Extra exercises for STK4051 - Computational statistics

Geir Storvik

Autum 2017

Exercise 1 (Convergence of Newton's method)
(Adapted from (Lange, 2010, exercise 5.13))
Assume we want to extract the root of the function

$$g(x) = x^m - c$$

for $m > 1, c > 0$.

(a). Find the (only) exact root of $g(x)$.

(b). Show that Newton's method is given by

$$x_n = x_{n-1}\left(1 - \tfrac{1}{m} + \tfrac{c}{mx_{n-1}^m}\right).$$

(c). Show that $x_n \geq c^{1/m}$ for all $x_{n-1} > 0$.

Hint: This part can be somewhat difficult. Perhaps easiest to start backwards, what requirements are needed for $x\left(1 - \frac{1}{m} + \frac{c}{mx^m}\right) \geq c^{1/m}$ and then relate this to $y = c/x^m$.

(d). Show that $x_n \leq x_{n-1}$ whenever $x_{n-1} \geq c^{1/m}$

(e). What does this imply if we start with $x_0 > 0$?

Exercise 2 (Divergence of Newton's method)
(Adapted from (Lange, 2010, exercise 5.12))
Assume we want to find the roots for the two functions

$$f(x) = \begin{cases} \sqrt{x} & x \geq 0 \\ -\sqrt{-x} & x < 0 \end{cases}$$

$$g(x) = x^{1/3}$$

What happens when you apply Newton's method in these cases?

Exercise 3 (Proof of convergence for fixed-point iteration)
Assume a function $G(x)$ which is contractive, that is

- $G(x) \in [a, b]$ whenever $x \in [a, b]$, and
- $|G(x_1) - G(x_2)| \leq \lambda |x_1 - x_2|$ for all $x_1, x_2 \in [a, b]$ for some $\lambda \in (0, 1)$

We will show that then there exist a unique fixed point x^* in the interval, and that the fixed-point algorithm will converge to it from any starting point in the interval. The fixed-point algorithm is given by

$$x_{k+1} = G(x_k)$$

(a). Show that for $x_0 \in [a, b]$ then $x_k \in [a, b]$ for all k.

(b). Show that $|x_{k+1} - x_k| \leq \lambda^k |x_1 - x_0|$

(c). Show that for $m > n$ we have $|x_m - x_n| \leq \frac{\lambda^n}{1-\lambda} |x_1 - x_0|$

Hint: Show first that $|x_m - x_n| \leq \sum_{k=n}^{m-1} |x_{k+1} - x_k|$ using Jensen's equality.

A Cauchy sequence is a sequence $\{z_k\}$ if for every $\varepsilon > 0$ there exist an $N > 1$ such that for $n, m > N$ we have $|z_n - z_m| < \varepsilon$. An important property of a Cauchy sequence is that $\lim_{k \to \infty} z_k$ exists.

(d). Argue why $\{x_k\}$ is a Cauchy sequence and use this to show that the limit x_∞ exists.

(e). Assume there exist another fixed point $y_\infty \neq x_\infty$. Then show that

$$|x_\infty - y_\infty| \leq \lambda |x_\infty - y_\infty|$$

What does this say about the uniqueness of the fixed point?

(f). Assume $f(x) = -x^2 + x + \frac{1}{4}$. Show that

(i) the function is not contractive for $x_0 \in (-\frac{1}{2}, \frac{1}{2})$

(ii) the fixed-point algorithm converges to $\frac{1}{2}$ for $x_0 \in (-\frac{1}{2}, \frac{3}{2})$

(iii) What happens if x_0 is outside this interval?

Exercise 4 (Variable selection and neighborhood)
Assume we have a linear regression model

$$Y = \beta_0 + \sum_{j: j \in \mathcal{M}} \beta_j x_j + \varepsilon$$

where $M \subset \{1, ..., p\}$. Our aim is to find the best subset M based on some data $\{(y_i, \boldsymbol{x}_i), i = 1, ..., n\}$ and some performance criterion (e.g. AIC).

For many of the optimization methods discussed, a neighborhood of a current solution is needed. We will look at different choices of neighborhoods in this exercise.

(a). Introduce the p-vector $\boldsymbol{\theta}$ where $\theta_j = 1$ if $j \in M$ and 0 otherwise. Argue that there is a one-to-one correspondence between M and $\boldsymbol{\theta}$.

(b). We will now consider four different neighborhoods:

$$\begin{aligned}
\mathcal{N}_1(\boldsymbol{\theta}) =& \{\boldsymbol{\theta}^*; \exists k \text{ such that } \theta_j^* = \theta_j \text{ for all } j \neq k \text{ and } \theta_k^* \neq \theta_k\}\\
\mathcal{N}_2(\boldsymbol{\theta}) =& \{\boldsymbol{\theta}^*; \exists k, k' \text{ such that } \theta_j^* = \theta_j \text{ for all } j \neq k, k',\ \theta_k^* \neq \theta_k \text{ and } \theta_{k'}^* \neq \theta_{k'}\}\\
\mathcal{N}_3(\boldsymbol{\theta}) =& \{\boldsymbol{\theta}^*; \exists k, k' \text{ such that } \theta_j^* = \theta_j \text{ for all } j \neq k, k',\ \theta_{k'}^* = \theta_k \text{ and } \theta_k^* = \theta_{k'}\}\\
\mathcal{N}_4(\boldsymbol{\theta}) =& \mathcal{N}_1(\boldsymbol{\theta}) \cup \mathcal{N}_3(\boldsymbol{\theta})
\end{aligned}$$

In each case, answer the following questions:

- What are the sizes of the neighborhoods?
- Do all solutions in Θ communicate?
- If the solutions communicate, what is the maximum of the number of moves needed to move between two arbitrary solutions?

(c). Consider a case where $p = 3$ and we want to maximize $f(\boldsymbol{\theta})$ where

$\boldsymbol{\theta}$	$f(\boldsymbol{\theta})$
000	0.5
001	0.2
010	0.2
100	0.1
011	2.0
101	2.1
110	0.5
111	1.5

Which of the methods covered in chapter 3 in the book (steepest ascent, simulated annealing, genetic algorithms, tabu algorithms) will be able to find the global optimum based on the first three different neighborhoods?

Discuss the pros and cons for the different neighborhoods.

Exercise 5 (Jensen's inequality)
Assume $\phi(\cdot)$ is a convex function and $g(\cdot)$ is a real-valued function with $\int g(x)dx < \infty$. Jensen's inequality is then that

$$\phi(\int g(x)dx) \leq \int \phi(g(x))dx.$$

(a). In statistics we often work with concave functions. Show that if $\phi(\cdot)$ is a concave function, then

$$\phi(\int g(x)dx) \geq \int \phi(g(x))dx.$$

(b). Assume now $f(x)$ is a density function for a continuous variable with cumulative distribution function $F(x) = \int_{-\infty}^{x} f(u)du$. Show that for $\phi(\cdot)$ concave we have

$$\phi(\int g(x)f(x)dx) \geq \int \phi(g(x))f(x)dx.$$

Express this result through expectations.

Hint: Define $y = F(x)$ and perform a reparametrization.

(c). Assume X follows the log-normal distribution. Show by Jensen's inequality that

$$E(X) \geq \exp[E(\log(X)]$$

Does this fit with the actual expectation of $E(X)$?

Exercise 6
N animals are distributed into four categories: $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$ according to the genetic linkage model (multinomial distribution with cell probabilities)

$$(\theta/4, (1-\theta)/4, (1-\theta)/4, (2+\theta)/4).$$

(a). $N = 197$. What is the likelihood for the data $\boldsymbol{x} = (34, 18, 20, 125)$?

(b). $N = 20$. What is the likelihood for the data $\boldsymbol{x} = (5, 0, 1, 14)$?

(c). For $\boldsymbol{x} = (34, 18, 20, 125)$:

(i) Use the Newton-Raphson algorithm to obtain the MLE ($\hat{\theta}$) of θ.

(ii) How did you assess convergence of the algorithm?

(iii) Compute the standard error for $\hat{\theta}$.

(iv) Plot the normalized likelihood and the associated normal approximation in the same figure. Discuss the adequacy of the normal approximation.

(v) Consider now the EM-algorithm. Define the complete data to be $\boldsymbol{y} = (y_1, y_2, y_3, y_4, y_5)$ where $y_j = x_j, j = 1, 2, 3$ while $y_4 + y_5 = x_4$. We now assume a multinomial model for the 5 variables with probabilities

$$(\theta/4, (1-\theta)/4, (1-\theta)/4, 1/2, \theta/4).$$

Construct and implement an EM-algorithm in this case.

(vi) Use bootstrapping to derive the uncertainty of $\hat{\theta}$ based on the EM-algorithm.

(d). Repeat (c) for $\boldsymbol{x} = (5, 0, 1, 14)$.

Exercise 7 (Mixture of Gaussians)
Assume $\boldsymbol{Y}_i = (X_i, C_i)$ are distributed according to

$$\begin{aligned}\Pr(C_i = k) =&\pi_k, & k = 1, ..., K\\ X_i|C_i = k \sim&N(\mu_k, \sigma_k^2)\end{aligned}$$

but where the C_i's are missing. The complete log-density for a single observation $\boldsymbol{y}_i$ is given by

$$\begin{aligned}\log f(\boldsymbol{y}_i) =&\log(\pi_{c_i}) + \log[\phi(x_i; \mu_{c_i}, \sigma_{c_i}^2)]\\ =&\sum_{k=1}^{K} I(c_i = k)[\log(\pi_k) + \log[\phi(x_i; \mu_k, \sigma_k^2)]]\end{aligned}$$

while the complete log-likelihood:

$$\log f_Y(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} I(c_i = k)[\log(\pi_k) + \log[\phi(x_i; \mu_k, \sigma_k^2)]$$

The E-step (taking into account that the C_i's are the only stochastic parts) gives

$$\begin{aligned}Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) =&E\{\sum_{i=1}^{n}\sum_{k=1}^{K} I(C_i = k)[\log(\pi_k) + \log[\phi(x_i; \mu_k, \sigma_k^2)]|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}\}\\ =&\sum_{i=1}^{n}\sum_{k=1}^{K} \Pr(C_i = k|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})[\log(\pi_k) + \log[\phi(x_i; \mu_k, \sigma_k^2)]\end{aligned}$$

Show that the M-step in the EM algorithm corresponds to

$$\begin{aligned}\pi_k^{(t+1)} =&\frac{1}{n}\sum_{i=1}^{n} \Pr(C_i = k|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})\\ \mu_k^{(t+1)} =&\frac{1}{n\pi_k^{(t+1)}}\sum_{i=1}^{n} \Pr(C_i = k|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})x_i\\ (\sigma_k^2)^{(t+1)} =&\frac{1}{n\pi_k^{(t+1)}}\sum_{i=1}^{n} \Pr(C_i = k|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})(x_i - \mu_k^{(t+1)})^2\end{aligned}$$

Exercise 8 (Variance estimation in the EM algorithm)
Consider the problem in Exercise 4.3 in Givens and Hoeting (2012). We will see how we can use Bootstrapping in order to obtain uncertainty estimates as well as point estimates

(a). Using the R-script for Exercise 4.3, write a function that has as input the data and some initial estimate of the parameters and returns the ML estimates.

(b). Write a script that generates $B = 1000$ bootstrap samples by sampling with replacement from the $n = 48$ observations. Run the script and obtain

- Variance estimates for $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$.
- Bias estimates for $\boldsymbol{\mu}$.
- 95% confidence intervals for $\boldsymbol{\mu}$.

Hint: Use the simple percentile intervals, that is the intervals obtained by finding the 0.025 and 0.975 empirical kvantiles among the bootstrap simulated estimates.

(c). For the bootstrap procedure above, do it take into account

- wrong model specifications?
- the actual amount of missing data?

(d). Consider now a parametric bootstrap approach, where first you generate $\boldsymbol{y}_i \sim N(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ for $i = 1, ..., n$ and then put those observations to missing that are missing in the original data set. Run this procedure and obtain

- Variance estimates for $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$.
- Bias estimates for $\boldsymbol{\mu}$.
- 95% confidence intervals for $\boldsymbol{\mu}$.

Comment on possible differences from the non-parametric version.

(e). For the parametric bootstrap procedure, do it take into account

- wrong model specifications?
- the actual amount of missing data?

Exercise 9 (Preliminaries for the stochastic gradient algorithm)
We will in this exercise shows some preliminary results that will be used in deriving properties of the stochastic gradient algorithm.

(a). Assume $\{a_t\}$ is a series of finite and non-negative numbers such that

$$\sum_{t=1}^{\infty} a_t = \infty.$$

Show that $\sum_{t=T}^{\infty} a_t = \infty$ for any $T \geq 1$.

(b). Assume $\alpha_t > 0$ and

$$\sum_{t=2}^{\infty} \frac{\alpha_t}{\alpha_1 + \cdots + \alpha_{t-1}} = \infty$$

Show that then $\sum_{t=1}^{\infty} \alpha_t = \infty$.

(c). Assume $\{a_t\}$ and $\{b_t\}$ are two series of finite and non-negative numbers such that $\lim_{t\to\infty} b_t$ exists and

$$\sum_{t=1}^{\infty} a_t = \infty, \quad \sum_{t=1}^{\infty} a_t b_t < \infty$$

Show that then $\lim_{t\to\infty} b_t = 0$.

(d). Consider a sequence

$$\theta^{t+1} = \theta^t - \alpha_t Z(\theta^t, \xi^t)$$

where $|Z(\theta^t, \xi^t)| < C$ with probability one. Show that

$$|\theta^t - \theta^*| \leq A_t \equiv |\theta^1 - \theta^*| + C(\alpha_1 + \cdots + \alpha_{t-1}).$$

Exercise 10 (Empirical information)
Assume $X \sim f(x;\theta)$ where we for simplicity assume both X and θ are univariate. Let $L(\theta;x) = f(x;\theta)$, $\ell(\theta;x) = \log f(x;\theta)$ and $s(\theta) = \ell'(\theta)$ be the likelihood, log-likelihood and scoring function, respectively. In the following, all the derivatives are with respect to θ.

(a). Assuming that differentiation and integration can be interchanged, show that

$$E[s(\theta;X)] = 0.$$

(b). Show that

$$E[l''(\theta;X] = -E[\tfrac{L'(\theta;X)^2}{L(\theta;X)^2}]$$

Use this to show that

$$I(\theta) = -E[l''(\theta;X)] = \text{Var}[s(\theta;X)] \qquad (*)$$

(c). Assume now $X_1, ..., X_n \overset{iid}{\sim} f(x;\theta)$. Show that the expected information based on all observations is given by

$$I_n(\theta) = n\text{Var}[s(\theta;X)]$$

where the last term is the variance based on a single observation.

(d). Consider a case where $X_i \sim f(x;\theta)$ where

$$f(x;\theta) = \tfrac{\Gamma(0.5(\nu+1))}{\sqrt{\nu\pi}\Gamma(0.5\nu)}(1 + \nu^{-1}(x-\theta)^2)^{-0.5(\nu+1)}$$

which is the T-distribution centered at θ with ν degrees of freedom.

Derive analytically $J(\theta;X)$, the observed information. For a simulated value of X, plot this as a function of θ in the range $[-5,5]$. Assume you have many realizations of X: $X_1, ..., X_n$, how can you use $J(\theta;X_i)$ to estimate $I(\theta)$?

(e). Assume $\theta^* = 0$. Simulate $X_1, ..., X_{1000}$ from $f(x;\theta^*)$. Use (*) to estimate $I(\theta)$ for values of θ in the range $[-5, 5]$. Plot these estimates together with the estimate of $I(\theta)$ from (d).

Repeat your simulations a few times.

Hint: In R the routine ***rt*** performs simulation from the T distribution centered at zero.

(f). Repeat (e) with $\theta^* = 3$.

(g). Comment on the results.

Exercise 11 (Stochastic gradient and neural network) (a). Take the ***Stoc_grad_NN.R*** script from the course web page. Modify the script to $n = 10\,000$. Run the script and inspect how the algorithm perform. Repeat the algorithm several times and look at the variability of the results.

Hint: You probably have to modify the script a bit, perhaps divide it into two. For the first part, the simulation of data, you want to keep that fixed (either by storing the data or by be sure that you use the same seed every time. However, for the run of the SG algorithm you should vary the seed!

(b). Try to modify the size of the subsample and look at the performance.

(c). Try to modify the learning rate $\gamma_t = c/(1+t)^\delta$ with $\delta \in [0.5, 1]$. How do the algorithm perform for different choices?

(d). Try to change the initial values of the β's to be completely random as well. How do the algorithm perform then?

Exercise 12 (SG and Geostatistics) (a). Consider the ***Geostat_SG.R*** script from the course web page. Try out different values of m and evaluate the performance of the algorithm.

(b). Include a call to the ***geoR*** package (last commands in the script) which should give the optimal solution. Does the results from the SG algorithm look reasonable?

(c). Try to increase n by doubling it and run the routine from the ***geoR*** package. How many times are you able to double it before the routines take too much time to produce results?

(d). Now try to increase n to 1000 and 10 000. Try out different values of m and learningrates to obtain as good performance as possible.

Exercise 13 (Inverse transform sampling)
Assume $U \sim \text{Uniform}[0, 1]$.

(a). Let $f(x)$ be a density function with corresponding cumulative distribution function $F(x)$ for which F^{-1} exists. Show that

$$X = F^{-1}(U) \sim f(x).$$

This way of simulation random variables is called the inverse transform sampling.

(b). Assume $f(x)$ is the Cauchy distribution given by

$$f(x) = \frac{1}{\pi\gamma\left[1+\left(\frac{x-x_0}{\gamma}\right)^2\right]}, \quad F(x) = \frac{1}{\pi}\tan^{-1}\left(\frac{x-x_0}{\gamma}\right) + \frac{1}{2}$$

Use the inverse transform sampling method to simulate 1000 variables from the Cauchy distribution. Calculate the estimated density from your samples together with the true density and confirm that the method works.

Hint: The R command ***runif(1000)*** generate 1000 uniform variables. The R command ***density(x)*** can be used to estimate a density based on the (vector of) observations ***x***.

Exercise 14 (Generating variables from the normal distribution)
Recall that if $\boldsymbol{Y}$ is a random vector and $X = \boldsymbol{g}(\boldsymbol{Y})$ with $\boldsymbol{Y} = g^{-1}(\boldsymbol{X})$, then

$$f_X(\boldsymbol{x}) = f_Y(\boldsymbol{g}^{-1}(\boldsymbol{x}))\left|\frac{\partial}{\partial \boldsymbol{x}} g^{-1}(\boldsymbol{x})\right|$$

(a). Assume $\boldsymbol{U} = (U_1, U_2)$ where U_1 and U_2 are independent and Unif$[0,1]$. Define

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2), \quad X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

Show that

$$\begin{aligned} U_1 =& e^{-0.5(X_1^2+X_2^2)} \\ U_2 =& \frac{1}{2\pi}\tan^{-1}(X_2/X_1) \end{aligned}$$

(b). Show that X_1, X_2 are independent and $N(0,1)$.

(c). Show how you then can generate from a general $N(\mu, \sigma^2)$ distribution.

(d). Assume now you want to simulate from a multivariate distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. How can you do that?

Exercise 15 (Checking random generators)
Random generators on computers are never completely random. Typically, one generated number is a function of the previous (or several previous) numbers.

There are many kinds of tests (formal and visual) for checking whether a random generator is good. One such is to test for independence between successive generated numbers.

Generate 10 000 variables $U_1, ..., U_{10000}$ from the uniform distribution. Plot U_i against U_{i+1}. Is there any indication of dependence?

Hint: If `u` contains the 10 000 variables, then `u[-1]` will contain the same data except the first one, while `u[-10000]` will contain the same data except the 10 000th one.

Exercise 16 (Calculation by Monte Carlo)

(a). Simulate 10000 samples from the $N(2, 1)$ distribution and store them in a vector called `z`.

Hint: The R command `rnorm(m,2,1)` simulates `m` random variables from the $N(2, 1)$-distribution.

(b). Plot the estimated density and the true density in the same figure.

Hint: The R command

```
> d <- density(z)
```

will estimate the density by a non-parametric kernel method. `d` will be a list containing two components with names `x` and `y`. `d$x` contain the values at which the density is estimated, while `d$y` are the corresponding estimates. Plot both curves in the same figure by

```
> matplot(cbind(d$x,d$x),cbind(dnorm(d$x,2,1),d$y),type="l")
```

(c). Estimate mean and variance by standard estimates from the sample. Compare with the true values.

(d). Assume now $Z \sim N(2, 1)$. For each of the two sub-exercises below, measure the time you spend on the problem:

(i) Find $\nu = E[Z|Z > 3.0]$ trough analytical calculation.

(ii) Find $\nu = E[Z|Z > 3.0]$ through simulation.

How many observations was the simulated estimate based on?

Hint: The R command `trunc <- z[z>3.0]` picks the set of z's larger than 3.0 from the vector `z`. The length of the vector `trunc` can be found by the command `length(trunc)`.

(e). Repeat $(a) - (d)$ with the following distributions: Exp(2), T-distribution with 4 degrees of freedom and Gamma(2) (calculate ν analytically only when it is not too difficult).

Exercise 17 (The level accuracy of t-test)

The perhaps most commonly applied statistical method of all is the t-test. We shall in this exercise examine how the level accuracy of this test depends on the underlying distribution. There are studies available in the statistical literature (with different conclusions!), but it is usually quicker to investigate the issue oneself using a modern software package.

Let $X_1, ..., X_n$ be an i.i.d. sample with mean μ and consider the hypothesis $H : \mu = 0$. The standard test is to reject when $|T| > c_0$ where

$$T = \frac{\sqrt{n}\overline{x}}{s}$$

and $\overline{x}$ and s are the mean and standard deviation respectively. c_0 is the $\alpha/2$ percentile of the t-distribution with $n - 1$ degrees of freedom for level α.

(a). Formulate the significance level as an expectation and use this to explain how you can estimate the significance level through Monte Carlo methods.

(b). Run 10000 repetitions to estimate the level for normal data when $n = 15$.

(c). Repeat (a) with exponential data and with a gamma distribution that is the sum of three exponentials. Translate the data so that expectation is equal to zero in each case.

(d). Repeat (a) and (b) when $n = 50$.

(e). Summarize your findings. What is the Monte Carlo variability in the estimates? Do you find the t-test robust? What is the relevance of your results for t confidence intervals?

Exercise 18 (Piecewise linear densities and the inversion method)

Assume $g(x)$ is a density of the form

$$g(x) = c\exp\{a_i + b_i x\} \qquad \text{for } x \in (z_{i-1}, z_i]$$

where $z_0 = -\infty$, $z_{k+1} = \infty$. and $b_1 > 0$ while $b_{k+1} < 0$.

(a). Show that

$$G_i = \int_{z_{i-1}}^{z_i} g(x)dx = \frac{c}{b_i}\exp(a_i)[\exp(b_i z_i) - \exp(b_i z_{i-1})]$$

and use this to find c. Aslo use the derivations to explain the constraints $b_1 < 0$ and $b_{k+1} < 0$.

(*b*). Show that

$$G(x) = \sum_{i=1}^{j-1} G_i + \frac{c}{b_j} \exp(a_j)[\exp(b_j x) - \exp(b_j z_{j-1})]$$

for $x \in (z_{j-1}, z_j]$.

(*c*). Show that for $\sum_{i=1}^{j-1} G_i < u \leq \sum_{i=1}^{j} G_i$ we have

$$G^{-1}(u) = z_{j-1} + \frac{1}{b_j} \log[1 + \frac{b_j}{c} \exp(-a_j - b_j z_{j-1})(u - \sum_{i=1}^{j-1} G_i)].$$

Show that this value is always in the interval $(z_{j-1}, z_j]$.

(*d*). Use the results above to describe an algorithm for simulating from $g(x)$ and relate this to the ARS algorithm.

Exercise 19 (Random variation in histograms and density estimates)
In this exercise you will need the commands `rnorm` and `rexp`. See `help(rexp)` for guidance on how to use `rexp`.

(*a*). Sample $n = 10$ N(1,1) variables and draw a histogram. Repeat $m = 4$ times and get a feeling for the random variation in histogram. Store the data.

(*b*). Repeat (*a*), but now use exponentials (subtract one so that the random variables have 0 means). Compare the shape of the histograms with those in (*a*). Are some of the shapes similar?

(*c*). Repeat (*a*) and (*b*) with $n = 100$.

Many statisticians prefer to use density estimates rather than histograms. Assuming `x` is a vector of random variables, a density estimate can be plotted by the command

```
plot(density(x))
```

(*d*). Make density plots of all the data you have generated for $n = 10$. Compare thoese corresponding to the normal with those corresponding to the exponential distribution. Is it easy to see differences?

(*e*). Repeat (*d*), but now with $n = 100$.

Exercise 20 (Testing rejection sampling)
The example in this exercise is (deceptively) simple. Nobody in their right mind would use rejection sampling to solve it. Yet it is sufficient to provide insight into a major issue with rejection algorithms. Besides it is always convenient to know answers theoretically when testing out a general method for use in problems that do not allow simple solutions.

Suppose you originally know that a random variable X of interest is normal $(0, 1)$. Further information is available through a measurement or an observation z, which we assume is the outcome of a random variable Z with conditional density $f(z|x)$ given $X = x$. We will further assume that $f(z|x) < A$ for all x, z and some constant $A < \infty$.

(a). Show that

$$p(x|z) = Cf(z|x)\text{exp}(-\frac{1}{2}x^2)$$

is the posterior density of X given z (C is a normalization constant).

Suppose we want to sample from this posterior distribution.

(b). Write down an algorithm based on rejection from the normal $(0, 1)$ (it is assumed that $f(z|x)$ is a computable function).

Let N be the number of attempts until acceptance.

(c). Calculate $E(N)$ when the conditional distribution of Z given $X = x$ is normal (ax, σ^2). Discuss the behavior of $E(N)$ as a function of z.

(d). Write a program which carries out the rejection sampling. Keep track on the number of trials before acceptance. ($f(x|z)$ may be a general function, depending on x, or the normal one introduced above).

(e). Run the program a suitable number (m) times to generate m simulations of X from the distribution $p(x|z)$. Use $a = 0.5$ and $\sigma^2 = 0.1$. Let $z \geq 0$. Start at 0 and increase it in steps of 0.5. (Use at least 100 repetitions).

(f). Compute the mean of N for the simulations. Compare with the theoretical values.

(g). Compute mean and variance of the simulations of X. Again compare with the theoretical values. (You must calculate $E(X|z)$ and $\text{var}(X|z)$ theoretically. Use theory for conditional Gaussian distributions.)

(h). Try to make some general conclusions from this exercise. If your algorithm took a long time to converge in some of the examples, how could it have been improved?

Exercise 21 (The impact of serial correlation in regression)
It is in economics and in many industrial applications of statistics quite common that error terms are serially correlated (also called auto-correlated). We shall in this exercise examine the effect of this on a simple regression method where such serial correlations are ignored. Consider the regression model

$$Y_t = \alpha + \beta x_t + \varepsilon_t, \quad t = 1, ..., n,$$

where α and β are the regression coefficients and ε_t the error terms. As basis for the study take the standard least-squares estimate of β, i.e.

$$\hat{\beta} = \sum_{i=1}^{n}(x_i - \overline{x})y_i / \sum_{i=1}^{n}(x_i - \overline{x})^2.$$

As model for ε_t take the AR(1) model

$$\varepsilon_t = a\varepsilon_{t-1} + \eta_t$$

where $\eta_1, ..., \eta_n$ is an i.i.d. sequence with mean 0 and variance σ_η^2. It can then be proved that

$$\mathrm{var}(\varepsilon_t) = \frac{\sigma_\eta^2}{1 - a^2}$$

and that

$$\mathrm{corr}(\varepsilon_t, \varepsilon_{t-s}) = a^{|t-s|},$$

assuming the model to be in stationary state. Although it is not hard to analyze the impact on the serial correlation analytically, we shall in the following run simulations. Let $n = 20$.

(a). Start by drawing $x_1, ..., x_{20}$ from the uniform distribution over $[-1, 1]$.

Let $x_1, ..., x_{20}$ be fixed throughout and take $\alpha = 0$ and $\beta = 2$ for the true model.

(b). Compute 100 copies of the error terms by sampling $\varepsilon_1, ..., \varepsilon_n$. Use $a = -0.9, -0.5, 0, 0.5, 0.8$ and 0.9 and employ the trick of common random numbers (which means that you use the same η-sequence). Arrange things so that $\sigma_\varepsilon = 1$ is the same value for all choices of a.

Hint: In order to generate the same sequence of random numbers several times, you can use the same seed number for each generation. The seed can be set to a given number `N` by the command

```
set.seed(N)
```

The command

```
eps <- arima.sim(20,model=list(ar=a),sd=sigma.eta)
```

simulates an AR(1) time-series of length 20 with $a =$ `a` and $\sigma_\eta =$ `sigma.eta`.

(c). Estimate bias and variability of $\hat{\beta}$ as a function of a simulation. Discuss your results and compare with the bias and variability you nominally have from the elementary least-squares theory that does not take serial correlations into account.

(d). Repeat the study in (c) but now use the standard estimate $\hat{\sigma}$ of the σ instead of $\hat{\beta}$.

(e). What is the impact of the results you have found for standard hypothesis tests for $\beta = \beta_0$ and confidence limits for β?

Exercise 22 (Importance sampling)
Let $\phi(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ be the standard normal density. Consider the integral

$$J = \int_{-\infty}^{\infty} (x+a)^2 \phi(x)\, dx = 1 + a^2.$$

(a). Compute J by Monte Carlo sampling from the standard normal. Use 100 and 1000 simulations and let $a = 0, 1, 2, 3, 4$.

(b). Calculate the theoretical standard deviation for the estimates and compare with the error of the actual values computed in (a). Note the dependency on a.

(c). Try importance sampling based om $g(x) = \phi(x - a)$. Redo (a) and (b).

(d). Formulate general conclusions. Note that g depends on a.

Exercise 23 (The extra Monte Carlo uncertainty)
Consider the Bayesian situation with an unknown parameter θ which has an a posteriori distribution conditional on the observations $\boldsymbol{x}$ given by

$$\theta \sim p(\theta|\boldsymbol{x}).$$

A reasonable estimate for θ is the a posteriori expectation $\hat{\theta} = E[\theta|\boldsymbol{x}]$.

Suppose the expectation is difficult to calculate analytically, but that we are able to simulate from the posterior distribution. Let $\theta_1^*, ..., \theta_m^*$ be a sample from $p(\theta|\boldsymbol{x})$. An alternative estimate for θ is then

$$\overline{\theta}^* = \frac{1}{m}\sum_{i=1}^{m} \theta_i^*$$

Show that

$$\begin{aligned} E[\overline{\theta}^*|\boldsymbol{x}] &= \hat{\theta}; \\ E[(\overline{\theta}^* - \theta)^2|\boldsymbol{x}] &= \left(1 + \frac{1}{m}\right) E[(\hat{\theta} - \theta)^2|\boldsymbol{x}]. \end{aligned}$$

Discuss this result.

Exercise 24 (Sampling by guide tables)
Let $P(X = r) = p_r$, $r = 1, 2, ..., M$ and $P_r = P(X \leq r)$. We shall in this exercise consider sampling from this distribution by the method of so-called guide tables. Define

$$g_i = \max\{j | P_j < i/M\},$$

which is known as the guide table. Consider the algorithm

```
1. Sample $U \sim$ uniform(0,1)
2. $X = \text{int}(MU + 1)$
3. $Y = g_X + 1$
4. Repeat
        If $(P_{Y-1} > U)$ then $Y = Y - 1$ else go to 5
5. Return $X = Y$.
```

(a). Show that the algorithm is correct.

(b). Show that the expected number of repetitions in step 4. is at most 2.

(c). Discuss how this procedure can be utilized in the most time-consuming part of the ARS algorithm.

Exercise 25 (Guide tables and continuous distributions)
Let $X = (X_1, ..., X_n)'$ be a random vector with density $f(x) = f(x_1, ..., x_n)$.

(a). Explain how the method of guide tables in Exercise 24 can be used to obtain approximate simulations of X. What n can be handled in practice?

(b). In many applications $f(x) = ch(x)$, where h is a computable expression, and c is a normalization constant which can only be found through numerical integration. Can the algorithm in (a) do without knowledge of c?

Exercise 26 (Sequential importance sampling)
In population ecology, variations of the population sizes for a specific animal is measured through time-series observations on the number of animals caught in traps. Assume y_t is the number of animals caught at time t (the time-scale is typically in years).

A simple model in this case (defining x_t to be the logarithm of the population size) is

$$\begin{aligned} x_1 &\sim N(\mu, \sigma^2/(1-a^2)) \\ x_t &\sim N(\mu + a(x_{t-1} - \mu), \sigma^2) \\ y_t &\sim \text{Poisson}(\exp\{x_t\}) \end{aligned}$$

The following data (which is also given on the web-page under the name `sim_animal_trap.txt`) are data simulated from the model above using $\mu = 2, a = 0.9, \sigma = 0.5$. The first row corresponds to the first 18 time-points and so on.

2	7	8	4	7	7	7	8	11	10	8	4	8	9	19	12	35	39
14	5	6	5	6	1	0	2	1	3	6	4	1	0	2	3	1	2
0	1	3	0	3	9	4	13	23	15	7	9	10	6	3	12	16	29
28	18	13	6	8	14	25	14	17	11	19	39	55	71	83	61	60	44
57	26	24	47	20	53	65	68	56	48	26	23	29	17	2	30	24	52
27	20	13	13	18	19	5	4	10	9								

(a). Show that $x_t \sim N(\mu, \sigma^2/(1-a^2))$ for all t. Discuss this property.

(b). Write down the posterior (or conditional) distribution for $\boldsymbol{x}_{1:T} = (x_1, ..., x_T)$ given $\boldsymbol{y}_{1:T} = (y_1, ..., y_T)$ (up to a proportionality constant).

The posterior distribution for $\boldsymbol{x}_{1:N}$ is difficult to simulate from directly. We will therefore consider a sequential importance sampling (SIS) algorithm in this case.

(c). Consider first a case where $\boldsymbol{x}_{1:T}$ is sampled from the prior, that is

$$g(\boldsymbol{x}_{1:t}) = g_1(x_1) \prod_{s=2}^{t} g_s(x_s|x_{s-1})$$

with $g_1(x_1) = N(\mu, \sigma^2/(1-a^2))$ and $g_s(x_s|x_{s-1}) = N(\mu + a(x_{s-1} - \mu), \sigma^2)$. Calculate the importance weight in this case and show that it can be written recursively as

$$w_t(\boldsymbol{x}_{1:t}) = w_{t-1}(\boldsymbol{x}_{1:t-1}) u_t(y_t, x_t)$$

for properly defined functions $w_1(x_1)$ and $u_t(y_t, x_t)$. (Here $\boldsymbol{x}_{1:t} = (x_1, ..., x_t)$.)

(d). Implement a Sequential Monte Carlo algorithm based on the previously results and try it out on the data given above. Use $N = 10\,000$ and resampling at each time-point.

For each t, estimate $\hat{x}_{t|t} = E[x_t|\boldsymbol{y}_{1:t}]$ and also estimate the 0.025 and 0.975 quantiles in the distribution $p(x_t|\boldsymbol{y}_{1:t})$. Plot these estimates and quantiles in the same plot.

Also calculate the effective sample size just before you do resampling. Plot this as a function of time.

Hint: Modify one of the R-scripts with Sequential Monte Carlo from the web-page.

(e). Assume our interest now is on $\hat{x}_{t|T} = E[x_t|\boldsymbol{y}_{1:T}]$, that is the state estimates based on all data. Explain how these estimates can be obtained from your Sequential Monte Carlo algorithm. Plot these estimates (together with 0.025 and 0.975 quantiles in the distribution $p(x_t|\boldsymbol{y}_{1:T})$ on top on those you plotted in (d). Discuss similarities and differences.

Also look at the number of unique values that the approximation of $\hat{x}_{t|T} = E[x_t|\boldsymbol{y}_{1:T}]$ is based on as a function of time.

Exercise 27 (Improvements of SIS)
We will in this exercise consider the same problem as the one in exercise 26, but now see if we are able to imrpove the algorithm described there by using better proposal distributions. The main idea is that the simple proposal used in exercise 26 do not take the data into account at all, and that using the data should help us in simulating more reasonable x's.

(a). Consider first a simple situation where $x \sim N(\mu, \sigma^2)$ and $y \sim \text{Poisson}(\exp\{x\})$.

Write down the posterior distribution for x given y, $p(x|y)$ (up to a proportionality constant).

(b). Consider the logarithm of $p(x|y)$, and assume we want to approximate this by a function of the form $\text{Const} - \frac{1}{2\tilde{\sigma}^2}(x - \tilde{\mu})^2$. What kind of distribution does this correspond to?

Argue why a reasonable approximation for e^x is

$$e^\mu + e^\mu(x - \mu) + \tfrac{1}{2}e^\mu(x - \mu)^2$$

in this case. Use this approximation to derive $\tilde{\mu}$ and $\tilde{\sigma}^2$.

(c). Consider now the setting of exercise 26. Use the approximation above to suggest a proposal distribution for x_t that is approximately $p(x_t|x_{t-1}, y_t)$.

Modify your algorithm to consider this proposal and run it on the given data.

(d). Use some measures to evaluate the performance of this modification compared to the simpler algorithm used in exercise 26. Which algorithm do you prefer?

Exercise 28 (Variance reduction)
Assume X has a Cauchy distribution, that is with density $f(x) = 1/[\pi(1 + x^2)]$. We want to estimate $\theta = \Pr(X > 2)$ (true value is 0.1476). We will consider four different estimates, each of the form $\hat{\theta} = \frac{1}{m}\phi(U_i)$ where $U_1, ..., U_m$ are random samples from some distribution. In each case, simulate $m = 10000$ samples for estimating θ. Use as an estimate for the variance the sample variance of $\phi(U_1), ..., \phi(U_m)$.

(a). Let $\phi(U) = I(U > 2)$ where U has a Cauchy distribution. Show that $E\phi(U) = \theta$. Estimate θ. What is the variance of the estimate?

Hint: The R command `phi <- as.integer(u>2)` gives $I(u > 2)$ for a vector `u`.

(b). Let $\phi(U) = \frac{1}{2}I(|U| > 2)$ where U has a Cauchy distribution. Show that $E\phi(U) = \theta$. Estimate θ. What is the variance of the estimate?

(c). Let $\phi(U) = \frac{1}{2} - 2f(U)$ where $U \sim U[0, 2]$. Show that $E\phi(U) = \theta$. Estimate θ. What is the variance of the estimate?

(d). Let $\phi(U) = \frac{1}{2}f(U)$ where $U \sim U[0, 1/2]$. Show that $E\phi(U) = \theta$ (hint: make a substitution $x = 1/u$). Estimate θ. What is the variance of the estimate?

Exercise 29 (Variance reduction)
Hammersley and Handscomb (1964) use the integration of $\phi(x) = (e^x - 1)/(e - 1)$ on $(0, 1)$ as a test problem of variance reduction techniques. Achieve as large a variance reduction as you can compared to the naive Monte Carlo integration based on uniform sampling on $(0, 1)$. (Hammersley and Handscomb achieved 4 million).

Exercise 30 (Common random numbers)
Let $\Psi(\theta) = E_\theta(X)$, where X is a random variable with distribution depending on θ. In the situations we have in mind this distribution is complicated, but $\Psi(\theta)$ can be approximated by sampling, so that

$$\hat{\Psi}_n(\theta) = \frac{1}{N}\sum_{j=1}^{N} X_j^* \qquad (1)$$

where $X_1^*, ..., X_n^*$ is an i.i.d. sample of X under θ.

(a). Write down the approximative distribution for the error $\hat{\Psi}_n(\theta) - \Psi(\theta)$ using $\sigma^2(\theta) = \mathrm{var}_\theta(X)$.

Frequently much interest is directed towards differences $\Psi(\theta_2) - \Psi(\theta_1)$ for two θ-values $\theta_1 < \theta_2$. This will be so if the derivative (or gradient) is to be estimated as part of some Monte Carlo numerical scheme. Another common situation is when we want to study differences in performance of some statistical procedure.

(b). Write down the approximative distribution of $\hat{\Psi}_n(\theta_1) - \hat{\Psi}_n(\theta_2)$ when $\hat{\Psi}_n(\theta_1)$ and $\hat{\Psi}_n(\theta_2)$ are obtained from different rounds of simulation.

When we sample X, we use a routine that can be written as a function

$$X = h(\theta, Z) \qquad (2)$$

where Z is a random vector, perhaps a long one, having distribution not dependent on θ. The function h is possible very complicated, and may not necessarily be available in analytical form. We assume here that it possesses the necessary smoothness.

Inserting (2) into (1) yields

$$\hat{\Psi}_n(\theta) = \frac{1}{N}\sum_{i=1}^{N} h(\theta, Z_j^*), \qquad (3)$$

where $Z_1^*, ..., Z_n^*$ is an i.i.d. sample from Z. Note that $\hat{\Psi}_n(\theta)$ in (3) for a given sequence $Z_1^*, ..., Z_n^*$ can be regarded as a deterministic function of θ. It is in practice possible to organize the computations so that the same sequence $Z_1, ..., Z_n^*$ goes into (3) for any value of θ by resetting the seed of the Monte Carlo generator.

(c). Determine the approximate distribution of $\widehat{\Psi}_n(\theta_2) - \widehat{\Psi}_n(\theta_1)$ when common random numbers are used.

Hint: Introduce $\rho(\theta_1, \theta_2) = \text{corr}\{h(\theta_2, Z^*), h(\theta_1, Z^*)\}$ and comment on how the magnitude of ρ influences the result.

Suppose that the difference $\theta_2 - \theta_1$ is fairly small so that

$$h(\theta_2, Z) \simeq h(\theta_1, Z) + \frac{\partial h(\theta_1, Z)}{\partial \theta}(\theta_2 - \theta_1).$$

(d). Use this approximation to find an alternative expression for the approximate distribution of $\widehat{\Psi}_n(\theta_2) - \hat{\Psi}_n(\theta_1)$.

(e). Compare the result on (d) with that in (b). Comment on the relevance for Monte Carlo based estimation of the derivative $\Psi'(\theta)$.

Exercise 31 (Auto-correlations from many time series)
Let $Y_1, ..., Y_n$ be a time series which is assumed to be stationary, which in particular mean, that

$$\rho(u) = \text{corr}\{Y_t, Y_{t+u}\}$$

does not depend on t. The standard estimate of $\rho(u)$ is to compute the correlation coefficient at the $n - u$ pairs $(Y_1, Y_{1+u}), (Y_2, Y_{2+u}), ..., (Y_{n-u}, Y_n)$.

Suppose we have 100 independent time series, which are assumed to have been generated by the same mechanism. Each of them is quite short, say $n = 10$. Let $\hat{\rho}_j(u)$ be the estimate of $\rho(u)$ for series j. It is tempting to convert the 100 estimates into a common one by taking the average, i.e.

$$\hat{\rho}(u) = \frac{1}{100}\sum_{j=1}^{100} \hat{\rho}_j(u).$$

We shall in this exercise examine the soundness of this procedure.

(a). Generate 100 short time series according to the AR(1) model described in exercise 21. Use $a = 0.5$ and $a = 0.8$.

Hint: The R command

```
arima.sim(model=list(ar=0.5),n=10)
```

simulates an $AR(1)$ series of length 10 with $a = 0.5$.

(b). Compute $\hat{\rho}(u)$ for $u = 1, 2, 3$.

Hint: The R command

```
y <- acf(x,lag.max=3,plot=F)
```

computes the autocorrelation function for a univariate or multivariate time series `x`. `y$acf[u+1,j,j]` contains $\hat{\rho}_j(u)$.

(c). Repeat (b) a few times and compare with the true values $\rho(u) = a^u$. Comment on your results.

Exercise 32 (The effect of model selection)
The following example is adapted from Hjorth (1994). Consider the time series data given in the table.

x	-5.00	-4.00	-3.00	-2.00	-1.00	0.00	1.00	2.00	3.00	4.00	5.00
y	1.05	3.20	1.60	3.05	2.45	2.90	2.00	2.20	4.60	4.00	3.50

When you plot them (do it), it will become clear that there is considerable uncertainty as to whether there is an underlying consistent growth or not. Yet this issue is of crucial importance for forecasting. If there is evidence of such a trend, we may take the view that it is likely to continue. Consider two competing models:

$$\begin{aligned} M_0 : &\quad y_t = \beta_0 + \varepsilon_t, \quad t = -n, -n+1, ..., -1, 0, 1, ..., n \\ M_1 : &\quad y_t = \beta_0 + \beta_1 t + \varepsilon_t, \quad t = -n, -n+1, ..., -1, 0, 1, ..., n \end{aligned}$$

where the ε_t's are independent errors. In a situation like this you might like to select one of the models from the empirical evidence available and use it to forecast. The problem addressed in this exercise is to what extent the model selection influence the bias and variability of the forecast. The traditional way in statistics is to ignore the issue completely and proceed with standard theory as if no data-driven selection had taken place at all. How wrong is this exactly?

As an example, consider a selection rule based on statistical significance. Let $\hat{\beta}_1$ be the least squares estimate of β, and s_1 its standard error. Proclaim M_1 if the hypothesis $H : \beta_1 = 0$ is rejected. This means that the procedure for predicting $\theta = E(y_{t_0})$ is

$$\hat{\theta} = \begin{cases} \overline{y}, & \text{if } \frac{|\hat{\beta}_1|}{s_1} < t_{\alpha/2} \\ \hat{\beta}_0 + \hat{\beta}_1 t_0, & \text{if } \frac{|\hat{\beta}_1|}{s_1} \geq t_{\alpha/2} \end{cases}$$

Here $\overline{y}$ is the mean of the observations, $(\hat{\beta}_0, \hat{\beta}_1)$ is the least squares estimates under M_1, s_1 is the estimated standard error of $\hat{\beta}_1$ and t_α is the α percentile of the t-distribution with $2n - 1$ degrees of freedom.

(a). For $t_0 = 5$, estimate $E\hat{\theta}$ and $sd(\hat{\theta})$ as a function of β_1 by running the following simulation experiment. Let $n = 5$, ε_t Gaussian distributed with $\sigma^2 = \text{var}(\varepsilon_t) = 1$, make your own choice of β_0 and let β_1 vary from, say -1.0 to 1.0 at step 0.05 (or according to another scheme if you so prefer). Let $\alpha = 0.05$. For each set of parameters, simulate data, compute $\hat{\theta}$ and repeat the number of times you find necessary. Register for each choice of β_1 in how many of the simulations model 1 was chosen.

(b). Plot a density plot of θ for $\beta_1 = 0.0, 0.2, 0.3, 0.6$.

(c). Repeat (a) for $t_0 = 8$.

(d). Compare results in (a) and (b) with those found by standard theory that ignores the data-dependent model selection that took place.

Exercise 33 (Importance resampling)
We shall in this exercise test the importance resampling algorithm on the same example as in Exercise 22. The objective is now sampling of random variables rather than approximate calculation of expectations. The simulated, importance resampling (SIR) algorithm, due to Rubin, is in general form as follows. Suppose we want a sample from some awkward distribution with density f. We assume that f can be computed up to a constant. Choose a more convenient distribution with density g. Draw a sample $X_1, ..., X_M$ from g. This is the first step. The second start by computing

$$w_i = \frac{f(X_i)}{g(X_i)}, \qquad i = 1, 2, .., M,$$

and

$$q_i = \frac{w_i}{\sum_j w_j}, \qquad i = 1, 2, ..., M.$$

Let Y be a random variable, defined conditionally of $X_1, ..., X_M$ with distribution

$$P(Y = X_i | X_1, ..., X_M) = q_i, \qquad i = 1, 2, ..., M.$$

Draw m samples of Y. It can then be proved that as $M \to \infty$, a sample of m independent variables from f appears in the limit.

Suppose $f(x) = \phi(x - a)$, where $\phi(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ is the standard normal density. We shall test the efficiency of the SIR algorithm when sampling from $g(x) = \phi(x)$. Use $m = 100$.

(a). First let $a = 1$. Vary M from 1000 and upwards. Try to find out how large M must be. You must compare the mean and the variance of the final sample to their known values. Q-Q plotting might be a good idea, and you must also worry about how large M must be in order to make the final sample an independent one. Try to think of a way to measure dependence.

(b). Repeat (a) for $a = 2, 3, 4$.

(c). Formulate general conclusions.

(d). Why can we do without the normalization constant in f?

Exercise 34 (Bootstrapping after model selection)
This is a companion to exercise 32. We will consider the same dataset as in that exercise. The data is stored in R in the dataset `model.select`.

Again there is a considerable doubt as to the underlying model, as is apparent when the data is scatter-plotted. Consider two models

$$\begin{aligned} M_0: &\quad y_t = \beta_0 + \varepsilon_t, \quad t = 1, 2, ..., n \\ M_1: &\quad y_t = \beta_0 + \beta_1 t + \varepsilon_t, \quad t = 1, 2, ..., n \end{aligned}$$

where model M_0 clearly is a special case of model M_1.

As predictor for $\theta = E[y_{t_0}]$ we suggest an adaptive procedure, i.e.

$$\hat{\theta} = \begin{cases} \overline{y}, & \text{if } \frac{|\hat{\beta}_1|}{s_1} < t_\alpha \\ \hat{\beta}_0 + \hat{\beta}_1 x, & \text{if } \frac{|\hat{\beta}_1|}{s_1} \geq t_\alpha \end{cases}$$

Here $\overline{y} = (y_1 + \cdots + y_n)/n$ is the mean of the observed observations, $\hat{\beta}_0$ and $\hat{\beta}_1$ are the least squares estimates of β_0 and β_1 under model M_1, s_1 is the estimated standard error of $\hat{\beta}_1$ and t_α a fractile in the t-distribution with $2n - 1$ degrees of freedom. We will in this exercise use Bootstrap methods for estimating the standard error of the estimate $\hat{\theta}$.

(a). Which model will be chosen for the given data-set with $\alpha = 0.05$? What is $\hat{\theta}$ for $t_0 = 5, 8$?

(b). Generate bootstrap samples by generating

$$y_t^* = \hat{\beta}_0 + \hat{\beta}_1 t + \varepsilon_t^*$$

where $\hat{\beta}_j$ are estimates obtain by model M_1 while $\{\varepsilon_t^*\}$ are resampled samples from the residuals obtained from model M_1. Report estimates of θ, standard errors and confidence intervals.

(c). Also try out bootstrapping when you resample the pairs (x, y).

(d). Discuss differences between these approaches and also compare with the measures you would have used if you only considered the chosen model.

Exercise 35 (Sampling by Metropolis-Hastings)
Assume we are interested in simulating from the bivariate Gaussian distribution $N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where

$$\boldsymbol{\mu} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}$$

This is of course a simple distribution for which we could perform simulation directly, but we will use it to illustrate the Metropolis-Hastings method.

(a). Implement a random walk Metropolis-Hastings algorithm were one component is changed at a time and the proposal distribution is Gaussian centered at the current value.

(b). Run the algorithm 1000 iterations for $a = 0$. Use simulations from the standard Gaussian distribution as starting points. Tune the standard deviations in the proposal distributions so that the acceptance rate become approximately 0.3.

Estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from your simulations. Use traceplots to see how many of the first samples you should discard.

Make a plot of your simulations in the two-dimensional space, drawing a line between each iteration.

Comment on the results.

(c). Now repeat the previous point with $a = 0.99$.

You probably need to tune the standard deviations in the proposal distributions again so that the acceptance rate become approximately 0.3.

Make similar estimates and plots and comment on the results.

Exercise 36 (Sampling random variables by Metropolis)
Let X be a random variable with density $f(x) = ch(x)$, where h is a computable expression. We shall in this exercise discuss sampling of X through the Metropolis algorithm. The aim is to demonstrate how the algorithm works, not to produce the most efficient sampling scheme conceivable.

Let $\varepsilon_1, \varepsilon_2, ..$ be a sequence of independently and identically distributed random variables. You may choose any distribution symmetrical about the origin. Suggested choice: The uniform over $(-a, a)$, where $a > 0$ has to be adapted for each application. In addition, let $U_1, U_2, ...$ be a sequence of uniforms over $(0, 1)$. Consider the recursion

$$X_{t+1} = X_t + \varepsilon_t I_t$$

where

$$I_t = 1, \qquad \text{if } U_t \leq \frac{h(X_t + \varepsilon_t)}{h(X_n)}$$

and $= 0$ otherwise.

(a). Show that this recursion is a special case of the Metropolis algorithm. Explain, in particular, the relevance of demanding ε_t to be a symmetric distribution.

(b). Suppose f is the standard normal density. Show that the ratio defining I_t becomes $\exp(-X_t\varepsilon_t - 0.5\varepsilon_t^2)$. I_t is more likely to become 1 if the signs of X_t and ε_t are alternate. Explain the meaning of this.

(c). Will you need all U_t to run the algorithm?

(d). Implement m parallel runs of the algorithm when f is the standard normal. Use, for example, $m = 100$. [Hint: Using the above description of the algorithm, you can implement this in vectorized form in R.]

(e). Discuss ways of finding out how long you have to let the recursion last. Here are some possibilities: Plot the m parallel runs in a joint plot against the iteration number. Compute means and variances for the m simulations at iteration t. You may plot against t. You may also compare against the known mean and variance of the standard normal.

(f). Discuss strategies for starting the iteration. Possibilities are start at a fixed point and random start.

(g). Run your program. Start at some fixed points . Experiment with the choice of a. Find out when you can stop the iteration. Is there an optimal region for a?

(h). Repeat (g) for various values of the starting point. Find out whether good choices for a changes with the starting point. (The experiment is highly relevant. In complicated situations it may be unknown where to start.).

(i). It there scope for changing a during the iteration? Does that invalidate the algorithm?

Exercise 37 (Sampling random pairs by Metropolis)
The purpose of this exercise is to extend the study in Exercise 36 to cover random pairs. Suppose (X, Y) has joint density $f(x, y) = ch(x, y)$, where h is a computable expression.

(a). Construct a Metropolis algorithm by direct extension of the one in Exercise 36. The recursion is to change (X, Y) simultaneously.

Often in more complicated problems Metropolis is designed in a one-at-a-time fashion. This means that candidates are drawn for X keeping Y fixed, and for Y keeping X fixed. For both variables use a uniform distribution over $(-a, a)$ to sample the candidates, as described in Exercise 36.

(b). Design a Metropolis algorithm that deals with each variable separately in this way. Write it out carefully as an algorithm. Identify the conditional densities q.

Consider the sampling of random Gaussian pairs with mean zero, variances one and correlation ρ. You will then need the conditional densities of X given $Y = y$ and Y given $X = x$. The former is Gaussian with mean ρy and variance $1 - \rho^2$. The latter is also Gaussian. The mean is ρx and the variance the same as before.

(c). Specialize the algorithm in (b).

(d). Write a program that samples such a Gaussian pair by Metropolis iterations.

(e). Run the program with $\rho = 0.0$, $\rho = 0.5$ and $\rho = 0.99$. Make the numerical investigations described in Exercise 36. In particular, study how long the iteration must run for different values of a and for different starting points. In particular: Investigate the importance of ρ.

You may use the statistical measures described in Exercise 36. For example, compute mean and variance for m replications of the simulations at each point during the iteration. It is in this situation natural to consider the correlation between the sampled random variables as an additional measure. At each point during the Metropolis iteration compute the correlation coefficient between the sampled X- and Y-values and plot it against the iteration number.

Exercise 38 (The Gibbs sampler in practice)
Assume again we are interested in simulating from the bivariate Gaussian distribution $N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where

$$\boldsymbol{\mu} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix},$$

that is the same setting as Exercise 35.

(a). Find the conditional distributions for x_1 given x_2 and x_2 given x_1.

(b). Implement the Gibbs sampler based on the condional distributions.

(c). Run the algorithm 1000 iterations for $a = 0$. Use simulations from the standard Gaussian distribution as starting points.

Estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from your simulations. Use traceplots to see how many of the first samples you should discard.

Make a plot of your simulations in the two-dimensional space, drawing a line between each iteration.

Comment on the results.

(d). Now repeat the previous point with $a = 0.99$.

Make similar estimates and plots and comment on the results.

(e). Compare the rate of convergence of the Gibbs sampler to that of the Metropolis algorithm. Such a comparison has to be properly formulated. Try to think of a way.

Exercise 39 (The Gibbs sampler)
The Gibbs sampler applies to vectors of random variables. We shall in this exercise consider random pairs (X, Y). The pedagogical idea is the same as in Exercises 36 and 37. By trying out the algorithm on a simple example where it is not needed, insight is gained into how the scheme operates and into the factors influencing its efficiency. The algorithm is so simple that it is self-explanatory.

```
Algorithm
  Select $X$        (initialization)
  Repeat
      Sample $Y$ from its conditional distribution given $X$.
      Sample $X$ from its conditional distribution given $Y$.
```

It can under general conditions be proved that a simulation of (X, Y) appears in the limit as the loop is continued on and on. We shall below actually prove this result in the simple example considered.

Let (X, Y) be bivariate normal, with means $E(X) = E(Y) = 0$, variances $\text{var}(X) = \text{var}(Y) = 1$ and correlation $\text{corr}(X, Y) = \rho$. The conditional distribution of Y given $X = x$ is then normal with mean ρx and variance $1 - \rho^2$, and the conditional distribution of X given $Y = y$ is defined by symmetry. Let $\{Z_n\}$ and $\{V_n\}$ be sequences of independent normal variables $(0, 1)$. Also assume independence between sequences.

(a). Show that the Gibbs sampler sets up the double recursion

$$Y_n = \rho X_n + \sqrt{1-\rho^2} Z_n, \qquad X_n = \rho Y_{n-1} + \sqrt{1-\rho^2} V_n.$$

It will be proved that as $n \to \infty$, (X_n, Y_n) converges to a sample of (X, Y) for any starting point $X_0 = \mu_0$. We shall also study the rate of convergence. Consider $\{X_n\}$ first.

(b). Show that $X_n = \rho^2 X_{n-1} + \varepsilon_n$, where $\varepsilon_n = \sqrt{1-\rho^2}(\rho Z_{n-1} + V_n)$.

Note that ε_n is normal with mean 0 and variance $\sigma_\varepsilon^2 = 1 - \rho^4$. Stochastic processes of the form $X_n = a X_{n-1} + \varepsilon_n$ is known as autoregressive of order one (or AR(1) for short). They are known to converge in distribution to a limit if $|a| < 1$. Take this result for granted. Explain why it applies here.

(c). Why is $E(X_n) = \rho^2 E(X_{n-1})$? Use this to establish that $E(X_n) = \rho^{2n}\mu_0$.

(d). Show that $\text{var}(X_n) = \rho^4 \text{var}(X_{n-1}) + \sigma_\varepsilon^2$. Since $\text{var} X_0 = 0$, this yields

$$\text{var}(X_n) = \frac{\sigma_\varepsilon^2}{1-\rho^4}(1 - \rho^{4n}).$$

Prove it.

(e). What is the limit for $E(X_n)$ and $\text{var}(X_n)$ when $n \to \infty$? Insert for σ_ε^2.

(f). The analysis for $\{Y_n\}$ is similar. Carry it out, i.e. repeat b)-e).

(g). Show that $E(X_n Y_n) = \rho E(X_n^2)$ and use this to calculate a formula for this expection.

(h). What happens to $E(X_n Y_n)$ when $n \to \infty$?

(i). Summarize your findings. What is the limit distribution of (X_n, Y_n)? Discuss the convergence speed. What is its dependence on ρ?

Exercise 40 (The Gibbs sampler in general)
Let $(X_1, ..., X_n)$ be a vector of random variables. Propose possible extensions of the algorithm in Exercise 39.

Exercise 41 (Combinations of Markov chains)
Assume you have two Markov chains described by the transition densities $P_1(y|x)$ and $P_2(y|x)$, both having a target distribution $\pi(x)$ as stationary distribution, that is

$$\pi(y) = \int_x \pi(x) P_j(y|x) dx . j = 1, 2.$$

Define now a new Markov chain by the transition density

$$P(y|x) = \alpha P_1(y|x) + (1 - \alpha) P_2(y|x)$$

were $\alpha \in [0, 1]$.

(a). Show that this new transition density also have $\pi(y)$ as stationary distribution

(b). Discuss the implication of this result with respect to constructing Markov chain Monte Carlo metohds.

Exercise 42 (Transformations)
Consider the sampling from the same bivariate normal distribution as in the exercises 40 and 38. We then know that (X, Y) can be represented as

$$X = Z \qquad Y = \rho Z + \sqrt{1 - \rho^2} V,$$

where Z and V are independent random variables, both normal $(0, 1)$. Explain why this is so.

(a). What is the convergence rate if you apply the Gibbs sampler to (Z, V) and obtain simulations of (X, Y) by using the transformation after the Gibbs sampler has converged?

(b). Discuss possible general lessons for designing simulation algorithms based on the Gibbs sampler or the Metropolis scheme.

Exercise 43 (Hamiltonian Monte Carlo)
Consider the logistic model

$$\begin{aligned} y_i \sim & \text{Binom}(1, p_i) \\ p_i = & \frac{\exp(\beta_1 + \beta_2 x_i)}{1 + \exp(\beta_1 + \beta_2 x_i)} \end{aligned}$$

in addition to prior assumptions $p(\boldsymbol{\beta}) = p(\beta_1)p(\beta_2)$ and $p(\beta_j) = N(0, \sigma_\beta^2)$ for $j = 1, 2$. Our aim will be to simulate from $p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{x})$. We will do so by the Hamiltonian Monte Carlo method.

(*a*). Show that in this case

$$H(\boldsymbol{\beta}, \boldsymbol{p}) = \tfrac{1}{2\sigma_\beta^2} \sum_{j=1}^{2} \beta_j^2 - \sum_{i=1}^{n} y_i(\beta_1 + \beta_2 x_i) + \sum_{i=1}^{n} \log(1 + \exp(\beta_1 + \beta_2 x_i)) + \tfrac{1}{2} \sum_{j=1}^{2} p_j^2$$

For a given ε, derive the updating equations for $\boldsymbol{\beta}$ and $\boldsymbol{p}$ using the Leapfrog method.

At the course web-page there is a script `HMC_logist.R` where both a Hamiltonian Monte Carlo and a Metropolis-Hastings algorithm is implemented (with multiple chains for each). Both algorithms have one tunring parameter (ε for HMC assuming L is fixed and σ_{prop}^2 for MH).

(*b*). Modify the tuning parameters such that you obtain reasable acceptance probabilities.

(*c*). Now change the number of iterations to 100 and burnin to 1 and run the script again (now with your updated tuning parameters).

Look at the Gelman-Rubin measures that also are provided. Based on this which algorithm would you prefer?

Hint: You might here think about the extra computational effort needed for the HMC algorithm. You may also repeat the simulations a few time to see if there is much variability in the results.

References

G. H. Givens and J. A. Hoeting. Computational statistics, volume 710. John Wiley & Sons, 2012.

K. Lange. Numerical analysis for statisticians. Springer Science & Business Media, 2010.