Limiting Behavior of DTMCs

It is sometimes useful to know the time-limiting behavior of a DTMC. This translates into the "long term," where the system has settled into some steady-state behavior.

Formally, we are looking for $\lim_{n\to\infty} \pi(n)$.

To compute this, what we want is $\lim_{n\to\infty} \pi(0)P^n$.

There are various ways to compute this. The simplest is to calculate $\pi(n)$ for increasingly large n, and when $\pi(n+1) \cong \pi(n)$, we can believe that $\pi(n)$ is a good approximation to steady-state. This can be rather inefficient if n needs to be large.

Classifications

It is much easier to solve for the steady-state behavior of some DTMC's than others. To determine if a DTMC is "easy" to solve, we need to introduce some definitions.

Definition: A state j is said to be *accessible* from state i if there exists an $n \geq 0$ such that $P_{ij}^{(n)} > 0$. We write $i \rightarrow j$.

Note: recall that $P_{ij}^{(n)} = P\left[X(n) = j \middle| X(0) = i\right]$

If one thinks of accessibility in terms of the graphical representation, a state j is accessible from state i if there exists a path of non-zero edges (arcs) from node i to node j.

State Classification in DTMCs

Definition: A DTMC is said to be *irreducible* if every state is accessible from every other state.

Formally, a DTMC is irreducible if

$$i \to j \qquad \text{for all } i,j \in S.$$

A DTMC is said to be *reducible* if it is not irreducible.

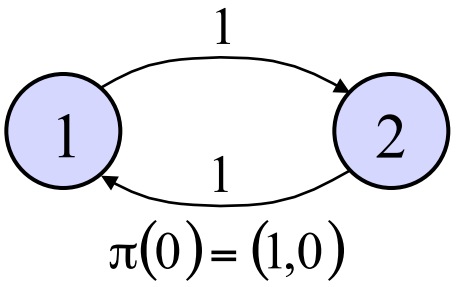
It turns out that irreducible DTMC's are simpler to solve. One need only solve one linear equation:

$$\pi = \pi P.$$

We will see why this is so, but first there is one more issue we must confront.

Periodicity

Consider the following DTMC:



$\pi(0) = (1,0)$

Does $\lim_{n\to\infty} \pi(n)$ exist? No!

However, $\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \pi(i)}{n}$ does exist; it is called the time-averaged steady-state distribution, and is denoted by π^*.

Definition: A state *i* is said to be *periodic with period* d if it can return to itself after *n* transitions **only** when *n* is some multiple of $d > 1$, *d* the smallest integer for which this is true. If $d = 1$, then *i* is said to be *aperiodic*.

Formally, $\exists d > 1, j \neq kd \Rightarrow P_{ii}^{(j)} = 0$

All states in the same strongly connected component have the same periodicity.

An irreducible DTMC is said to be periodic if all its states are periodic

A steady-state solution for an irreducible DTMC exists if all the states are aperiodic

A time-averaged steady-state solution for an irreducible DTMC always exists.

Transient and Recurrent States

- The probability of *first return* to state s after exactly n epochs is:

$$f_s(n) = \Pr\{\, X(n) = s, X(n-1) \neq s, \ldots, X(1) \neq s \mid X(0) = s \,\}$$

  - not to be confused with $p_{s,s}(n) = \mathbf{P}^n(s, s)$
  - relationship: $p_{s,s}(n) = \sum_{k=1}^{n} f_s(k) \cdot p_{s,s}(n-k) \quad \text{for } n \geq 1$

- Probability to *eventually return* to s: $f_s = \sum_{n=1}^{\infty} f_s(n)$
  - state s is called *transient* if $f_s < 1$

$\Rightarrow$ there is a non-zero probability that the DTMC will *not* return to s

  - state s is called *recurrent* if $f_s = 1$

$\Rightarrow$ it is impossible to never come back to a recurrent state

Mean Recurrence Time

- For recurrent state s, the mean number of epochs between two successive visits to s:

$$m_s = \sum_{n=1}^{\infty} n \cdot f_s(n)$$

  - state s is called *positive recurrent* if $m_s < \infty$
  - state s is called *null recurrent* if $m_s = \infty$

- State s is *ergodic* if s is aperiodic *and* positive recurrent
  - a DTMC is called ergodic when all its states are ergodic

- Fact: in a *finite*, aperiodic and irreducible DTMC all states are ergodic
  - there are also infinite DTMCs that are ergodic

ECE/CS 541: Computer System Analysis. ©2006 William H. Sanders. All rights reserved. Do not copy or distribute to others without the permission of the author.

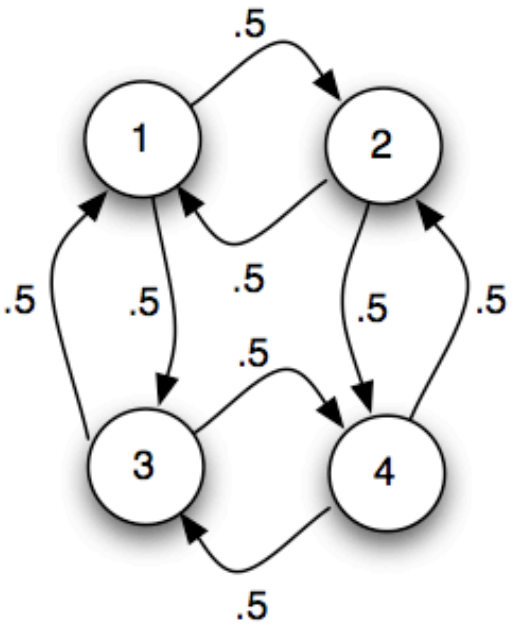
Connected States and Type

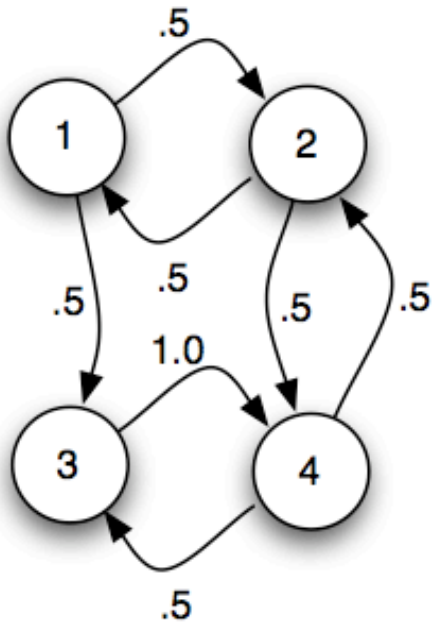
Let s and s' be mutually reachable from each other. Then:

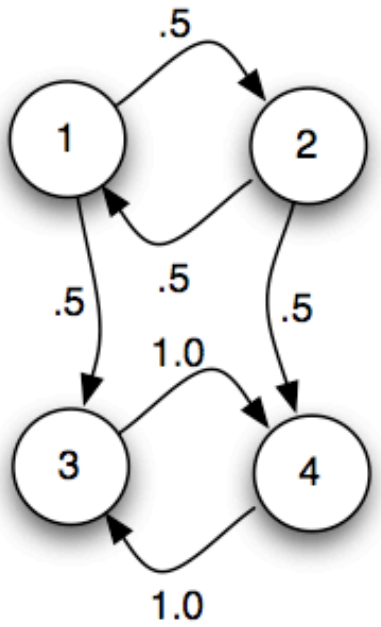
s is transient	iff	s' is transient
s is null-recurrent	iff	s' is null-recurrent
s is positive recurrent	iff	s' is positive recurrent
s has period d	iff	s' has period d

Examples on Board

Which of these are (a) irreducible? periodic?







Steady-State Solution of DTMCs

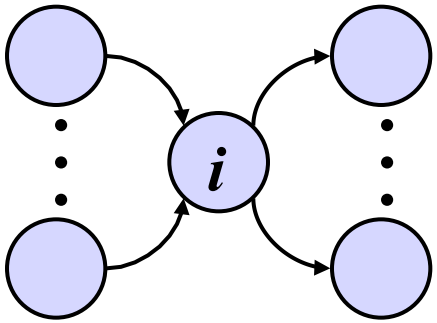
The steady-state behavior can be computed by solving the linear equation $\pi = \pi P$, with the constraint that $\sum_{i=1}^{n} \pi_i = 1$. For irreducible DTMC's, it can be shown that this solution is unique. If the DTMC is periodic, then this solution yields π^*.

One can understand the equation $\pi = \pi P$ in two different ways.

- In steady-state, the probability distribution $\pi(n + 1) = \pi(n)P$, and by definition $\pi(n + 1) = \pi(n)$ in steady-state.
- "Flow" equations.

Flow equations require some visualization. Imagine a DTMC graph, where the nodes are assigned the occupancy probability, or the probability that the DTMC has the value of the node.

Flow Equations



Probability must be conserved, i.e.,

$$\sum \pi_i = 1.$$

Let $\pi_i P_{ij}$ be the "probability mass" that moves from state j to state i in one time-step. Since probability must be conserved, the probability mass entering a state must equal the probability mass leaving a state.

Prob. mass in = Prob. mass out

$$\sum_{j=1}^{n} \pi_j P_{ji} = \sum_{j=1}^{n} \pi_i P_{ij}$$

$$= \pi_i \sum_{j=1}^{n} P_{ij}$$

$$= \pi_i$$

Written in matrix form, $\pi = \pi P$.

Solution of Steady State Equations

Start with $\pi = \pi P$

Transformations $0 = \pi P - \pi$ where I is the identity matrix

$$0 = \pi(P - I)$$

$$(P - I)^T \pi^T = 0^T$$

Where T denotes tranpose. π^T is a column vector

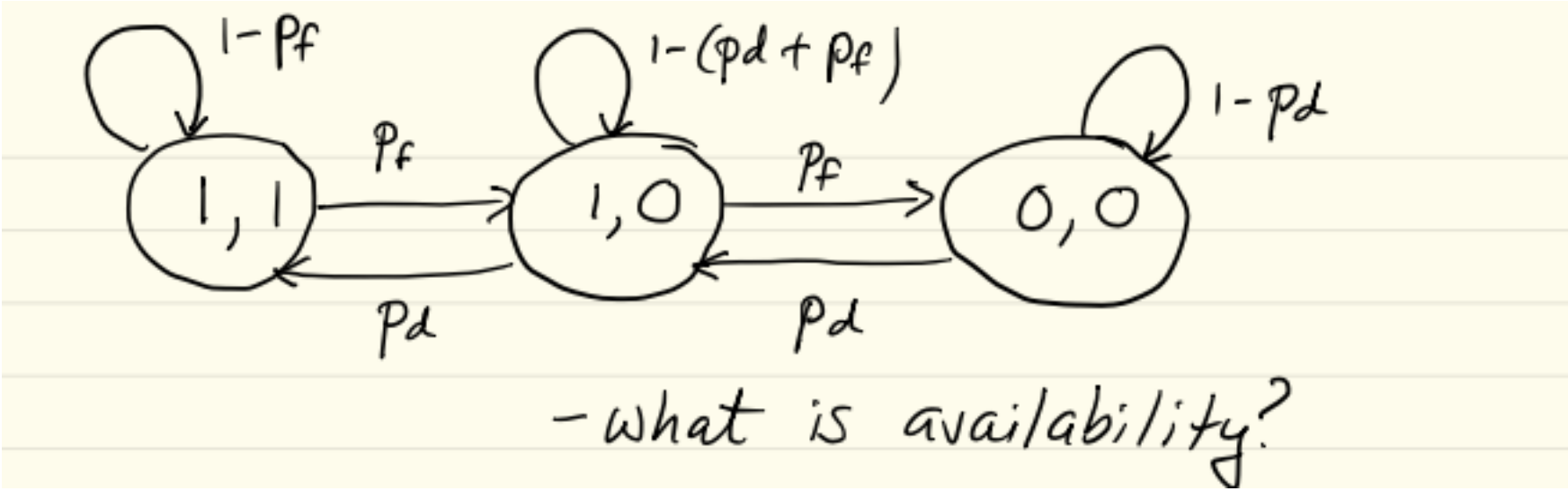
The matrix has rank n-1, which means the problem is under-specified. We add the constraint that the sum of the elements of the solution vector be 1

This can be done by adding a column of 1's to the matrix, and a corresponding 1 to the rhs

$$((P - I)^T [1\ 1\ 1 ... 1]^T) = [0\ 0\ 0\ ...0\ 1]^T$$

Solution using Maple

Example



pf	0.00027778		
pd	0.125		
	state 1	state 2	state 3
state 1	0.99972222	0.00027778	0
state 2	0.125	0.87472222	0.00027778
state 3	1	0.125	0

Solution using Maple

```
> with(LinearAlgebra)
```

(2)

[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, CompressedSparseForm, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm, FromSplitForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd,

Solution using Maple

Adding 1 to rows for probability constraint

pf	0.00027778		
pd	0.125		
	state 1	state 2	state 3
state 1	0.99972222	0.00027778	0
state 2	0.125	0.87472222	0.00027778
state 3	1	0.125	0

```
> r1 := [0.999722222, 0.000277778, 0, 1]
```

$$r1 := [0.999722222, 0.000277778, 0, 1]$$

```
> r2 := [0.125, 0.874722222, 0.000277778, 1]
```

$$r2 := [0.125, 0.874722222, 0.000277778, 1]$$

```
> r3 := [0, 0.125, 0.875, 1]
```

$$r3 := [0, 0.125, 0.875, 1]$$

```
>
> P := Matrix([r1, r2, r3])
```

$$P := \begin{bmatrix} 0.999722222 & 0.000277778 & 0 & 1 \\ 0.125 & 0.874722222 & 0.000277778 & 1 \\ 0 & 0.125 & 0.875 & 1 \end{bmatrix}$$

Solution using Maple

Create matrix to be used in solution

```
> Id := Matrix([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0]])
```

$$Id := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

```
> PId := P + (-1)Id
```

$$PId := \begin{bmatrix} -0.000277777999999951 & 0.000277778000000000 & 0. & 1. \\ 0.125000000000000 & -0.125277778000000 & 0.000277778000000000 & 1. \\ 0. & 0.125000000000000 & -0.125000000000000 & 1. \end{bmatrix}$$

```
> PIdT := Transpose(PId)
```

$$PIdT := \begin{bmatrix} -0.000277777999999951 & 0.125000000000000 & 0. \\ 0.000277778000000000 & -0.125277778000000 & 0.125000000000000 \\ 0. & 0.000277778000000000 & -0.125000000000000 \\ 1. & 1. & 1. \end{bmatrix}$$

Solution using Maple

Set up and solve Ax = b

```
> b := Vector([0, 0, 0, 1])
```

$$b := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

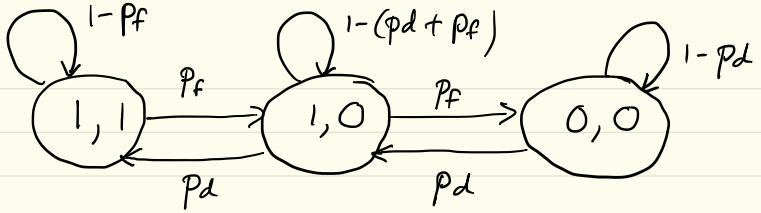
```
> soln := LinearSolve(PIdT, b)
```

$$soln := \begin{bmatrix} 0.997777786949577 \\ 0.0022172857448262 4 \\ 0.0000049273055970 1074 \end{bmatrix}$$



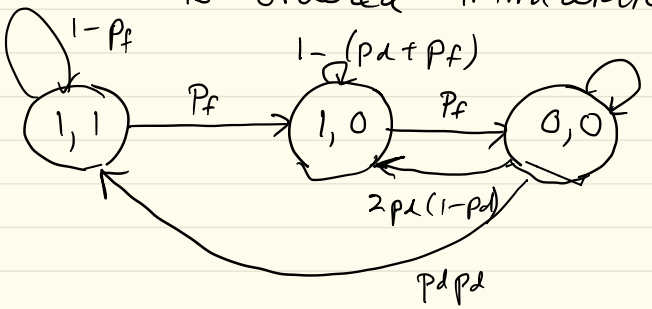
1 minus probability(in state 2) == availability

Compare replacement strategies



- what is availability?

Now suppose that when system fails another spare is ordered immediately



$1 - (p_d p_d + 2p_d(1-p_d))$

$= 1 - 2p_d + p_d p_d$

pf	0.00027778		
pd	0.125		
	state 1	state 2	state 3
state 1	0.99972222	0.00027778	0
state 2	0.125	0.87472222	0.00027778
state 3	0.015625	0.21875	0.765625

Compare replacement strategies

Just modify row 3 of transition matrix

Now solve model where replacement ordered immediately when system goes down

$r3p := [0.015625, 0.21875, 0.765625, 1]$

$$r3p := [0.015625, 0.21875, 0.765625, 1]$$

> $newP := Matrix([r1, r2, r3p])$

$$newP := \begin{bmatrix} 0.999722222 & 0.000277778 & 0 & 1 \\ 0.125 & 0.874722222 & 0.000277778 & 1 \\ 0.015625 & 0.21875 & 0.765625 & 1 \end{bmatrix}$$

> $newPId := newP + (-1)Id$

$$newPId := \begin{bmatrix} -0.000277777999999951 & 0.000277778000000000 & 0. & 1. \\ 0.125000000000000 & -0.125277778000000 & 0.000277778000000000 & 1. \\ 0.0156250000000000 & 0.218750000000000 & -0.234375000000000 & 1. \end{bmatrix}$$

> $newPIdT := Transpose(newPId)$

$$newPIdT := \begin{bmatrix} -0.000277777999999951 & 0.125000000000000 & 0.0156250000000000 \\ 0.000277778000000000 & -0.125277778000000 & 0.218750000000000 \\ 0. & 0.000277778000000000 & -0.234375000000000 \\ 1. & 1. & 1. \end{bmatrix}$$

> $soln2 := LinearSolve(newPIdT, b)$

Compare replacement strategies

Solve new system and compare probabilities of being in non-operational state

> $soln2 := LinearSolve(newPIdT, b)$

$$soln2 := \begin{bmatrix} 0.997780409352888 \\ 0.00221696313314837 \\ 0.0000026275139635 1867 \end{bmatrix}$$

ratio of probability being in failed state system 1 / probability being in failed state system 2

$$\frac{soln[3]}{soln2[3]};$$

$$1.87527284932571$$

>