that determines the transitions between two states,

```
fGetTransitions <- function(1Test, from, to){
  aLen <- length(1Test$cloud)
  vTransition <- vector(length=(aLen - 1))
  for(i in 2:aLen){
    if(aDF[i - 1, (2 + from)] & aDF[i, (2 + to)])
      vTransition[i] <- 1
  }
}
```

```

    else
      vTransition[i] <- 0
    }
    return(vTransition)
  }

```

Then determine all possible combinations of state transitions, and fill the adjacency matrix,

```

lCombs <- expand.grid(rep(list(1:4), 2))
mAdjacency <- matrix(nrow=4, ncol=4)
for(i in 1:16){
  print(i)
  mAdjacency[lCombs[i, 1], lCombs[i, 2]] <- sum(fGetTransitions(lTest,
                                                                lCombs[i, 1], lCombs[i, 2]))
}

```

Finally use iGraph to visualise the results,

```

library(igraph)
rownames(mAdjacency) <- c('C=0,R=0', 'C=1,R=0', 'C=0,R=1', 'C=1,R=1')
colnames(mAdjacency) <- rownames(mAdjacency)
ig <- graph.adjacency(mAdjacency, mode="directed", weighted=TRUE)
plot(ig, edge.width=E(ig)$weight / 10000 + 1, edge.label=round(E(ig)$weight, 3),
     edge.arrow.size=1.5, edge.curved=TRUE)

```

The resultant graph should look something like Figure 14.5.

14.4 Proofs

Problem 14.4.1. Prove that the Gibbs sampler can be viewed as a case of Metropolis-Hastings.

The acceptance ratio for the Metropolis-Hastings is of the form,

$$r = \frac{p(\theta_{t+1}|data)}{p(\theta_t|data)} \times \frac{J(\theta_t|\theta_{t+1})}{J(\theta_{t+1}|\theta_t)}. \quad (14.27)$$

In Gibbs sampling the jumping probabilities are conditional distributions, so that,

$$J(\theta_{t+1}|\theta_t) = p(\theta_{t+1}|\theta_t, data). \quad (14.28)$$

This means we can rewrite the above ratio as,

$$r = \frac{p(\theta_{t+1}|data)}{p(\theta_t|data)} \times \frac{p(\theta_t|\theta_{t+1}, data)}{p(\theta_{t+1}|\theta_t), data} \quad (14.29)$$

$$= \frac{p(\theta_{t+1}, \theta_t|data)}{p(\theta_{t+1}, \theta_t|data)} \quad (14.30)$$

$$= 1 \quad (14.31)$$

So in other words we always accept, as we do in Gibbs sampling.

Problem 14.4.2. For the following bivariate normal density,

$$\begin{pmatrix} u \\ c \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right],$$

show that the conditional densities are of the form,

$$u|c \sim \mathcal{N}(a + \rho(c - b), \sqrt{1 - \rho^2}) \quad (14.32)$$

$$c|u \sim \mathcal{N}(b + \rho(u - a), \sqrt{1 - \rho^2}) \quad (14.33)$$

There exist many versions of this proof online.

Problem 14.4.3. Show that the following changes of variables for a bivariate normal (with $\rho = 0.5$) as described in this Chapter result in uncorrelated components,

$$C = c - u \quad (14.34)$$

$$U = c + u \quad (14.35)$$

Examining first C , we have that its mean,

$$\mathbb{E}(C) = \mathbb{E}(c) - \mathbb{E}(u) \quad (14.36)$$

$$= b - a. \quad (14.37)$$

And its variance,

$$\text{var}(C) = \text{var}(c) + \text{var}(u) - 2\text{covar}(c, u) \quad (14.38)$$

$$= 2 - 2\rho \quad (14.39)$$

$$= 2(1 - \rho) \quad (14.40)$$

Examining the covariance between C and U ,

$$\text{covar}(C, U) = \text{covar}(c - u, c + u) \quad (14.41)$$

$$= 0. \quad (14.42)$$

As desired, this means that the bivariate normal for the transformed coordinates is of the form,

$$\begin{pmatrix} C \\ U \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} b - a \\ a + b \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right]$$

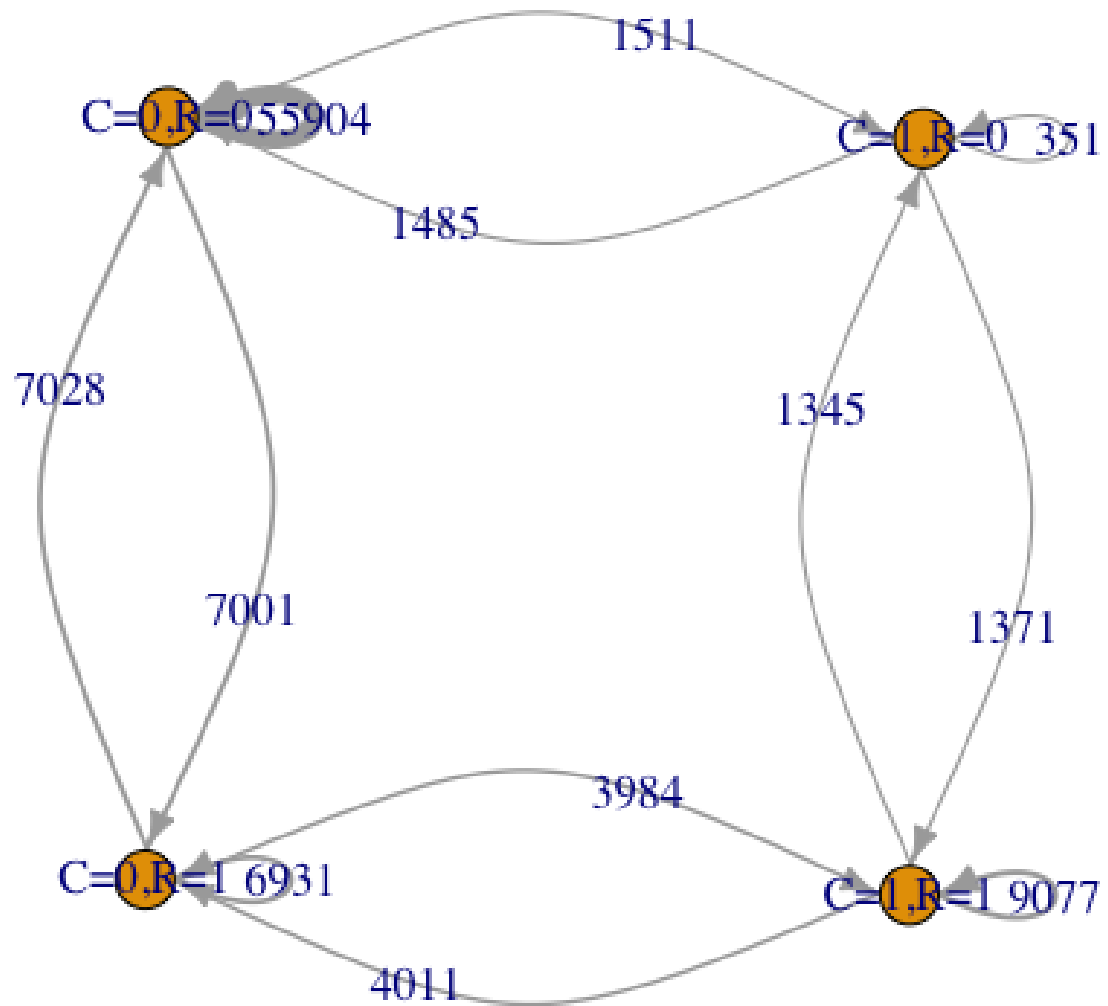


Figure 14.5: A graph representing the transitions between the four states for the Gibbs sampler of the cloud and rain example.

Bibliography

- [1] RG Jarrett. A note on the intervals between coal-mining disasters. *Biometrika*, 66(1):191–193, 1979.
- [2] Lawrence Joseph, Theresa W Gyorkos, and Louis Coupal. Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard. *American Journal of Epidemiology*, 141(3):263–272, 1995.