Análise e Modelagem de Desempenho de Sistemas de Computação:

``Markov Models in Performance Evaluation ’’

Virgilio A. F. Almeida

1st semester 2009
Week #7

Computer Science Department
Federal University of Minas Gerais
Brazil
Markov Models – Chapter 10

– Introduction
– Modeling Context
– Motivating Examples
– Model Construction
– Model Solution
– Model Interpretation
– Model Assumptions and Limitations
– Generalized Birth-Death Models
– Beyond the Basics
Modeling Context
Motivating Examples

Figure 10.2. England example.
Motivating Examples

- The lad is always doing something in one of four locations: drinking in a Leeds pub, sightseeing in London, kayaking in the Lake District, or hiking in the Yorkshire moors.

- If the lad is in a Leeds pub, he is either likely to go sightseeing in London the following day (60%), or he will still be found in a Leeds pub (40%).

- If the lad is in London, he is likely to be found in a Leeds pub the following day (20%) or will decide to go hiking in the Yorkshire moors (80%).

- Once found in the Lake District, there is very good chance that the lad will still be found kayaking the following day (70%), but there is a possibility that he will next be found hiking the moors (20%) or back in a Leeds pub (10%).

- When found hiking in the moors, there is a good chance that he will still be hiking the following day (50%). However, he sometimes goes to a Leeds pub (30%) and sometimes decides to go kayaking in the Lake District (20%) the following day.

Simply knowing these patterns, the mother finds herself having to answer certain questions of others.

- **Father’s question:** What percentage of days is the son actually not drinking in Leeds?

- **Lake District relatives’ question:** Once the son finishes a day of kayaking in the Lake District, how long will it typically be before he returns?

- **Policeman’s question:** How many days each month can the bobbies expect to see the son driving to London after drinking in Leeds?

- **Kayak renters’ question:** How many visits each month does the son typically visit their shop and typically how long does the son keep their kayak out each visit?
Model Construction

Random Walk Through England: Model Construction

In order to construct the model, the first task is to enumerate all the possible states in which the system might find itself. This is the system’s "state space." In this example, there are four possible states in which the mother may find the lad:

- Drinking in a Leeds pub (state 1)
- Sightseeing in London (state 2)
- Kayaking in the Lake District (state 3)
- Hiking in the Yorkshire moors (state 4)

This state space is mutually exclusive and collectively exhaustive. This means that it is not possible to be in more than one state at time (i.e., mutually exclusive) and that it is not possible to be in any state other than in one of the identified states (i.e., collectively exhaustive).

The second task is to identify the state transitions. These are the "single step" transitions. That is, if the system (i.e., the lad) is in state, identify those possible states that the system may find itself during the immediately following time step (i.e., the next day). In current example, the following are the possible state transitions:

- If in state 1 (i.e., drinking in a Leeds pub), the lad may find himself next in state 1 or state 2 (i.e., still drinking in a Leeds pub sightseeing in London).
- If in state 2 (i.e., sightseeing in London), the lad may find himself next in state 1 or state 4 (i.e., drinking in a Leeds pub or hik the Yorkshire moors).
- If in state 3 (i.e., kayaking in the Lake District), the lad may find himself next in states 1, 3, or 4 (i.e., drinking in a Leeds pub, kayaking in the Lake District, or hiking in the Yorkshire moors).
- If in state 4 (i.e., hiking in the Yorkshire moors), the lad may find himself next in state 1, 3, or 4 (i.e., drinking in a Leeds pub, kayaking in the Lake District, or still hiking the Yorkshire moors).
Model Construction

Figure 10.4. Markov model of the England example.
Model Construction

Figure 10.3. Database server example.

Markov Model of the database server example.
Model Solution

The balance equations for Fig. 10.4, for states 1, 2, 3, and 4, respectively, are:

\[(0.2 \times P_2) + (0.1 \times P_3) + (0.3 \times P_4) = 0.6 \times P_1\]
\[0.6 \times P_1 = P_2\]
\[0.2 \times P_4 = 0.3 \times P_3\]
\[(0.8 \times P_2) + (0.2 \times P_3) = 0.5 \times P_4.\]

\[
(4 \times P_{(1,1,0)}) + (2 \times P_{(1,0,1)}) = 6 \times P_{(2,0,0)}
\]
\[
(3 \times P_{(2,0,0)}) + (4 \times P_{(0,2,0)}) + (2 \times P_{(0,1,1)}) = 10 \times P_{(1,1,0)}
\]
\[
(3 \times P_{(2,0,0)}) + (4 \times P_{(0,1,1)}) + (2 \times P_{(0,0,2)}) = 8 \times P_{(1,0,1)}
\]
\[
3 \times P_{(1,1,0)} = 4 \times P_{(0,2,0)}
\]
\[
(3 \times P_{(1,1,0)}) + (3 \times P_{(1,0,1)}) = 6 \times P_{(0,1,1)}
\]
\[
3 \times P_{(1,0,1)} = 2 \times P_{(0,0,2)}
\]

Database example

© 2009 Almeida and Menascé. All Rights
Model Interpretation

Random Walk Through England: Model Interpretation

In Section 10.3, four motivating questions were posed that can now be answered.

- *Father's question:* What percentage of days is the son actually not drinking in Leeds?
  
  *Answer:* 74%. Since the steady state probability of being in state 1 (i.e., drinking in a Leeds pub) is 0.2644 (i.e., 26%), the rest of the time the lad is sightseeing, kayaking, or hiking.

- *Lake District relatives' question:* Once the son finishes a day of kayaking in the Lake District, how long will it typically be before he returns?
  
  *Answer:* 3.33 days. The mean time between entering a particular state (i.e., the state's "cycle time") is the inverse of the steady state probability of being in that state. Since the steady state probability of being in state 3 (i.e., kayaking in the Lake District) is 0.2308, the cycle time between successive entries into state 3 is $1/0.2308 = 4.33$ days. Since it takes one day for the lad to kayak, the time from when he finishes a day of kayaking until he typically starts kayaking again is $4.33 - 1 = 3.33$ days. (Note: To grasp an intuitive sense of this relationship of cycle time being the inverse of the steady state probability, consider a hypothetical five state model where the lad cyclically visits each of the five states for one day each. Since each of the five states is equivalent, the lad is in each state 20% of his time. Thus, the cycle time between successive entries to each state is $1/0.20 = 5$ days and the time between leaving a state until the state is reentered is $5 - 1 - 4$ days.)

- *Policeman's question:* How many days each month can the bobbies expect to see the son driving to London after drinking in Leeds?
  
  *Answer:* 4.76 days. Consider a 30 day month. The steady state probability of being found drinking in a Leeds pub (i.e., state 1) on any particular day is 0.2611. Thus, out of 30 days, $30 \times 0.2611 = 7.83$ days will find the lad drinking. However, since the lad decides to go to London with only probability 0.6 after a day of drinking in Leeds, the bobbies can expect to find the lad on the road to London $7.93 \times 0.6 = 4.76$ days each month.

- *Kayak renters' question:* How many visits each month does the son typically visit their shop and typically how long does the son keep their kayak out each visit?
Model Assumptions

- **Memoryless Assumption:** It is assumed that all the important system information is captured in the state descriptors of a Markov model. That is, simply knowing which state the system is in, uniquely defines all relevant information. This leads to subtleties. For instance, it implies that it doesn’t matter how one arrives (i.e., by which path) to a particular state, only that one is currently in the particular state. Previous history of previous states visited is irrelevant when determining which state will be visited next. This assumption also implies that the length of time that the system is in a particular state, by continually taking self loops, is irrelevant. The only thing that is important in determining which state will be visited next is that the system is in a particular state at the current time. This places a nontrivial burden on choosing an appropriate notation for the state descriptor. For instance, if the jobs have different characteristics, this must be evident from the state descriptor. If the order in which jobs arrive to the systems makes a difference, this, too, must be captured in the state descriptor. Knowing the current state alone is sufficient. This is the defining Markov characteristic and any other information is unnecessary as it applies to the system’s future behavior. That is, previous history can be forgotten. This explains the term “memoryless” as it applies to Markov models.

- **Resulting Limitation:** Because everything must be captured in the state descriptor, Markov models are susceptible to state space explosion. For example, if there are 10 customers at the CPU, each unlike any other customer (i.e., a multi-class model) and the CPU is scheduled first-come-first-serve (i.e., where the order of customer arrivals is important), then the state descriptor of the CPU must contain which customer occupies which position in the queue. This leads to $10! = 3,628,800$ different possibilities/states for the CPU alone. However, if the customers behave similarly (i.e., a single class model) and the CPU is scheduled round robin with a short quanta (i.e., where the order of customer arrivals is not important), then the state descriptor of the CPU can be represented by a single number (i.e., the number of customers present). Having large state spaces implies additional complexity and a potential loss of accuracy.

- **Exponential Assumption:** Without going into depth, the exponential distribution is the only continuous distribution that is memoryless. For instance, suppose the average amount of CPU time required by a customer is 10 seconds. Knowing that the customer has already received 5 seconds worth of CPU time but not yet finished (i.e., previous history, which is irrelevant under the Markov memoryless assumption), the average amount of CPU time still needed is again 10 seconds. This is analogous to flipping a fair coin. The average number of flips to get a head is two. However, knowing that the first flip resulted in a tail, the average number of flips still needed to get a head is again two. Thus, Markov models assume that the time spent between relevant events, such as job arrival times and job service times, is exponentially distributed.

- **Resulting Limitation:** To mitigate the limitation imposed by exponential assumptions, the concept of phases (or stages) can be introduced. For example, again suppose that the average amount of CPU time required by a customer is 10 seconds. By partitioning the total service requirement into two phases of service (i.e., each phase being exponentially distributed with an average of five seconds), the CPU state for each customer can be decoupled into two states. That is, each customer can be in either its first stage of service or its second stage of service. This technique opens up a whole host of other distributions (i.e., not simply exponential) that can be closely approximated. However, the price is again a potential state space explosion since the state descriptors must now contain this additional phase information.
Generalized Birth-Death Models

Figure 10.6. Generalized birth-death state-space diagram.

Following the solution approach of the previous sections, the system of flow balance equations is

\[
\begin{align*}
\text{flow in} &= \text{flow out} \\
\mu_1 P_1 &= \lambda_0 P_0 \\
\lambda_0 P_0 + \mu_2 P_2 &= \lambda_1 P_1 + \mu_1 P_1 \\
&\vdots \\
\lambda_{k-1} P_{k-1} + \mu_{k+1} P_{k+1} &= \lambda_k P_k + \mu_k P_k \\
&\vdots
\end{align*}
\]

After some algebraic manipulation and using the conservation of total probability, \( P_0 + P_1 + P_2 + \cdots = 1 \), the solution is obtained as
Generalized Birth-Death Models

\[ P_0 = \left[ \sum_{k=0}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1} \]

where the first term in the summation is defined to be 1. Therefor being in any particular state \( k \) is:

**Equation 10.8.2**

\[ P_k = \left[ \sum_{k=0}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \quad \text{for } k = 0, 1, 2, \ldots \]

From this generalized steady-state solution, obtaining expressio

**Equation 10.8.3**

utilization = \( P_1 + P_2 + \cdots = 1 - P_0 \)

**Equation 10.8.4**

throughput = \( \mu_1 P_1 + \mu_2 P_2 + \cdots = \sum_{k=1}^{\infty} \mu_k P_k \)

**Equation 10.8.5**

queue length = \( 0P_0 + 1P_1 + 2P_2 + \cdots = \sum_{k=1}^{\infty} kP_k \)
• **Recurrent state**: A recurrent state in a Markov model is a state that can always be revisited in the future, regardless of the subsequent states visited. In the figure, states A, B, C, F, G, H, and I are recurrent states.

• **Transient state**: A transient state in a Markov model is a state where, depending on the subsequent states visited, it may be possible to return to the state after leaving it. States D and E are transient, since, say, after leaving state D, it is then not possible to ever return to states D or E.

• **Fact**: Each state in the Markov model is either recurrent or transient. The set of recurrent states and the set of transient states are mutually exclusive and collectively exhaustive.

• **Fact**: All states reachable from a recurrent state are recurrent.

• **Fact**: All states that can reach a transient state are transient.

• **Periodic state**: A periodic state is a recurrent state where the system can only return to the periodic state in p steps, where p is the period, as large as possible, and where \( p > 1 \). States A, B, and C are periodic with a period of around state H prohibits it (and also states F, G, and I) from being periodic. For instance, state F can return to state D in 8, 9, ..., steps. Since p must be greater than 1, F is not periodic since it can return to itself in either 4 or 5 steps.

• **Fact**: All states reachable from a periodic state are periodic with the same period.

• **Chain**: A chain is a set of recurrent states that can all reach each other. The set is as large as possible. States E and F form one chain. States F, G, H, and I form another chain. The diagram is, thus, a multi-chain Markov model.
2. Consider the following "system." Eight students are always in the weight room at the Rec Center. As soon as one student exits, another enters. Upon entering, a student goes to treadmill, then to the stationary bike, and then to the rowing machine. There is a single treadmill, stationary bike, and rowing machine. A student exercises for an average of 5 minutes on the treadmill, 8 minutes on the stationary bike, and 4 minutes on the rowing machine. A typical student makes two cycles through the three pieces of equipment before leaving. If a particular piece of equipment happens to be busy when a student wants to use it, he/she patiently waits until it becomes free.

- Use MVA to find the average number of students leaving the weight room per hour and also the average amount of time that each student stays in the weight room.

- Plot the average amount of time that each student stays in the weight room as a function of the number of students allowed in the weight room. Vary the number of students allowed in the weight room from 1 to 15.

- Suppose that it is desired to place a cap on the maximum utilization that any piece of equipment should be used (i.e., to allow for equipment maintenance time). If this cap is set at 80%, what is the maximum number of students that should be allowed in the room?

- Which piece of equipment is the system bottleneck? If a certain percentage of students were allowed to bypass this particular piece of equipment, what percentage of students should be allowed to bypass it so that it is no longer the system bottleneck?

- If an additional stationary bike were purchased and students were randomly assigned (i.e., equally likely) to use either bike, by how much would the average time be reduced so that a student spends in the weight room?
Exercise solution

Given the database example in §12.4 and balancing the system by upgrading the speed of the CPU and the slow disk.

Requested answer the following questions:

- By how much would the speed of the CPU and slow disk need to be improved to achieve a balanced system?
- How much would these upgrades improve overall system performance (i.e., throughput and response time)?

Solution To make utilization of CPU and slow disk equal to the utilization of the fast disk, we must decrease the service time to some amount, that it is for the fast disk. This way, utilization, which is ε, will be equal for all devices.

The service demand for the fast disk is equal to 7.5 seconds. The CPU has a service demand of 10 and the slow disk a service demand of 15. Therefore, the CPU must increase speed by \( \frac{33}{3}\% \), and demand 7.5\(^{\text{33}}\).

Analogously, the slow disk must increase speed by a factor 2 to get the service demand down from 15 to 7.5 milliseconds.

In order to find the overall improvement, the model must be solved with the old and the new values. Chapter 12 (page 324 — 326) contains the solved values. The spreadsheet ex. 12-3-mva.xls sob and the new model. The results from the spreadsheet are (almost) identical to the results in the book \(^{\text{54}}\).

The results are shown in figures \(^{90} — 93\) on pages \(^{8} — 8\).

As you can see, response time improves from 56.41 seconds to 37.50 or about 34% improvement. The throughput improves from 0.0531 tps to 0.0800 tps or about 51% improvement.
An Internet Service Provider (ISP) has \( m \) dial-up ports. Connection requests arrive at a rate of \( \lambda \) requests/sec. An average session duration is equal to \( 1/\mu \) seconds (i.e., each session completes at a rate of \( \mu \) req/sec).

- Draw a state transition diagram for the ISP showing the arrival and departure rates [\textit{Hint}: the state \( k \) represents the number of active connections].

- Give an expression for the probability \( p_k \) (\( k = 0, \ldots, m \)) that \( k \) connections are active as a function of \( \lambda \), \( \mu \), and \( m \).

- Find an expression for the probability \( p_{\text{busy}} \) that a customer finds all dial-up ports busy as a function of the state probabilities \( p_k \).