Rabbits and Wolves

In general, recurrence relations can involve several simultaneous definitions.

Classic example: population models.

- $W_n = \#$wolves in year n
- $R_n = \#$rabbits in year n

A simple model:

$$W_n = aW_{n-1} + bR_{n-1}$$
$$R_n = cR_{n-1} - dW_{n-1}$$

- a depends on number of offspring wolves have, fraction of wolves that die of old age, and ways other than starvation.
- b depends on what sources of food wolves have besides rabbits. Roughly speaking, more rabbits means more wolves.
- c depends on number of offspring that rabbits have, how they might die besides wolves.
- d depends on many rabbits are eaten by each wolf.

This is a very simple-minded model, but its solutions have the right qualitative behavior: oscillations of populations.

The continuous version of these problems is also studied using differential equations.

In this course, we focus on discrete recurrences (difference equations) involving only one sequence.

Linear Homogeneous Recurrence Relations

A *linear homogeneous recurrence relation* of *order* (or *degree*) k with constant coefficients has the form:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k},$$

where $c_1, \ldots, c_k$ are real numbers, $c_k \neq 0$.

- *linear*: $a_n =$ sum of multiples of previous terms
- *homogeneous*: no terms except those involving a_j's
- *order* k: lowest term is a_{n-k}

Examples:

- $a_n = a_{n-1}a_{n-2}$: homogeneous, nonlinear, order 2
- $a_n = a_{n_1}^2$: homogeneous, nonlinear, order 1
- $a_n = \log(a_{n-1})$: homogeneous, nonlinear, order 1
- $a_n = a_{n-1} + 2a_{n-2} + n^3 + 2$: linear, nonhomogeneous, order 2
- $a_n = na_{n-3}$: linear, homogeneous, order 3, nonconstant coefficients
- $f_n = f_{n-1} + f_{n-2}$ (Fibonacci numbers): linear, homogeneous, order 2, constant coefficients.

Unique determination

Theorem 0: Given a recurrence relation of order k $a_n = f(a_{n-1}, \ldots, a_{n-k})$ and initial conditions $a_0, \ldots, a_{k-1}$, the whole sequence $\{a_n\}$ is uniquely determined.

Proof: By strong induction that a_n, is uniquely determined for all n.

Base cases: True for $a_0, \ldots, a_{k-1}$ by assumption.

Inductive step: If $n \geq k$, then $a_{n-1}, \ldots, a_{n-k}$ are uniquely determined by induction hypothesis, and $a_n = f(a_{n-1}, \ldots, a_{n-k})$ so a_n is uniquely determined.

Note: If only some of $a_0, \ldots, a_{k-1}$ given, then the sequence is not uniquely determined in general.

Solutions to Homogeneous Equations

Theorem 1: If $\{b_n\}$ and $\{b'_n\}$ are solutions to the linear homogeneous equation with constant coefficients

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}, \qquad (1)$$

then $\{Bb_n + B'b'_n\}$ is also a solution, where B and B' are arbitrary constants.

Proof: Since $\{b_n\}$ is a solution to (1), it must be the case that

$$b_n = c_1 b_{n-1} + c_2 b_{n-2} + \cdots + c_k b_{n-k},$$

Multiplying both sides by B, it follows that

$$Bb_n = c_1 Bb_{n-1} + c_2 Bb_{n-2} + \cdots + c_k Bb_{n-k} \qquad (2)$$

Similarly,

$$B'b'_n = c_1 B'b'_{n-1} + c_2 B'b'_{n-2} + \cdots + c_k B'b'_{n-k} \qquad (3)$$

Adding (2) and (3) gives us

$$Bb_n + B'b'_n = c_1(Bb_{n-1} + B'b'_{n-1}) + \cdots + c_k(Bb_{n-k} + B'b'_{n-k})$$

Thus, $\{Bb_n + B'b'_n\}$ is a solution to (1).

- Text proves the result for second-order recurrences only. The same proof works in general.
- The set of solutions forms a *vector space*.

First-order recurrences

Solving linear, homogeneous first-order recurrences is easy:
If $a_n = c_1 a_{n-1}$, then $a_n = c_1^n a_0$.
Given a_0, a_n is uniquely determined.

This is the only solution.

Second-order recurrences

Consider second-order linear, homogeneous recurrences with constant coefficients. They have the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}. \qquad (4)$$

(Ignore the setting of a_0 and a_1 for now.)

Idea: assume that the solution has the form $a_n = r^n$. Then must have

$$r^n = c_1 r^{n-1} + c_2 r^{n-2}.$$

One solution: $r = 0$. I.e., $a_n = 0$ for all n. Otherwise:

$$r^2 = c_1 r + c_2.$$

$r^2 - c_1 r - c_2 = 0$ is the *characteristic equation* of this recurrence relation.
If r_1 is a solution to this equation, then it's easy to check that $a_n = r_1^n$ solves the recurrence relation. Using Theorem 1, the most general result is:

Theorem 2: Suppose r_1 and r_2 are the roots of $r^2 - c_1 r - c_2 = 0$, the characteristic equation of (4).

(a) If $r_1 \neq r_2$, then $\{a_n\}$ is a solution to (4) iff $a_n = Ar_1^n + Br_2^n$, for some constants A and B.

(b) If $r_1 = r_2$, then $\{a_n\}$ is a solution to (4) iff $a_n = Ar_1^n + Bnr_1^n$.

Proof: Suppose that $r_1 \neq r_2$. It's easy to check that $a_n = r_1^n$ is a solution, assuming no constraints on a_0, a_1.
Proof is by strong induction:
Base case — $n = 2$: $a_2 = c_1 a_1 + c_2 a_0$ since $r_1^2 = c_1 r_1 + c_2$

Inductive step: $a_n = c_1 a_{n-1} + c_2 a_{n-2} = c_1 r_1^{n-1} + c_2 r_1^{n-2} = r_1^n$ since r_1 is a solution to the characteristic equation.

Similarly, $a_n = r_2^n$ is a solution. The fact that $Ar_1^n + Br_2^n$ is a solution follows from Theorem 1.

If $r_1 = r_2$, must check that $a_n = nr_1^n$ is a solution.
If $r_1 = r_2$, then $r^2 - c_1 r - c_2 = (r - r_1)^2 = r^2 - 2r_1 r + r_1^2$

- $c_1 = 2r_1$ and $c_2 = -r^2$.

Now plug in nr_1^n for a_n and use induction:

$$\begin{aligned} nr_1^n &= 2(n-1)r_1^n - (n-2)r_1^n \\ &= 2r_1(n-1)r^{n-1} - r_1^2(n-2)r_1^{n-2} \\ &= c_1a_{n-1} + c_2a_{n-2} \end{aligned}$$

Again, r_1^n is a solution so, by Theorem 1, $Ar_1^n + Bnr_1^n$ is a solution.

Why are these the only solutions? Given any initial conditions for a_0 and a_1, can solve for A and B.

- If $r_1 \neq r_2$: $a_0 = A + B$, $a_1 = Ar_1 + Br_2$.
- If $r_1 = r_2$: $a_0 = A$, $a_1 = (A + B)r_1$

Notes:

- Solutions r_1, r_2 may be complex numbers
- Same idea works to solve linear, homogeneous equations of order k with constant coefficients.

Fibonacci Revisited

For Fibonacci: $f_n = f_{n-1} + f_{n-2}$, $f_0 = f_1 = 1$

Characteristic equation: $r^2 - r - 1 = 0$:

Solutions: $(1 \pm \sqrt{5})/2$.

General solution to recurrence:

$$f_n = A\left(\frac{1+\sqrt{5}}{2}\right)^n + B\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Since $f_0 = f_1 = 1$, must have

- $A + B = f_0 = 1$
- $A\left(\frac{1+\sqrt{5}}{2}\right) + B\left(\frac{1-\sqrt{5}}{2}\right) = f_1 = 1$

Tedious calculation shows:

- $A = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)$
- $B = -\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)$

Therefore:

$$f_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right]$$

Coverage of Final

- everything covered by first two prelims
  - slight emphasis on more recent material
- probability: 6.1–6.5 (but not Poisson and inverse binomial distribution)
  - basic definitions: probability space, events
  - conditional probability, independence, Bayes Thm.
  - random variables, uniform + binomial distribution
  - expected value and variance
- logic: 7.1–7.4, 7.6; *not* 7.5
  - translating from English to propositional (or first-order) logic
  - truth tables and axiomatic proofs
  - algorithm verification
  - first-order logic
- recurrence relations:
  - setting up equations (5.2)
  - basic definitions (5.4)
  - second-order homogeneous linear equations (5.5)

Ten Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)
- **Induction**: Recognize it in all its guises.
- **Exemplification**: Find a sense in which you can try out a problem or solution on small examples.
- **Abstraction**: Abstract away the inessential features of a problem.
  - One possible way: represent it as a graph
- **Modularity**: Decompose a complex problem into simpler subproblems.
- **Representation**: Understand the relationships between different possible representations of the same information or idea.
  - Graphs vs. matrices vs. relations
- **Refinement**: The best solutions come from a process of repeatedly refining and inventing alternative solutions.
- **Toolbox**: Build up your vocabulary of abstract structures.

- **Optimization**: Understand which improvements are worth it.
- **Probabilistic methods**: Flipping a coin can be surprisingly helpful!

Connections: Random Graphs

Suppose we have a random graph with n vertices. How likely is it to be connected?

- What is a *random* graph?
  - If it has n vertices, there are $C(n, 2)$ possible edges, and $2^{C(n,2)}$ possible graphs. What fraction of them is connected?
  - One way of thinking about this. Build a graph using a random process, that puts each edge in with probability 1/2.

- Given three vertices a, b, and c, what's the probability that there is an edge between a and b and between b and c? 1/4
- What is the probability that there is no path of length 2 between a and c? $(3/4)^{n-2}$
- What is the probability that there is a path of length 2 between a and c? $1 - (3/4)^{n-2}$
- What is the probability that there is a path of length 2 between a and every other vertex? $> (1-(3/4)^{n-2})^{n-1}$

Now use the binomial theorem to compute $(1-(3/4)^{n-2})^{n-1}$

$$\begin{aligned} & (1-(3/4)^{n-2})^{n-1} \\ = \ & 1-(n-1)(3/4)^{n-2} + C(n-1,2)(3/4)^{2(n-2)} + \cdots \end{aligned}$$

For sufficiently large n, this will be (just about) 1.

Bottom line: If n is large, then it is almost certain that a random graph will be connected.

Theorem: [Fagin, 1976] If P is *any* property expressible in first-order logic, it is either true in almost all graphs, or false in almost all graphs.

This is called a *0-1 law*.

Connection: First-order Logic

Suppose you wanted to query a database. How do you do it?

Modern database query language date back to SQL (structured query language), and are all based on first-order logic.

- The idea goes back to Ted Codd, who invented the notion of relational databases.

Suppose you're a travel agent and want to query the airline database about whether there are flights from Ithaca to Santa Fe.

- How are cities and flights between them represented?
- How do we form this query?

You're actually asking whether there is a path from Ithaca to Santa Fe in the graph.

- This fact cannot be expressed in first-order logic!