Example sheet 4

Markov chains

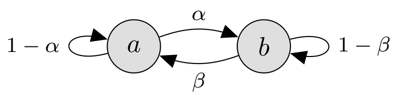
Data Science—DJW—2020/2021

Question 1. For the Cambridge weather simulator, section 10.2 of lecture notes, show that

$$\mathbb{P}(X_3 = r \mid X_0 = g) = \sum_{x_1,x_2} P_{gx_1} P_{x_1x_2} P_{x_2r}.$$

Explain your reasoning carefully.

Question 2. Here is the state space diagram for a Markov chain. Find the stationary distribution.



Question 3. For this Markov chain, draw the state space diagram, and give pseudocode to compute the stationary distribution.

```
def rw():
    MAX_STATE = 9
    x = 0
    while True:
        yield x
        d = numpy.random.choice([-1,0,1], p=[1/4,1/2,1/4])
        x = min(MAX_STATE, max(0, x + d))
```

Question 4. Let $X_n \in \mathbb{N}$ be the number of infectious people on day n of an epidemic, and consider the Markov chain model

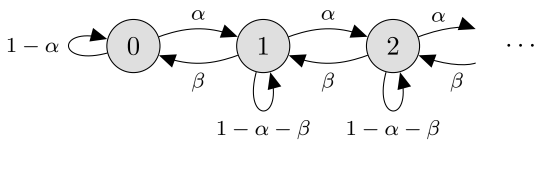
$$X_{n+1} = X_n + \text{Poisson}(rX_n/d) - \text{Bin}(X_n, 1/d).$$

We would like to compute the probability that, starting from state $X_0 = x$, the epidemic dies out i.e. hits state 0. In order to solve this by computer, we'll cut the state space down to $\{0, \dots, N\}$, for some sufficiently large N, by amalgamating all the states with $\geq N$ infected and letting the transition from state N back to itself have probability 1.

(a) Give pseudocode to compute the transition matrix. You should give your answer in terms of `binom.pmf` and `poisson.pmf`, the likelihood functions for the two distributions in question.

(b) Give pseudocode to compute the probability that the epidemic dies out, starting from any initial state $x \in \{0, \dots, N\}$. (For $r = 1.1$ and $d = 14$, for $X_0 = 50$, the probability is 0.7%.)

Question 5. Here is the state space diagram for a Markov chain, with state space $\{0, 1, 2, \dots\}$. It is parameterized by α and β, with $0 < \alpha < \beta$ and $\alpha + \beta < 1$. Let $\pi_n = (1 - \alpha/\beta)(\alpha/\beta)^n$, $n \geq 0$. Show that π is a stationary distribution.



Question 6. *[This question is about a handy trick called 'detailed balance' that can, if you're lucky, make it very easy to find the stationary distribution of a Markov chain. You'll need it in Part II Machine Learning and Bayesian Inference.]*

(a) Consider a Markov chain with transition matrix P, and suppose π is a distribution that satisfies

$$\pi_x P_{xy} = \pi_y P_{yx} \quad \text{for all states } x \text{ and } y.$$

Such a distribution is said to be *in detailed balance*. Prove that π is a stationary distribution.

(b) We're given a connected, undirected graph. Consider a random walk on the vertices of this graph, that at each timestep follows one of the edges chosen at random, each edge from its current vertex equally likely. Find the stationary distribution.

Question 7. Consider a moving object with noisy location readings. Let X_n be the location at timestep $n \geq 0$, and Y_n the reading. Here's the simulator.

```
def hmm():
    MAX_STATE = 9
    x = numpy.random.randint(low=0, high=MAX_STATE+1)  # initial location X₀
    while True:
        e = numpy.random.choice([−1,0,1])
        y = min(MAX_STATE, max(0, x + e))  # noisy reading of location
        yield y
        d = numpy.random.choice([−1,0,1], p=[1/4,1/2,1/4])
        x = min(MAX_STATE, max(0, x + d))  # new location at next timestep
```

We'd like to infer the location X_n, given readings $y_0, \ldots, y_n$.

(a) Draw the causal diagram.

(b) Give justifications for the following three equations, which give an inductive solution. First the base case,

$$\Pr(x_0 \mid y_0) = \text{const} \times \Pr(x_0) \Pr(y_0 \mid x_0),$$

and next two equations for the induction step,

$$\Pr(x_n \mid h) = \sum_{x_{n-1}} \Pr(x_{n-1} \mid h) \Pr(x_n \mid x_{n-1})$$

$$\Pr(x_n \mid h, y_n) = \text{const} \times \Pr(x_n \mid h) \Pr(y_n \mid x_n).$$

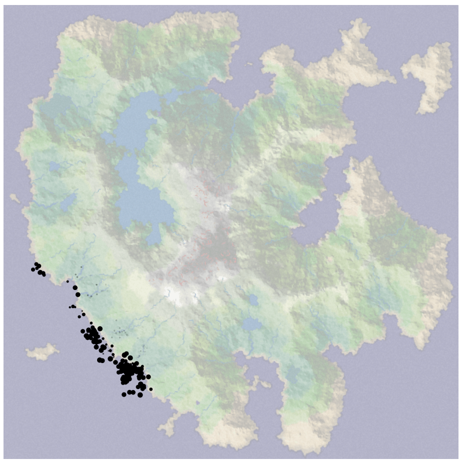
In these two equations, h stands for $(y_0, \ldots, y_{n-1})$, and we'll assume we've already found $\Pr(x_{n-1}|h)$.

(c) Give pseudocode for a function that takes as input a list of readings $(y_0, \ldots, y_n)$ and outputs the probability vector

$$[\pi_0, \ldots, \pi_{\texttt{MAX\_STATE}}], \qquad \pi_x = \mathbb{P}(X_n = x \mid y_0, \ldots, y_n).$$

(d) If your code is given the input $(3, 3, 4, 9)$, it should fail with a divide-by-zero error. Give an interpretation of this failure.

[Note. In this exercise, the state space is $\{0, 1, \ldots, \texttt{MAX\_STATE}\}$, which is very small, so it's practical to compute the distributions exactly. When the state space is large, we need computational approximations. You are encouraged to follow the notebook at `https://github.com/damonjw/datasci/blob/master/ex4.ipynb`, in which you will be asked to implement a "particle filter" to solve this problem on a large map.]



Hints and comments

Question 1. In section 10.3 we calculated $\mathbb{P}(X_2 = r \,|\, X_0 = g)$, using two tools: the memoryless property of Markov chains, and 'resetting the clock'. Use the same method here. Start by using the law of total probability, conditioning on X_1. Then condition on X_2. (Or you can do X_2 first, or you can do both together.)

Question 2. The equations to solve are

$$\pi_x = \sum_y \pi_y P_{yx} \text{ for all } x \qquad \text{and} \qquad \sum_x \pi_x = 1.$$

In section 11.1 we derived these, but for answering questions you should just remember them and apply them. Also, in your aswer, you should mention *irreducibility*, as defined in the theorem in section 11.1. You should end up with the answer

$$\pi_a = \frac{\beta}{\alpha+\beta} \qquad \pi_b = \frac{\alpha}{\alpha+\beta}.$$

Question 3. First identify the state space, i.e. the set of possible values for x. Looking at the code, we see that x can only ever be an integer in $\{0, 1, \dots, 9\}$, so this is the state space. Next, draw arrows to indicate transitions between states. Make sure that at every node you draw, the probabilities on all outgoing edges sum up to one. You don't need to draw every state in your state space diagram: just show typical states, and also edge cases, as in the diagram in section 11.

Your code should first of all create the transition matrix. Start with `numpy.zeros((10,10))`, and then fill in the values according to your state space diagram. Then give the code from exercise 11.1.1 in lecture notes. For the exam you don't need to remember the syntax, but you do need to be able to describe it in enough detail for a first-year undergraduate to implement.

Question 4. This is a hitting probability question, just like example 10.3.2 from lecture notes. The only tricky bit is working out the transition matrix. For the matrix

$$\mathbb{P}(X_{n+1} = j \mid X_n = i) = \mathbb{P}(j = i + I - R) = \mathbb{P}(I = R + j - i)$$

where I is the number of newly infected, and R is the number of recoveries. Use the law of total probability, conditioning on R.

You should implement this in Python. Unless you implement it yourself, and do the sanity check that the rows of your transition matrix sum to one, it's very easy to make a mistake! For $r = 1.1$ and $d = 14$, cutting the state space down to $\{0, \dots, 200\}$ is sufficient.

Question 5. When a question asks "show that π is a stationary distribution", you don't have to set about trying to *solve* the equations

$$\pi_x = \sum_y \pi_y P_{yx} \quad \text{for all } x.$$

You just need to *verify* that the π you're given does indeed solve these equations. For a simple state space like the one in this question, for a given x, the transition probability P_{yx} is zero for most states y, so it's easy to verify. Write out one equation for π_0, and then write out another equation for π_x for a generic $x > 0$.

You should also show that π is indeed a distribution! In this question, to show that $\sum_{n=0}^{\infty} \pi_n = 1$, either remember the formula for a geometric series ($\sum_{n=0}^{\infty} ar^n = 1/(1-r)$ for $|r| < 1$), or spot that π_n is the p.m.f. for a Geometric distribution and must necessarily sum to 1. (There are two flavours of the Geometric distribution. Wikipedia lists both.)

Question 6. As for question 5, you don't need to solve the stationarity equations, you just need to verify that they are satisfied, in other words that $\pi_x = \sum_y \pi_y P_{yx}$. Simply substitute in $\pi_y = \pi_x P_{xy} / P_{yx}$, which comes from detailed balance. (Then, adjust your reasoning to cope with pairs x, y where $P_{yx} = 0$.)

For the random walk: first write out the transition probability: $P_{xy} = 0$ if there's no $x \leftrightarrow y$ edge, and $P_{xy} = 1/n_x$ otherwise, where n_x is the number of edges incident at vertex x. In indicator notation, $P_{xy} = 1_{x \leftrightarrow y} / n_x$. Then, see if you can spot a solution to the detailed balance equation.

Question 7. The causal diagram was discussed in section 10.1. It is

$$\begin{tikzcd}
X_0 \arrow[r] \arrow[d] & X_1 \arrow[r] \arrow[d] & X_2 \arrow[r] \arrow[d] & \cdots \\
Y_0 & Y_1 & Y_2 &
\end{tikzcd}$$

Part (b). Follow the strategy from section 10.4. The third equation takes some cleverness. Try a partial expansion of the conditional probability, $\Pr(x_n \mid h, y_n) = \Pr(x_n, y_n \mid h) / \Pr(y_n \mid h)$.

Part (c). Let $\pi^{(n)}$ be the probability vector at timestep n. Compute $\pi^{(0)}$ from the first equation. Then, iteratively apply the next two equations, to compute $\pi^{(n)}$ from $\pi^{(n-1)}$. Your implementation should use two matrices, $P_{ij} = \mathbb{P}(X_n = j \mid X_{n-1} = i)$ and $Q_{xy} = \mathbb{P}(Y_n = y \mid X_n = x)$. The first is the transition matrix that we're used to from Markov Chains, and the second is called the emission matrix.

Supplementary questions

These questions are not intended for supervision (unless your supervisor directs you otherwise).

Question 8. The code from question (d) can fail with a divide-by-zero error. This is undesirable in production code! One way to fix the problem is to modify the Markov model to include a 'random teleport'—to express the idea 'OK, our inference has gone wrong somewhere; let's allow our location estimate to reset itself'. We can achieve this mathematically with the following model: with probability $1-\varepsilon$ generate the next state as per line 9, otherwise pick the next state uniformly from $\{0, 1, \dots, \texttt{MAX\_STATE}\}$. Modify your code from question (c) to reflect this new model, with $\varepsilon = 0.01$.

Alternatively, we could fix the problem by changing the model to express 'OK, this reading is glitchy; let's allow the code to discard an impossible reading'. How might you change the Markov model to achieve this?

Question 9. The Markov model for motion from question 3 is called a *simple random walk (with boundaries)*; it chooses a direction of travel independently at every timestep. This is not a good model for human movement, since people tend to head in the same direction for a while before changing direction.

(a) Let $V_n \in \{-1, 0, 1\}$ be a Markov chain: let $V_{n+1} = V_n$ with probability 0.9, and let V_{n+1} be chosen uniformly at random from $\{-1, 0, 1\}$ with probability 0.1. Draw a state space diagram for this Markov chain.

(b) Interpret V_n as the velocity of our moving object at timestep n, and let $X_{n+1} = \max(0, \min(9, X_n + V_n))$. Update your code from question 3 to reflect this model.

Question 10 (Google PageRank). Consider a directed acyclic graph representing the web, with one vertex per webpage, and an edge $v \to w$ if page v links to page w. Consider a random web surfer who goes from page to page according to the algorithm

```
d = 0.85
def next_page(v):
    neighbours = list of pages w such that v → w
    a = random.choice(['follow_link','teleport'], p=[d,1−d])
    if a=='follow_link' and len(neighbours) > 0:
        return random.choice(neighbours)
    else:
        V = list of all web pages
        return random.choice(V)
```

This defines a Markov chain. Explain why the chain is irreducible. Show that the stationary distribution π solves

$$\pi_v = \frac{1-d}{|V|} + d \sum_{u:u\to v} \frac{\pi_u}{|\Gamma_u|}$$

where $|V|$ is the total number of web pages in the graph, and $|\Gamma_u|$ is the number of outgoing edges from u.

Compute the stationary distribution for this random web surfer model, for the graph in lecture notes Example 10.3.2. Repeat with $d = 0.05$. What do you expect as $d \to 0$? What do you expect if $d = 1$?

The equation for π_v defines a scaled version of PageRank, Google's original method for ranking websites.

Question 11 (Drift models). For a random model such as the epidemic in question 4, we can approximate it with a deterministic model $(x_0, x_1, \dots)$ by simply taking the expectation:

$$x_{n+1} = \mathbb{E}(X_{n+1} \mid X_n = x_n).$$

Write down a recurrence relation for x_n, and solve it. Run the random simulator several times, each time with initial state $x_0 = 100$, and plot the outcomes. Also plot the solution to the deterministic model. What do you notice?

Question 12 (Thompson sampling). In reinforcement learning, an agent typically has a choice between exploring possible moves versus exploiting the best moves that have been learnt so far. Here is a concrete example:

A compulsive gambler has a choice of two machines to play. The first has probability θ_1 of paying out, the second has probability θ_2, and all payouts are £10. The gambler doesn't know the values of θ_1 and θ_2, so treats them as unknown parameters, both with uniform prior distributions. The gambler adopts the following strategy (known as *Thompson sampling*) for choosing which machine to play:

1. Find the posterior distribution of (Θ_1, Θ_2) given the number of wins and losses on each machine so far;
2. Generate a single sample (θ_1, θ_2) from this distribution;
3. Play machine 1 if $\theta_1 \geq \theta_2$, and play machine 2 otherwise;
4. Repeat.

After n plays, let $w_1^{(n)}$ be the number of wins on machine 1, $\ell_1^{(n)}$ the number of losses, and $w_2^{(n)}$ and $\ell_2^{(n)}$ likewise for machine 2.

(a) Find the posterior distribution of (Θ_1, Θ_2) given $(w_1^{(n)}, \ell_1^{(n)}, w_2^{(n)}, \ell_2^{(n)})$.

(b) Let $X_n = (w_1^{(n)}, \ell_1^{(n)}, w_2^{(n)}, \ell_2^{(n)})$. Explain why $(X_0, X_1, \dots)$ is a Markov chain, and sketch the state space diagram.

(c) Implement the Thompson sampling strategy, and simulate it for the case where the true values are $\theta_1 = 0.1$ and $\theta_2 = 0.05$. Plot the posterior histograms for Θ_1 and Θ_2 after 1, 50, 100, 500, 1000 plays.

(d) Describe your findings. How does Thompson sampling resolve the explore/exploit tradeoff?