

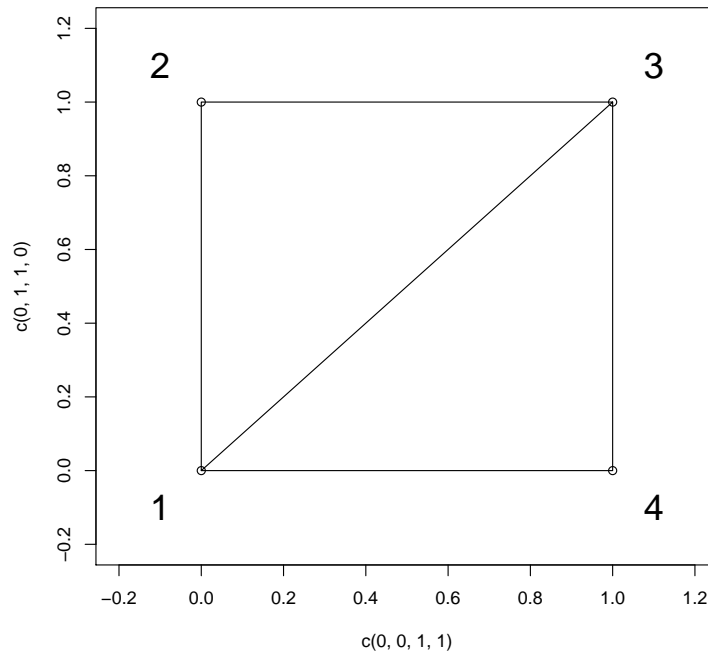
36-724 Homework 2

Due: Monday, February 6, 14:30 EST, by email to Dansci
(724homeworksgohere@gmail.com).

```
pdf("my-file-name.pdf")
plot(...) #your plot commands here
dev.off()
```

Please submit 1) your R code, set to output your computational solutions, with a name like `yourname_hw1.R`; 2) your analytical solutions as a PDF (in LaTeX, scanned-in pencil/pen, whichever for now), named `yourname_hw1.pdf`.

- Homework 1 grading. You will be sent the code from someone else in the class, as well as a general solution. Run each of these scripts using the command `source()` and verify that the output quantities match. Indicate which quantities do not match, bearing in mind that simulated quantities need not match exactly.
- Markov Chains I. The weather in a city has been observed to follow a Markov chain: If yesterday was sunny, there is a 75% chance it will be sunny today. If yesterday was cloudy, there is a 50% chance it will be cloudy today. The weather is only sunny or cloudy in this city. Let W_i be 1 if the weather on day i is sunny.
 - What is the state transition matrix for this Markov chain?
 - Find the limiting distribution of the weather in this city: that is, solve for $Pr(\text{sunny}) = \mu_1$ in the limiting distribution vector algebraically.
 - Compute** this result by solving for the eigenvectors of the transition matrix (the function `eigen()` returns the eigenvalues and eigenvectors of a matrix.).
 - Compute** this result by directly simulating the Markov chain (W_i, W_{i+1}, \dots) in R. Start with a sunny day ($W_0 = 1$), take 110 draws of the chain, and discard the first 10 results (the “burn-in” period).
 - Compute** this result in R by approximating the limiting distribution by P^n , and compare to P^{n+1} to ensure convergence.
- Markov Chains II. Consider a random walk on a square with an extra diagonal edge:



The walker walks from one node to the next at each time point, choosing an outgoing path with equal probability; i.e. when at node 1, the next transition is made to 2, 3 or 4 with probability $1/3$ each.

- (a) What is the state transition matrix for this Markov chain?
 - (b) Find the limiting distribution of the walker's state: that is, solve for $Pr(X_1)$, $Pr(X_2)$ and $Pr(X_3)$ in the limiting distribution vector algebraically noting that $Pr(X_4) = 1 - Pr(X_1) - Pr(X_2) - Pr(X_3)$.
 - (c) **Compute** this result by solving for the eigenvectors of the transition matrix.
 - (d) **Compute** this result by directly simulating the Markov chain (W_i, W_{i+1}, \dots) . Start with the walker on state 1 ($W_0 = 1$), take 550 draws of the chain, and discard the first 50 results (the "burn-in" period).
 - (e) **Compute** this result by approximating the limiting distribution by P^n , and compare to P^{n+1} to ensure convergence.
4. Bayesian Analysis of Markov Chains I. Take the weather example in question 1, but instead of the known probabilities, use the observations in the vector `weather` in the file `weathersequence.RData` from the website. **Compute** the following steps:
- (a) Collect the observations in the chain into the observation matrix

$$\begin{bmatrix} X_{00} & X_{01} \\ X_{10} & X_{11} \end{bmatrix}.$$

If the prior distribution for each "sunny day" probability (p_{01} and p_{11}) is $Beta(\frac{1}{2}, \frac{1}{2})$, determine the posterior distributions $f(p_{01}|X_{00}, X_{01})$ and $f(p_{11}|X_{10}, X_{11})$.

- (b) Take one draw from the posterior distributions for p_{01} and p_{11} to form a draw from the random matrix. Using this, find the limiting distribution of this matrix using the eigenvalue method.
- (c) Repeat this procedure 100 times to simulate from the posterior distribution of the fraction of sunny days, $f(\mu_1(p_{01}, p_{11})|X)$.
- (d) Confirm this result by plotting a histogram of your observations

```
hist(mu.1)
```

and then drawing a vertical line of the “true” value obtained in Problem 1.

```
abline(v=true.value, col=2)
```

5. Bayesian Analysis of Markov Chains II. Take the random walk example in question 2, but instead of the known probabilities, use the observations in the vector `walker` in the file `walker.RData` from the website. **Compute** the following steps:

- (a) Collect the observations in the chain into the observation matrix

$$\begin{bmatrix} X_{11} & \dots & X_{14} \\ & \dots & \\ X_{41} & & X_{44} \end{bmatrix}$$

If the prior distribution for each state transition probability group $(p_{i1}, p_{i2}, p_{i3}, p_{i4})$ is $Dirichlet(\frac{1}{2}, \dots, \frac{1}{2})$, determine the posterior distributions for each row of the state transition matrix.

- (b) Take one draw from each of the posterior draws for each row to form a draw from the random matrix. Using this, find the limiting distribution of this matrix using the eigenvalue method.
- (c) Repeat this procedure 100 times to simulate from the joint posterior distribution of the states, $f(\mu_1(p), \dots, \mu_4(p)|X)$.
- (d) Confirm this result by plotting a histogram of your observations for state 1

```
hist(mu.1)
```

and then drawing a vertical line of the “true” value obtained in Problem 2.

```
abline(v=true.value, col=2)
```

6. Metropolis Algorithm I. Consider a standard t_6 target distribution that we wish to sample with the Metropolis algorithm, with a $Normal(X_{last}, \sigma^2)$ proposal (for pedagogy’s sake). The R function `dt()` will give the density of the standard t -distribution. **Compute** the following steps:

- (a) Write a Metropolis sampler corresponding to this target and proposal as a function in R. Ensure that the proposal width σ and the number of iterations `nn` are options in the function. The output of this function should be the Markov chain. Run it for 10000 steps. What is the mean acceptance rate? What is the mean absolute difference between adjacent values of the sampler? Show a trace plot of the sampler, an autocorrelation plot, and a histogram of the draws.

- (b) Run this function for a range of sample widths `nn=10000` iterations over the range of widths `seq(0.1, 6, by=0.1)`. Use the `sapply(seq(0.1, 6, by=0.1), FUN=function(kk) my.metropolis.sampler(kk, nn=10000))` form to obtain a matrix for your solution. Report the mean acceptance probabilities, mean transition lengths, and the lag-1 autocorrelations for each sampler.
- (c) Based on these runs, which sampler width is the most efficient in producing a sample from the target density? What is the empirical acceptance probability for that width?

7. Metropolis Algorithm II. Consider an equal mixture of two Pareto distributions with parameters $(2, 1)$ and $(3, 1)$; that is, the density function is $p(x) = \mathbb{I}(x > 1)(\frac{1}{2} \frac{2}{x^3} + \frac{1}{2} \frac{3}{x^4})$. You may encode this in R with the function

```
dpareto.simple <- function(xx) {
  outcome <- xx
  outcome[xx>=1] <- (2/xx[xx>=1]^3 + 3/xx[xx>=1]^4)/2
  outcome[xx<1] <- 0
  return(outcome)
}
```

to guard against improper values. **Compute** the following steps:

- (a) Write a Metropolis sampler for this distribution with proposal distribution $X'_{n+1} \sim N(X_n, \sigma^2)$ with $\sigma^2 = 1$ and run it for 10000 steps. What is the mean acceptance rate? What is the mean absolute difference between adjacent values of the sampler? Show a trace plot of the sampler, an autocorrelation plot, and a histogram of the draws.
- (b) Rerun the sampler with proposal variances $\sigma^2 = \{0.01, 0.09, 0.25, 4, 16\}$. Report the mean acceptance probabilities, mean transition lengths, and the lag-1 autocorrelations for each sampler.
- (c) Based on these runs, which sampler is the most efficient in producing a sample from the target density?