MSc Statistical Programming 2023 Assessed Practical Assignment

Preamble

- **Assignment due date:** The report is due Wednesday noon Hilary Term week 0 2024.
- **Submission instructions:** Submission is through the Inspera system. Please make sure your practical number (*e.g.* P123) is on the first page of your submission.
- **Submitted assignment format:** You are required to submit both a knitr code `.rnw` file, as well as a single PDF, which you get from compiling your knitr `.rnw` file. Your `.rnw` file should be self-contained, and should compile when run in this way from an empty R session (for example file `"P123.rnw"`)

```
knitr::knit("P123.rnw"); tinytex::latexmk("P123.tex")
```

  To ensure an empty R session if you're using RStudio, try restarting Rstudio entirely, or clearing the workspace and restarting R. For compilation, you may assume all files mentioned in this assessment are available to you in the same directory as your `.rnw` file, when it is being compiled. Your PDF should contain the answers to the questions below, in the order the questions are set. For each part of the question, you should submit a combination of R code, the output of the code, plots or tables, and any further text elaborating on the answer, as required. Please show code in the output file when directed to do so. All code used must be included in the `.rnw` file you submit. There is no word or page limit on the report length.

- **Example:** An example question PDF, along with a solution `.rnw` file and solution PDF, are included online. You are welcome to start from the example solution `.rnw` file as a template, and modify it to include your answers to the assignment PDF. The example solution PDF was compiled using code as above.
- **Assessment criteria:** You will primarily be assessed around how well you answer each question, and whether the `.rnw` file you submit compiles on a new system. You will secondarily be assessed for the following: the quality of your report including figures and tables; the quality of your R code including the appropriate use of functions as well as structure, variable and function naming; and the clarity of your written answers. If your submitted `.rnw` file does not compile due to missing CRAN R libraries, these will be installed and the code re-run.
- **Code requirements:** Provided code must be written solely in R. You can use any R package used in the course lectures or practicals. You can use R packages not covered in this course, provided they are reasonably related to material covered in the course - for example, additional `tidyverse` or `ggplot2` related libraries, or things related to `knitr` like `kableExtra`. Note that you might find it helpful to wrap library calls like `suppressPackageStartupMessages(library(testthat))` to avoid printing warning messages to the output file.

British house prices

*Please note: throughout this question, in your answers, unless otherwise stated, please **DO NOT** include R code in your output report*

In this question, we are going to use `Average-prices-2023-07.csv`, which contains average house prices for houses in the UK, available from [1]. These data are available and republished under the following attribution statement

> Contains HM Land Registry data, Crown copyright and database right 2020. This data is licensed under the Open Government Licence v3.0.

We are further going to use `series-181023_cumulative.csv`, which contains information about inflation in the UK, available from [2]. This data is also available under an Open Government Licence v3.0.

1. Read in the house prices data, and make a plot of the average house price for the `England` region, as a function of time.
2. Make a new plot that augments your previous plot by also plotting the average price of houses in the 4 Oxford regions, *i.e.* those that have `Oxford` in the region name. Restrict the date range to only dates where both England and Oxford level regional information are available.
3. The prices of houses in Oxford are elevated, but are they the most elevated regions? For each region in England, as defined by regions where the first character of the `Area_code` variable is `E`, compare the per-month average price for that region with the England region, and then take a median of the ratio (*i.e.* if `x` and `y` are vectors representing the prices for that region and England respectively, matched by date, then you could take `median(x / y)`). From this, find the 10 regions with the highest values of the median ratio that you just calculated. Make a table for those 10 regions that includes the name of the region, the median ratio just calculated, the initial and final house prices for the dates examined, and the percentage increase between the initial and final prices, for houses in that region for the studied time period. Are any of the Oxford regions included?
4. Read in the inflation data, and by restricting it to the per-month entries, merge it into the housing data. Over the range of dates for which both inflation and the `England` region house prices are defined, make a single plot that shows both the increase in English house prices, as well as inflation, over the studied period. Which has risen faster over the studied period, house prices or inflation? *You may find it useful to use multiple y axes*

Chromosome painting

*Please note: throughout this question, in your answers, unless otherwise stated, please **DO** include R code in your output report. Further, you don't need to validate the input arguments of any function you write*

Humans, like many species, are diploid, meaning we contain two copies of each chromosome, one inherited from each parent. These inherited chromosomes are not direct copies of the parental chromosomes, but are themselves formed by a process called recombination. In recombination, new molecules of DNA are formed by taking the two copies of DNA present in the cell, lining them up, forming breaks, and recombining the segments of DNA from the two donor molecules, to produce new molecules that are mosaics of the original ones. As a consequence, genetic material on chromosomes is shuffled up during each generation. As a further consequence, when populations of different ancestries combine to form a new population, it is possible to try and determine the ancestral background along the chromosome of offspring in this new population, by comparing their DNA to that from the original source populations.

In statistical genetics, chromosome painting is a technique that is often used to determine the ancestry of the chromosomes that people inherit. Here we'll focus on looking at variation at specific genetic locations (called variant sites, or just variants), specifically where there are two common DNA variants present (called alleles). We'll call a haplotype the collection of alleles carried by a particular chromosome of DNA at those variant sites. With chromosome painting, we'll make use of a haplotype reference panel, which is a collection of haplotypes, and we'll seek to model a given target haplotype as a mosaic of the haplotype reference panel. When performing chromosome painting for ancestry inference, we use a haplotype reference panel where the haplotypes come from known populations, and use this to determine the ancestry of target haplotypes. For example, commercially, 23andMe offer this as part of their standard genotyping service (try searching for "23andMe chromosome painting" on the internet).

In this practical question, you will implement a simplified version of a standard algorithm for chromosome painting, called the Li and Stephens algorithm [3]. The Li and Stephens algorithm uses a hidden Markov model (HMM), and we will use known features of HMMs to help us test that the algorithm has been implemented correctly. In this question you will also see how you can decrease the computational burden of the algorithm, and increase its speed, by careful consideration of how probabilities are calculated. Finally you will use your implementation of the Li and Stephens painting algorithm on real haplotypes from the 1000 Genomes Project [4, 5], to paint African American sample haplotypes from the Southwest USA using labelled haplotypes from known European (Northern and Western European ancestry) and African (Yoruban) haplotypes.

With our HMM, we'll consider a fixed haplotype reference panel H with K rows (number of reference haplotypes) and T columns (number of genetic variants), with $H_{k,t} \in \{0, 1\}$ to indicate the presence of the reference 0 and alternate 1 allele of the k^{th} haplotype and the t^{th} genetic variant. We'll consider a target haplotype $O = o$ that we want to paint as a vector of length T with entry $o_t \in \{0, 1\}$ being the allele carried at the t^{th} variant.

With HMMs, there are different things we may seek to infer. Here, we'll infer probabilities associated with a random variable Q of length T which represents which reference haplotype is being copied, where $P(Q_t = k|O = o)$ will be our target measure, and which measures the probability that at variant t, the target haplotype O (random variable O, observation o) is copying from reference haplotype k. By summing this across labelled reference haplotypes from

different populations, we can infer how the genetic ancestry varies along a chromosome.

To formulate an HMM, we need certain probabilities. Here we are going to use the following

1. Initial probabilities
$$\pi_k = P(Q_1 = k) = \frac{1}{K} \tag{1}$$
2. Transition probabilities
$$A_{j,k} = P(Q_{t+1} = k | Q_t = j) = \begin{cases} \frac{1-0.999}{K} + 0.999, & j = k \\ \frac{1-0.999}{K}, & j \neq k \end{cases} \tag{2}$$
3. Emission probabilities (given e an error rate, default 0.1)
$$b_{k,t} = P(O_t = o_t | Q_t = k) = \begin{cases} (1-e), & o_t = H_{k,t} \\ e, & o_t \neq H_{k,t} \end{cases} \tag{3}$$

In an HMM two algorithms we might want to run are called the forward and the backward algorithm, to calculate matrices α and β, both with K rows and T columns. We can implement the forward algorithm, to calculate $\alpha_{k,t} = P(O_1 = o_1, ..., O_t = o_t, Q_t = k)$ as follows

1. Initialization
$$\alpha_{k,1} = \pi_k b_{k,1}, \qquad 1 \leq k \leq K \tag{4}$$
2. Induction (from $t = 2$ to $t = T$)
$$\alpha_{k,t} = \left(\sum_{i=1}^{K} \alpha_{i,t-1} A_{i,k} \right) b_{k,t}, \qquad 1 \leq k \leq K \tag{5}$$

Similarly, the backward algorithm calculates $\beta_{k,t} = P(O_{t+1} = o_{t+1} ..., O_T = o_T | Q_t = k)$, and is calculated as follows

1. Initialization
$$\beta_{k,T} = 1, \qquad 1 \leq k \leq K \tag{6}$$
2. Induction (from $t = T - 1$ to $t = 1$)
$$\beta_{k,t} = \sum_{i=1}^{K} A_{k,i} b_{i,t+1} \beta_{i,t+1}, \qquad 1 \leq k \leq K \tag{7}$$

After running both the forward and backward algorithms, it is possible to calculate γ, a matrix with K rows and T columns, with entries $\gamma_{k,t}$ given by

$$\gamma_{k,t} = P(Q_t = k | O = o) = \frac{\alpha_{k,t} \beta_{k,t}}{P(O = o)} \tag{8}$$

where $P(O = o) = \sum_{k=1}^{K} \alpha_{k,T}$. It is $\gamma_{k,t}$ that represents the probability that the target haplotype copies from the reference haplotype k at SNP t, and this is what we are particularly interested in, and what is ultimately used for ancestry inference using chromosome painting.

1. Implement the forward algorithm for the Li and Stephens algorithm described above as a function called `forward` with arguments `haps` (representing H the haplotype reference panel, a matrix with K rows and T columns), `hap` (representing o the observed target haplotype, a vector of length K) and `error` with default value `0.1` (representing e, a vector of length 1, as the error rate). `forward` should return the α matrix as defined above.

2. Now we can use known features of HMMs to check that we've implemented the forward algorithm correctly. If it is the case that the reference and target haplotypes always match, then the α matrix returned by the forward algorithm will have a specific form. Specifically, the first column will have entries all equal to $(1-e) \times \frac{1}{K}$, and then each successive column will be equal to the previous one multiplied by $(1-e)$. Write a unit test in R using the `testthat` framework that verifies this expected behaviour (for instance by setting `haps` and `hap` to be entirely 0). *Here and in other unit tests, choose small but distinct values for K and T, say between 5 and 20*

3. Implement the backward algorithm for the Li and Stephens algorithm described above in a function called `backward` with the same arguments as `forward` with the same meaning in the same order. `backward` should return the β matrix as defined above.

4. Write a function called `gamma` with the same arguments as `forward` with the same meaning in the same order, and use this to calculate the γ matrix as defined above, using the `forward` and `backward` functions you just wrote.

5. Again, using known features of HMMs, we can verify that the `gamma` function works as expected. Specifically, we expect that the column sums of the γ matrix should sum to 1. Write a unit test using the `testthat` framework where both `haps` and `hap` contain 0 or 1 entries at random, and verify that `gamma` produces this expected behaviour.

6. Explain what the computational complexity of the forward and backward functions are, as a function of K and T, in an informal manner, in no more than a few sentences each. Verify these results computationally. *Here and in further questions, do not show code used to verify computational complexity in the output, but do show result tables or plots*

7. Consider the induction step of the forward algorithm, specifically, when we perform this step

$$\alpha_{k,t} = \left(\sum_{i=1}^{K} \alpha_{i,t-1} A_{i,k} \right) b_{k,t} \tag{9}$$

   Now note that we can also do this in the induction step by first calculating the following which is constant for all k

$$\phi = \left(\frac{1 - 0.999}{K} \right) \sum_{i=1}^{K} \alpha_{i,t-1} \tag{10}$$

   and then updating α for each k using the following

$$\alpha_{k,t} = \left(\phi + 0.999 \alpha_{k,t-1} \right) b_{k,t} \tag{11}$$

   Implement this change in a new version of the forward algorithm called `forward2`. Write a unit test using the `testthat` framework that verifies that `forward` and `forward2` give the same output for the same input. Explain in a few sentences what you expect the computational complexity of `forward2` to be versus `forward`, and verify this computationally.

8. Similarly, consider the induction step of the backward algorithm, specifically, where we perform this step

$$\beta_{k,t} = \sum_{i=1}^{K} A_{k,i} b_{i,t+1} \beta_{i,t+1} \tag{12}$$

Now note that we can also do this in the induction step by first calculating

$$\phi = \left(\frac{1 - 0.999}{K} \right) \sum_{i=1}^{K} (\beta_{i,t+1} b_{i,t+1}) \tag{13}$$

and then updating β for each k using the following

$$\beta_{k,t} = \phi + 0.999 \beta_{k,t+1} b_{k,t+1} \tag{14}$$

Implement this change in a new version of the backward algorithm called `backward2`. Write a unit test using the `testthat` framework that verifies that `backward` and `backward2` give the same output for the same input. Explain in a few sentences what you expect the computational complexity of `backward2` to be versus `backward`, and verify this computationally.

9. Make a new version of the gamma function called `gamma2` that uses `forward2` and `backward2` to calculate the γ matrix. Write a unit test using the `testthat` framework to verify that `gamma` and `gamma2` give the same output for the same input.

10. Explain in a few sentences what you expect the computational complexity of both `gamma` and `gamma2` to be, and verify it computationally.

11. Download the reference haplotype file `refpanel.txt` and target haplotype file `samples.txt` from the website, which are based on real 1000 Genomes Project human data. Read these files into R, taking note that both files have both row and column names, representing the sample population and name (row name) and genetic variant ID (column name). Now, the reference haplotypes have labels which indicate which populations they are from, which are either CEU = Northern and Western European ancestry, or YRI = Yoruba in Ibadan, Nigeria. Similarly, the target haplotypes have labels ASW = African Ancestry in Southwest US. Paint the first 5 target haplotypes one a time using the reference panel using the `gamma2` function you wrote. From the painting results, sum the contribution at each variant from the African YRI haplotypes, as well as the European CEU haplotypes, for these 5 samples separately, as they vary across the investigated region (*i.e.* calculate $\sum_{k \in I_{YRI}} \gamma_{k,t}$ to be the African contribution at a variant t, where I_{YRI} is a set indicating which haplotypes are from the YRI population). For which of these 5 sample haplotypes does the painting suggest an entirely African genetic background over the investigated chromosome?

References

[1] UK House Price Index: data downloads July 2023;. Accessed: 2023-11-03. https://www.gov.uk/government/statistical-data-sets/uk-house-price-index-data-downloads-july-2023#download-the-data.

[2] Office of National Statistics *CPIH INDEX 00: ALL ITEMS 2015=100*;. Accessed: 2023-11-03. https://www.ons.gov.uk/economy/inflationandpriceindices/timeseries/l522/mm23/previous.

[3] Li N, Stephens M. Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. Genetics. 2003 Jan;165(4):2213-33. Available from: http://www.genetics.org/content/165/4/2213.

[4] "The 1000 Genomes Project Consortium". A global reference for human genetic variation. Nature. 2015 Oct;526(7571):68-74. Available from: http://www.nature.com/nature/journal/v526/n7571/full/nature15393.html.

[5] Byrska-Bishop M, Evani US, Zhao X, Basile AO, Abel HJ, Regier AA, et al. High-coverage whole-genome sequencing of the expanded 1000 Genomes Project cohort including 602 trios. Cell. 2022 Sep;185(18):3426-40.e19. Available from: https://www.sciencedirect.com/science/article/pii/S0092867422009916.