Exercises: Computer Lab 1

I highly recommend that you use Rstudio as your R GUI. I also recommend that you learn to use R Markdown as a way to document your code and to write your coursework solutions. You can get started with R Markdown by clicking File → New File → R Markdown and following the instructions.

1. Suppose $X_1, \dots, X_{10}$ are iid $U[0,1]$ random variables. Find the distribution of

   $$R(X) = \max_i\{X_i\} - \min_i\{X_i\}.$$

   What is $\mathbb{P}(R(X) > 0.99)$?

   If you are struggling, note that the code from the problems discussed in lectures is available on the module webpage.

2. Estimate the integral

   $$I = \int_0^{10} \frac{1}{(1+x^2)} dx$$

   using at least two different choices for the proposal density $g(\cdot)$.

   Find 95% confidence intervals for your estimates using the central limit theorem.

3. Let

   $$I_1 = \int_0^1 e^{-x^2} \mathrm{d}x \quad \text{and} \quad I_2 = \int_0^1 (\cos(50x) + \sin(20x))^2 \mathrm{d}x$$

   Estimate both of these integrals using Monte Carlo (using an estimator of your choice). Show that the root mean square error of your estimator scales as $O(n^{-1/2})$. To do this, you will need to repeat the analysis multiple times for a range of values of n (i.e., for $n = 10, 50, 100, 500, 1000, 5000, 10000$ estimate the integral 100 times and calculate the standard error).

   Calculate these integrals using the mid-ordinate rule. Show that the error now scales as $O(n^{-2})$. Note that because the mid-ordinate rule is very accurate for 1d intervals, you should only consider values of n between (2 and 100 say).

Computer class 2 exercises

Richard Wilkinson

Question 1

Suppose $X_1, \dots, X_{10} \sim N(\mu, \sigma^2)$, and that we observe data

$$\{2.561, -0.328, 2.607, 3.466, 2.012, 1.293, -2.301, 1.914, 5.779, 1.369\}$$

- Assume that $\sigma = 2$. Use a Monte Carlo test to test the null hypothesis

  $$H_0 : \mu = 1$$

  against the alternative

  $$H_1 : \mu \neq 1.$$

- Now assume that $\mu = 1$. Use a Monte Carlo test to test the null hypothesis

  $$H_0 : \sigma^2 = 1$$

  against the alternative

  $$H_1 : \sigma^2 > 1.$$

Question 2

We will now consider the example from the lecture notes in which we test for randomness in spatial patterns.

Download the data from MOLE, and load the data using the command

```r
load(Class1data.Rdata)
```

Locations of 50 points are stored in the vector `spatial`. Check that you can plot the data with the command

```r
plot(spatial)
```

Use a Monte Carlo test to test the null hypothesis that the distribution of points is uniform over the unit square.

Hint: The hardest part to this problem is computing the nearest neighbour distances for any particular pattern of points. You should think about how to do this, but in case you are stuck, one approach is given below.

```r
nnsum<-function(x){
    n<-length(x[,1])
    mx1<-matrix(x[,1],n,n,byrow=F)
    mx2<-matrix(x[,2],n,n,byrow=F)
    distances<-((mx1-t(mx1))^2+(mx2-t(mx2))^2)^0.5
    distances<-distances+diag(100,n,n)
    # so that we don't pick the zeros on the diagonal
    return(mean(apply(distances,2,min)))
}
```

Question 3

The MASS package contains information on the birth weight of 189 babies. Let's look at the difference between babies born to mothers who smoke and those whose mothers are non-smokers.

```
library(MASS)
attach(birthwt)
bwt.smoke <- bwt[smoke==1]
bwt.nonsmoke <- bwt[smoke==0]
```

Part i)

We'll look below at whether the two groups have the same mean (which is the usual question of interest), but to begin with consider testing the null hypothesis

$$H_0 : \sigma^2_{smoke} = \sigma^2_{nonsmoke}$$

against the alternative that

$$H_1 : \sigma^2_{smoke} < \sigma^2_{nonsmoke}.$$

Note that a useful test statistic to use is

$$T = \frac{s_{smoke}}{s_{nonsmoke}}$$

where s^2 is the usual estimator of variance.

- Use a randomization test to test H_0 vs H_1. What do you conclude?
- What parametric test would you use to test these hypotheses?

Part ii)

Now assume that both distributions are Gaussian.

- Repeat the hypothesis test using a Monte Carlo test.
- Check the Gaussian assumption. Which of these three approaches (Monte Carlo, randomization, or classical tests) do you prefer? Which do you think is most trustworthy?

Part iii)

Use a randomization test to test

$$H_0 : \mu_{bwt.smoke} = 2600$$

against the alternative

$$H_1 : \mu_{bwt.smoke} \neq 2600$$

Computer class 3 exercises

Richard Wilkinson

Question 1

- Generate 100 $N(0,1)$ random variables. Plot the CDF and overlay the true $N(0,1)$ CDF on top.
- Estimate the probability that $X < 0.5$ using the ECDF and calculate the true value. Note that in R, `ecdf(X)` returns the ECDF of a random sample X as a function. This can then be evaluated, for example, `ecdf(X)(0.5)` etc
- Repeat this process a large number of times to convince yourself that your estimator is unbiased.
- [*Optional*] The Dvoretzky-Kiefer-Wolfowith (DKW) inequality says that for any $\epsilon > 0$, and any $n > 0$, then for all $x \in \mathbb{R}$

  $$P(\sup_x |F_n(x) - F(x)| \geq \epsilon) \leq 2e^{-2n\epsilon^2}$$

  where F_n is the ECDF based on a sample of size n, and F is the true CDF. Convince yourself empirically that this result is true.

Question 2

Consider the `hills` dataset in the MASS package in R, which contains data on the record time for each of 35 Scottish hill races. We want to build a model to predict the record time on the basis of the race distance and the total amount of height gained during the route. Because we are worried about outliers in the data, we will use a robust regression approach using M estimators.

We can fit a robust linear model to this dataset using the command

```
library(MASS)
fit <- rlm(time~dist+climb, hills, maxit=100)
summary(fit)
```

```
##
## Call: rlm(formula = time ~ dist + climb, data = hills, maxit = 100)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -10.75039  -3.28395  -0.03358   3.53791  65.70100
##
## Coefficients:
##             Value   Std. Error t value
## (Intercept) -9.6067  1.7545    -5.4754
## dist         6.5507  0.2451    26.7237
## climb        0.0083  0.0008     9.9199
##
## Residual standard error: 5.209 on 32 degrees of freedom
```

The coefficient standard errors reported by `rlm` rely on asymptotic approximations, and may not be trustworthy in a sample of size 35. Thus, we will use bootstrapping to estimate confidence intervals.

1. Calculate a bootstrap 95% confidence interval for the coefficient of `dist` using model-based resampling.
2. Recalculate this confidence interval by now using case resampling - i.e. by bootstrap resampling (x_i, y_i), re-fitting the model to each bootstrap sample, and forming the bootstrap distribution of β. Contrast

this with your answer from the previous part, and with a 95% confidence interval obtained from the asymptotic standard error estimates reported by the `summary(fit)` command.

3. An alternative model is proposed, which includes an interaction term between `dist` and `climb` and a quadratic `climb` term. Calculate the mean-square prediction error for both models using leave-one-out cross validation. Which model do you prefer?

```
fit2 <- rlm(time~dist*climb+I(climb^2), hills, maxit=100)
```

Note that there are built-in commands in R for doing bootstrapping and cross-validation. You should *not* use these (except to check your answer), but instead write your own R commands.

Question 3

Failure of the bootstrap: Suppose $X_1, \dots, X_n \sim U[0, \theta]$.

1. Show that the maximum likelihood estimator of θ is

$$\hat{\theta} = \max_{i=1,\dots,n} X_i$$

2. Calculate the CDF of $\hat{\theta}$

3. Generate a toy dataset of size $n = 100$ (assuming that $\theta = 1$) using the code

```
set.seed(1)
n=100
X = runif(n,min=0, max=1)
```

4. The parametric bootstrap works by generating a bootstrap sample from the fitted parametric model, i.e., simulating

$$X_1^*, \dots, X_n^* \sim U[0, \hat{\theta}],$$

and then fitting the parametric model, i.e., setting

$$\hat{\theta}^* = \max_{i=1,\dots,n} X_i^*$$

Explain why for the parametric bootstrap

$$\mathbb{P}(\hat{\theta}^* = \hat{\theta}) = 0.$$

Simulate 10^5 (parametric) bootstrap replicates and compare the distribution of these with the non-parametric bootstrap estimate.

5. The non-parametric bootstrap creates bootstrap samples by sampling from the empirical CDF. Simulate 10^5 non-parametric bootstrap samples and compare the true distribution of $\hat{\theta}$ to the histogram from the non-parametric bootstrap and with the parametric bootstrap.

6. If $\hat{\theta}^*$ is a bootstrap estimate of $\hat{\theta}$, show that

$$\mathbb{P}(\hat{\theta}^* = \hat{\theta}) = 1 - (1 - (1/n))^n$$

and hence that

$$\mathbb{P}(\hat{\theta}^* = \hat{\theta}) \to 1 - e^{-1} \approx 0.632$$

as $n \to \infty$.

Computer class 4 exercises

Richard Wilkinson

Question 1

In this question we will compare the performance of 3 models on the hills dataset.

```
library(MASS)
M1 <- lm(time ~ dist + climb, data=hills)
M2 <- lm(time ~ dist+climb + I(climb^2), data=hills)
M3 <- lm(time ~ dist*climb+I(dist^2)+I(climb^2), data=hills)
```

Note that model 3 fits the training data better than model 2 which is better than model 1 (as they must be as they are nested models).

```
print(deviance(M1))
```

```
## [1] 6891.867
```

```
print(deviance(M2))
```

```
## [1] 4514.554
```

```
print(deviance(M3))
```

```
## [1] 4298.945
```

Although model 3 achieves the best fit to the data, we may be concerned that it is over-fitting. To test this, we will use cross-validation to assess the predictive skill of the three models. We will use the cvTools package in R to do cross-validation, but it is easy to write your own code to do this if you wish.

```
install.packages('cvTools')
```

First we need to create the folds. Choose K=5 to begin with.

```
library(cvTools)
```

```
## Loading required package: lattice
```

```
## Loading required package: robustbase
```

```
folds <- cvFolds(n=dim(hills)[1], K = 5, R = 1)
```

Examine the folds object you have just created. Note how every observation has been randomly assigned to one of 5 folds. By changing the value of R we can create several different random assignments of the data into folds.

To fit the model using cvTools we need to create a call function which runs the command we wish to repeat. To fit model 1 we could do

```
call_M1 <- call <- call('lm', formula=time ~ dist + climb)
```

We then use the cvTool command to fit the model on the data 5 times, leaving out one fold each time.

```
(CV5fold_M1 <- cvTool(call_M1, data=hills,y=hills$time, folds=folds))
```

```
##            CV
## [1,] 16.82385
```

This reports the root mean square predictive error (rmspe) for the model. Read the help pages for mode detail.

Change the value of R in the cvFolds command and repeat the analysis to see how variable your answer is.

Repeat the analysis for M2 and M3. Which model would you use for prediction?

Try 10-fold and leave-one-out cross validation. Does your conclusion about the predictive skill of the models change?

Question 2

The density of the standard Cauchy distribution is

$$f(x) = \frac{1}{\pi(1+x^2)}.$$

By using the substitution $x = \tan(u)$ or otherwise show that

$$F(x) = \frac{1}{\pi}\tan^{-1}(x) + \frac{1}{2}.$$

Use the inversion method to derive an algorithm for generating a Cauchy random variable.

Implement this in R and use it to generate 10^6 random variables. Check your estimated values of

$$\mathbb{P}(X \leq -10), \mathbb{P}(X \leq -5), \mathbb{P}(X \leq 0), \mathbb{P}(X \leq 5), \text{ and } \mathbb{P}(X \leq 10)$$

against the true values using the built in CDF in R (`pcauchy`).

Question 3

Consider the density function

$$g(x) = \tfrac{1}{2}\mathrm{e}^{-|x|}$$

for $-\infty < x < \infty$. Show how $g(x)$ may be sampled from by considering it to be the mixture of two exponential distributions (hint: you may find point (d) on slide 17 useful).

(a) Derive a rejection sampling algorithm for sampling a standard normal random variable using g as the proposal distribution. Implement your method, and simulate 10^5 N(0,1) rvs. Check your answer.

(b) What is the acceptance probability of a single random draw from $g(x)$ for your algorithm? Check this numerically using your code.

Computer class 5 exercises

Richard Wilkinson

Question 1

In this question we will use rejection sampling to solve a Bayesian inference problem. To begin with, we will consider a problem for which a conjugate analysis exists.

Suppose that we want to learn about the unknown parameter p, to which we assign a U[0,1] distribution. We collect data $x_1, \ldots, x_{10}$ which are independent $Bin(20, p)$ random variables. Show that the likelihood times the prior for this problem is proportional to

$$f_1(p) = p^{\sum x_i}(1-p)^{200-\sum_i x_i}$$

We are told that $\sum_{i=1}^{10} x_i = 50$.

- Describe a rejection sampling algorithm for sampling from the posterior distribution using a $U[0,1]$ distribution as the proposal density g and use it to draw a histogram of the posterior distribution.
- In this case, we can calculate the posterior distribution analytically using a conjugate analysis. Show that the posterior distribution is

  $$\pi(p|x_1, \ldots, x_{10}) = \text{Beta}(51, 151).$$

- Check your code by plotting the pdf of this distribution on top of a histogram of samples you generated using the rejection algorithm.

Question 2

We will now tackle the same problem as in question 1 but using importance sampling instead.

- Using a $U[0,1]$ distribution as the importance distribution, use importance sampling to generate a weighted sample

  $$\{p_i, w_i\}_{i=1}^N$$

  of particles and weights that approximates the posterior distribution.
- Calculate the posterior mean of p. Note that we can approximate any integral by a weighted sum. So for example,

  $$E(p|x) = \int p\pi(p|x)dp \approx \frac{\sum w_i p_i}{\sum w_i}.$$

  Alternatively, we can use the weighted version of statistical estimators in the Hmisc library, for example, `wtd.mean`. You may need to install Hmisc the first time you use it (`install.packages('Hmisc')`)
- We can resample the particles to get an unweighted sample of particles. To do this, first convert the weights into probabilities,

  $$W_i = \frac{w_i}{\sum w_i}$$

  and then sample from $\{p_i\}_{i=1}^N$ with replacement, picking particle i with probability W_i. Calculate the number of unique particles in your unweighted sample.
- Use the resampled particles to plot a histogram of the posterior distribution.
- Repeat the steps above using a $Beta(10, 30)$ distribution as the importance distribution.

- The variance of the importance weights is a useful measure of how successful a given importance distribution will be - we want the variance to be as small as possible. A related quantity that is often used is the effective sample size (ESS)
$$ESS = \frac{1}{\sum W_i^2}$$
where the W_i are the normalised weights ($\sum W_i = 1$). If all the weights are the same (i.e. they have zero variance), then the ESS = N, i.e. the sample is as effective as a sample of N unweighted particles. Whereas in the worst case where all the weights are 0 except for one which has $W = 1$, then the ESS=1, i.e., the sample is equivalent to a single unweighted sample. Calculate the ESS for your two importance distributions to see which gives a better sample.
- What choice of importance distribution would give the best possible ESS?

Question 3

This problem is described in the notes. Here we will work through the details.

Patients suffering from leukaemia are given a drug, 6-mercaptopurine (6-MP), and the number of days x_i until freedom from symptoms is recorded for patient i:

$$6^*, 6, 6, 6, 7, 9^*, 10^*, 10, 11^*, 13, 16, 17^*, 19^*, 20^*, 22, 23, 25^*, 32^*, 32^*, 34^*, 35^*,$$

where a * denotes censored observation. The time x to the event of interest follows a *Weibull* distribution:

$$f(x|\alpha, \beta) = \alpha\beta(\beta x)^{\alpha-1} \exp\{-(\beta x)^\alpha\}$$

for $x > 0$. For censored observations, we can show that

$$P(x > t|\alpha, \beta) = \exp\{-(\beta t)^\alpha\}.$$

We want to estimate the posterior mean of θ, and the posterior 5th and 95th percentiles.

Define d to be the number of uncensored observations and $\sum_u \log x_i$ to be the sum of logs of all uncensored observations. If we use the following prior distributions for α and β

$$f(\alpha) = 0.001 \exp(-0.001\alpha), \qquad f(\beta) = 0.001 \exp(-0.001\beta).$$

then we can show that the log of the posterior distribution is proportional to

$$\log f(\theta|x) \propto h(\theta) := d \log \alpha + \alpha d \log \beta + (\alpha - 1) \sum_u \log x_i - \beta^\alpha \sum_{i=1}^n x_i^\alpha - 0.001\alpha - 0.001\beta + K,$$

where $\theta = (\alpha, \beta)^T$.

- Use importance sampling to estimate the posterior mean of α and β, using an $Exp(1)$ distribution for both parameters. Does this work well?

We will now use the Laplace approximation to design a better choice of the proposal density g.

- Obtain the posterior mode of θ, i.e., maximise $h(\theta)$ defined above. You can do this in R by writing a function to evaluate h and then using the `optim` command. Note that optim does minimization by default.

- Find the Hessian (matrix of second derivatives) of $h(\theta)$ at $\theta = m$, either by deriving it analytically, or estimating it using numerical differentiation (the `hessian` command in the numDeriv package works well),

$$M = \left(\begin{array}{cc} \frac{\partial^2}{\partial\alpha^2} h(\theta) & \frac{\partial^2}{\partial\alpha\partial\beta} h(\theta) \\ \frac{\partial^2}{\partial\alpha\partial\beta} h(\theta) & \frac{\partial^2}{\partial\beta^2} h(\theta) \end{array} \right).$$

- Use an importance sampling algorithm to estimate the posterior mean and 5th and 95th percentiles of this distribution. Use a multivariate Gaussian distribution as your proposal, with mean m and covariance matrix $V = -M$. To simulate from a multivariate normal, you can either use the Cholesky decomposition of V, or use the mvtnorm package in R (you may need to install it using `install.packages('mvtnorm')` the first time you use this). Note that you will need to use `wtd.quantile` or resample the particles and use `quantile` to get the quantiles.
- Resample the particles to get an unweighted sample, and plot the posterior distribution of α and β.