Math087 - Review for Quiz 2

George McNinch

quiz-date 2025-04-09

1 quiz 2

A. *Summary*

Since the first quiz, here are some of the modeling techniques we've discussed

1. Graph models, including those related to `max flow/min cut`. (e.g. the *MBTA* example) and to *matching in bipartite graphs* and questions concerning the existence of *perfect matching* (e.g. in the example pairing graduates and jobs)
2. Use of *finite state machines* as model components, and related matrices. This includes for example our *population model* where the edge labels in the state machine reflect morbidity and fertility
3. *Markov processes*, again involving state machines and (stochastic) matrices. This includes for example our stock market example.
4. *Monte Carlo methods* using random (or pseudo-random) numbers for simulations. This includes our example of a fish-tank retailer.

B. *Problems and questions.*

1. `max-flow / min-cut`

   Consider a (finite) weighted, directed graph G with source node `s` and terminal node `t`. To find the `min-cut` for the graph, we must consider all possible partitions of the nodes into an s-group and a t-group. For each such partition, we compute the `cut-value`, namely the sum of the labels on each of the directed edges connecting a node in the `s`-group with a node in the `t`-group.

   The smallest such `cut-value` is the `min-cut` for G.

   It turns out that `min-cut` is the solution to a linear program which is dual to the `max-flow` linear program for G.

   a. What is the objective function for the `max-flow` linear program?

   b. What does *strong duality* say about the relationship between `max-flow` and `min-cut`?

2. *Bi-partite graphs and matching*

   Consider a bipartite graph G with vertices $U \cup V$ and edges E.

   a. What is meant by a *matching* for G? What does it mean to say that a matching is *maximal*? *perfect*?

b. For a subset $X \subseteq U$ of vertices, the neighborhood $N(X)$ is the subset of V defined by

$$N(X) = \{v \in V \mid [u, v] \in E \quad \text{for some } u \in U\}$$

If $X = \{a, b\}$ and $N(X) = \{c\}$, can G have a perfect matching? Why or why not?

c. Give the statement of *Hall's Matching Theorem* (aka *Hall's Marriage Theorem*).

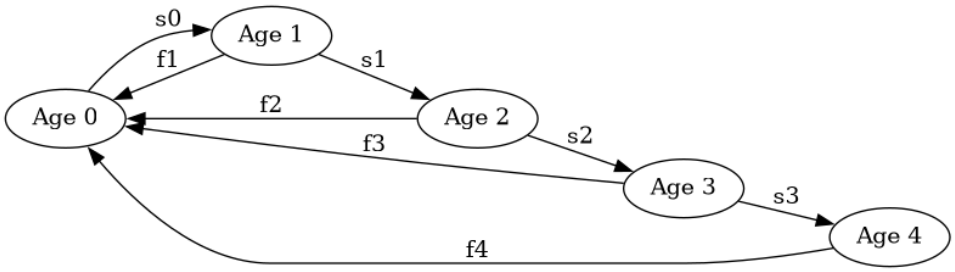
3. *state machines*

Let's recall our population model. We'll model a certain species of insects with a maximum life span of 4 years, and we'll suppose for `i=0,1,2,3` that for an individual insect of age `i` the probability of survival is given by the number `s_i`, and for `i=1,2,3,4` that for an individual of age `i` the probability of reproduction is given by the number `f_i`. We represent the population of these insects by a vector

```
import numpy as np
p = np.array([ p0, p1, p2, p3, p4 ])
```

where `pi` represents the population of these insects having age `i`.

a. The following directed graph represents a finite state machine modeling the population.



Find a `5 x 5` matrix `M` so that if `p` represents the population in a given year, then `M @ p` represents the population in the subsequent year.
(Remember that `M @ p` is the `numpy` syntax for the product of `M` and the column vector `p`).

b. If `p` represents the population in a given year, what matrix calculation must be carried out to find the population after 10 subsequent years?

c. If

```
ones = np.array([ 1, 1, 1, 1, 1 ])
```

explain what is computed by the limit as $n \to \infty$ of the expression

```
ones @ np.linalg.matrix_power(M,n) @ p
```

4. *Markov processes*

a. What is meant by a *stochastic matrix*?

b. A Markov process is described by a transition diagram (which we assume to be finite). The rows and columns of the corresponding matrix `P` are labeled by the vertices of the diagram, and for two vertices `a,b` the entry `P[a][b]` is given by the edge label of the transition diagram, which represents the probability that the system will change from state `a` to state `b`.
If `vert` is the full list of vertices of the transition diagram, explain why

```
sum([ P[a][v] for v in vert])
```

is equal to 1. Must

```
sum([ P[v][a] for v in vert])
```

be equal to 1?

c. Assume that the transition diagram is strongly connected and aperiodic, and let `P` be the corresponding stochastic matrix. According to the Perron-Frobenius Theorem, which of the following statements is correct:

- every eigenvalue of `P` is smaller than 1.
- `P` has `lambda = 1` as an eigenvalue, all other eigenvalues `mu` of `P` have the property that `|mu| < 1`, and the matrix `P - I` has rank `n -1` if `P` is an `n x n` matrix.
- `P` has `lambda = 1` as an eigenvalue, all other eigenvalues `mu` of `P` have the property that `|mu| < 1`, and the dimension of the 1-eigenspace is equal to 1.

d. Assume that the transition matrix is strongly connecte and aperiodic, and let `v` be a 1-eigenvector normalized so that

```
ones = np.vector(n*[1]) # = [1,1,...,1]

ones @ v == 1
```

If `a` is a vertex of the transition diagram, explain why the long-term probability of the system being in the state corresponding to `a` is given by the value `v[a]`.

5. *Monte Carlo methods*

Suppose you have to catch a bus early every weekday morning. On a good day, you catch the fast bus that arrives first. Sometimes you miss the fast bus and must take the slow bus. And sometimes you miss both busses. After some observations, you determine that on average, you catch the fast bus twice a week, you catch the slow bus twice a week, and you miss the bus once a week.

a. What is the probability that you miss the bus on any given weekday?

b. What is the probability that you catch the fast bus three days in a row?

Consider the following code which yields a list of events corresponding to 3 weeks of weekdays; on each weekday, you either catch the fast bus – marked as 'f' –, you catch the slow bus – marked as 's' – or you 'miss' the bus – marked as 'm').

```
import numpy as np
from pprint import pprint
rng = np.random.default_rng()

def wait_for_bus(fast = 2./5,slow = 2./5):
    return str(rng.choice(['f','s','m'],p=[fast,slow,1-fast-slow]))

pprint([ wait_for_bus() for _ in range(15) ])
```

```
['s', 's', 'f', 's', 'm', 's', 'f', 's', 'f', 's', 'm', 'm', 's', 'f', 's']
```

c. In the given output, do we say the expected number of *misses*?

d. Do we catch the fast bus the expected number of times?