Performance Evaluation: Markov Models, revisited

CSCI 8710
E. Kraemer

Working with Markov models

- State space enumeration
- State transition identification
- Parameterization

- Calibration and solution

Example 1: Random walk through England

- Young man spending one year in England; checks in with Mom at 3 pm daily

1. **State space enumeration**
2. **State transition identification**
3. **Parameterization**

State space enumeration

Mom detects 4 states:

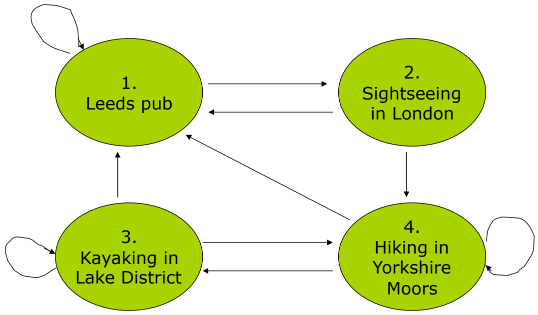
- Drinking in Leeds pub
- Sightseeing in London
- Kayaking in Lake District
- Hiking in Yorkshire moors

State space enumeration

State transition identification

- If in Leeds one day, then next day:
  - Sightseeing in London
  - Again in Leeds pub
- If in London one day, then next day:
  - In Leeds pub
  - Hiking in Yorkshire Moors
- If kayaking in Lake District one day, then next day:
  - Still kayaking in Lake district
  - Hiking in Yorkshire Moors
  - In Leeds pub
- If hiking in Moors one day, then next day:
  - Hiking in Moors
  - In Leeds pub
  - Kayaking in Lake District

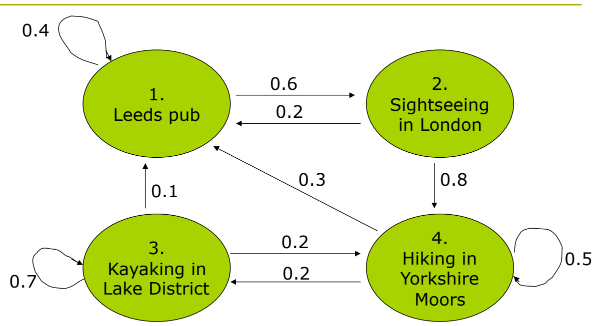
State transistion identification



Parameterization

- If in Leeds one day, then next day:
  - Sightseeing in London (60%)
  - Again in Leeds pub (40%)
- If in London one day, then next day:
  - In Leeds pub (20%)
  - Hiking in Yorkshire Moors (80%)
- If kayaking in Lake District one day, then next day:
  - Still kayaking in Lake district (70%)
  - Hiking in Yorkshire Moors (20%)
  - In Leeds pub (10%)
- If hiking in Moors one day, then next day:
  - Hiking in Moors (50%)
  - In Leeds pub (30%)
  - Kayaking in Lake District (20%)

State transition identification



Questions to be answered:

- Dad wants to know:
  - *What percentage of days is son actually not drinking in Leeds?*
- Relatives in Lake District want to know:
  - *Once he finishes a day of kayaking in the Lake District, how long will it typically be before he returns?*
- Policeman wants to know:
  - *How many days each month can bobbies expect to see son driving to London after drinking in Leeds?*
- Kayak renter wants to know:
  - *How many times per month does son typically visit kayak shop?*
  - *How long does he typically keep kayak checked out?*

Model Solution

- steady state probabilities of being in each state
- Independent of initial state
- Solve system of linear equations, each encoding notion that flow in = flow out

Flow in = flow out

- P1
  - Flow in =
    - 0.4 * P1 +
    - 0.2 * P2 +
    - 0.1 * P3 +
    - 0.3 * P4
  - Flow out =
    - 0.4 * P1
    - 0.6 * P1
  - Can subtract out “self loop”
  - 0.2*P2 + 0.1*P3 + 0.3 * P4 = 0.6 * P1

Flow in = flow out

- P2
  - Flow in = 0.6 * P1
  - Flow out = 0.8 * P2 + 0.2 * P2
  - Since no self-loop, sum of outflows = 1
  - Can write:
    - 0.6 * P1 = P2

Flow in = flow out

- P3:
  - Flow in = 0.2* P4
  - Flow out = 0.1 * P3 + 0.2 * P3
  - 0.2 * P4 = 0.3 * P3
- P4:
  - Flow in = 0.8 * P2 + 0.2 * P3
  - Flow out = 0.3 * P4 + 0.2 * P4
  - 0.8 * P2 + 0.2 * P3 = 0.5 * P4

To solve:

- 0.2 * P2 + 0.1 * P3 + 0.3 * P4 = 0.6 * P1
- 0.6 * P1 = P2
- 0.2 * P4 = 0.3 * P3
- 0.8 * P2 + 0.2 * P3 = 0.5 * P4
- P1 + P2 + P3 + P4 = 1.0
- Drop one of first four (save for a "check"), and solve system of 4 equations in 4 unknowns

Solution

- P1 = 55/208 = 0.2644
- P2 = 33/208 = 0.1586
- P3 = 48/208 = 0.2308
- P4 = 72/208 = 0.3462

Answering the questions …

- Dad wants to know:
  - *What percentage of days is son actually not drinking in Leeds*?
  - P1 = 0.2644
  - So, NOT(P1) = 1 – 0.2644 = 0.74
  - 74% of the time not drinking in Leeds

Answering the questions

- Relatives in Lake District want to know:
  - *Once he finishes a day of kayaking in the Lake District, how long will it typically be before he returns*?
  - Kayaking at Lake is state 3
  - P3 = 0.2308 = steady state probability of being in state 3
  - Mean time between entering state is inverse: 1/0.2308 = 4.33 days
  - Start to start is 4.33 days
  - Finish to start is 3.33 days
  - 3.33 days

Answering the questions

- Policeman wants to know:
  - *How many days each month can bobbies expect to see son driving to London after drinking in Leeds?*
  - *Drinking in Leeds = P1 = 0.2644*
  - *30 days * 0.2644 = 7.93 days drinking*
  - *P1 -> P2 = 0.6*
  - *7.93 * 0.6 = 4.76 days per month*

- Kayak renter wants to know:
  - *How many times per month does son typically visit kayak shop?*
  - *How long does he typically keep kayak checked out?*
  - *P3 entered only from P4*
  - *P4 = 0.3462*
  - *0.3462 * 30 days = 10.39 days/month*
  - *P4 -> P3 = 0.2*
  - *10.39 * 0.2 = 2.08 times/month*

- P3 = prob kayaking = 0.2308
- 30 * 0.2308 = 6.92 days/month
- 6.92 days/2.08 new visits = 3.33 days/visit

Example #2: Database server support

- System with one CPU and two disks
  - Users remotely access server: login, perform DB transactions, logout
  - Max of 2 simultaneous users; high demand; can assume consistent 2 simultaneous users
  - Each transaction alternates between using CPU and disk
  - Disks are different speeds: 2X and 1X
  - D_cpu = 10 sec
  - File access probability is equal across disks
  - D_fast = 15 sec
  - D_slow = 30 sec

Questions to be answered:

- User wants to know:
  - *Expected response time*
- Sys admin wants to know:
  - *Utilization of each resource*
- Company pres wants to know:
  - *What happens to performance if number of users doubles?*
- Company nay-sayer wants to know:
  - *Given that fast disk is about to fail and all files will have to be moved to slow disk, what will the new response time be?*

State space enumeration

- Two users, each of which can be at any one of three devices
- Notation: (X, Y, Z)
  - X = number of users at CPU
  - Y = number of users at fast disk
  - Z = number of users at slow disk

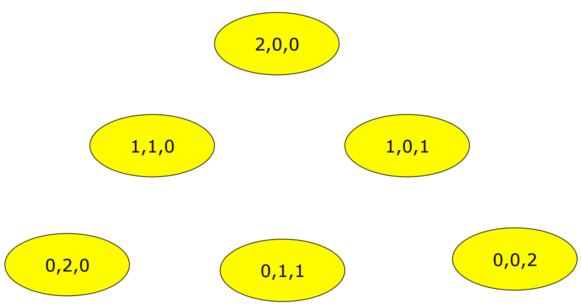
Other notations?

- (CPU, CPU)
- (CPU, FD)
- (CPU, SD)
- (FD, FD)
- (FD, SD)
- (FD, CPU)
- (SD, SD)
- (SD, CPU)
- (SD, FD)
- … more states, some statistically identical (FD, SD) and (SD, FD), etc. …
- Model more complex, might be needed for multiclass model, but not for this example …

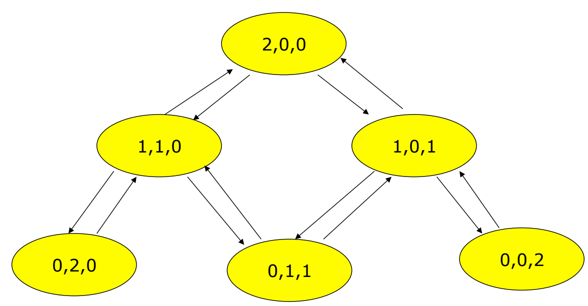
State space enumeration

- (2,0,0)
  - Both users at CPU
- (1,1,0)
  - One user at CPU, one at fast disk
- (1,0,1)
  - One user at cpu, one at slow disk
- (0,2,0)
  - Two users at fast disk
- (0,1,1)
  - One user at fast disk, one user at slow disk
- (0,0,2)
  - Two users at slow disk

State space enumeration



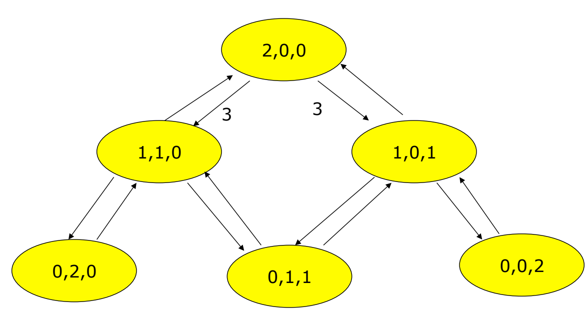
State transition identification



Parameterization

- Start with (2,0,0):
  - CPU is actively working
  - D_cpu is 10 seconds : 6 t/m
    - Of those at (2,0,0), half go to fast, half to slow

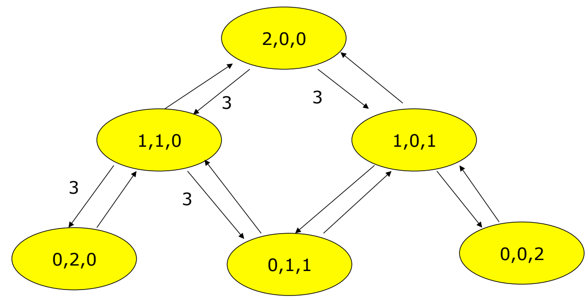
State transition identification



Parameterization

- Consider (1,1,0)
  - One user executing at CPU
  - One user at fast disk
  - Still, rate from CPU is 6, with half going to fast(3) and half going to slow (3), so …

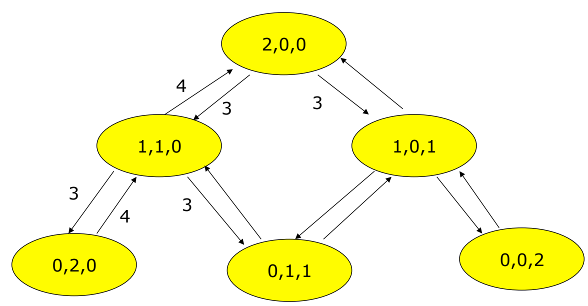
State transition identification



Parameterization

- Fast disk satisfies user requests at rate of 4 t/m (D_fast = 15 sec)
- All users at fast disk next visit CPU, so …

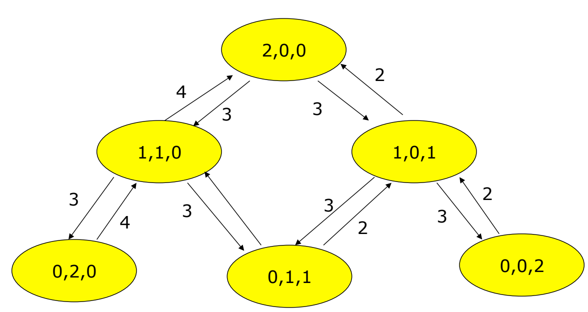
State transition identification



Parameterization

- Similar logic for slow disk side …

State transition identification



Model Solution

- Flow in = Flow out
- P(2,0,0) = 4 * P(1,1,0) + 2*P(1,0,1)
- … construct remainder as in-class exercise

Solution

- P(2,0,0) = 16/115 = 0.1391
- P(1,1,0) = 12/115 = 0.1043
- P(1,0,1) = 24/115 = 0.2087
- P(0,2,0) = 9/115 = 0.0783
- P(0,1,1) = 18/115 = 0.1565
- P(0,0,2) = 36/115 = 0.3131

Interpreting the model

- User wants to know:
  - *Expected response time*

Interpreting the model

- User wants to know *expected response time*
  - *R = M/X_O – Z*
  - *Z = 0*
  - *M = users in system = 2*
    - *So, R = M/X_0*
  - *X = throughput = utilization * service rate*
    - *U_cpu = P(2,0,0) + P(1,1,0) + P(1,0,1)*
    - *U = 0.1391 + 0.1043 _ 0.2087 = 0.4521*
    - *X = 0.4521 bs/ts * 1 t/10 bs = 0.04521 t/s*
    - *R = 2/0.04521 = 44.24 t/sec*

Interpreting …

- Sys admin wants to know:
  - *Utilization of each resource*

Interpreting the model …

- Sys admin wants to know *utilization of each resource*
    - *U_cpu = P(2,0,0) + P(1,1,0) + P(1,0,1)*
    - *U = 0.1391 + 0.1043 _ 0.2087 = 0.4521*

    - *U_fast = P(1,1,0) + P(0,2,0) + P(0,1,1)*
    - *U_fast = 0.1043 + 0.0783 + 0.1565 = 0.3391*

    - *U_slow = P(1,0,1) + P(0,1,1) + P(0,0,2)*
    - *U_slow = 0.2087 + 0.1565 + 0.3131 = 0.6783*

Interpreting the model

- Company pres wants to know:
  - *What happens to performance if number of users doubles*?

Interpreting the model …

- Company pres wants to know w*hat happens to performance if number of users doubles*?
  - … build model with 4 active users instead of 2
  - … now have 15 states in diagram
  - … solve for throughput based on new utilizations, derived from state probabilities
  - … calculate response time ..
    - X_0 increases from 2.7126 t/min to 3.4768 t/min
    - R increases from 44.24 sec to 69.03 sec

Interpreting …

- Company nay-sayer wants to know:
  - *Given that fast disk is about to fail and all files will have to be moved to slow disk, what will the new response time be?*

- Company nay-sayer wants to know: *Given that fast disk is about to fail and all files will have to be moved to slow disk, what will the new response time be?*
  - *… now, solve model with just two devices .. CPU and slow disk .. Only three states … derive probabilities*

    P(2,0) = 0.0769

    P(1,1) = 0.2308

    P(0.2) = 0.6923

  X_0 = 1.8462 t/min, R = 65 sec