Topics in Applied Probability

Collection Editor:
Paul E Pfeiffer

Topics in Applied Probability

Collection Editor:
Paul E Pfeiffer

Authors:
Paul E Pfeiffer
Daniel Williamson

Online:
< http://cnx.org/content/col10964/1.2/ >

C O N N E X I O N S

Rice University, Houston, Texas

Table of Contents

Preface[1]

Topics in Applied Probability is meant to supplement the collection Applied Probability[2] with examples, further problems, and m-files. This collection is not meant to stand alone but rather serve as a miscellany of further content not included in the original collection. Topics discussed in this collection include Martingale Sequences and Markov Procedures as well as a few others. In addition, the collection of m-files contains both the user-defined programs and a collection of m-files for specific problems with properly formatted data.

MATLAB Files

These files can be entered into MATLAB by calling the appropriate file. These m-files come from a variety of sources (e.g., exams or problem sets, hence the odd names) and may be useful for examples and exercises. In order to allow for easy access to the helpful m-files, a supplementary link has been placed in the upper right side of each module in the "Topics in Applied Probability" collection. Selecting this link will launch the download of a zip document containing a suite of m-files. Unzipping this file and saving its contents to your local system will allow for easy and reliable access when working with MATLAB.

[1]This content is available online at <http://cnx.org/content/m31871/1.2/>.
[2]*Applied Probability* <http://cnx.org/content/col10708/latest/>

Cost with Price Breaks[3]

Let D be the demand random variable. Suppose there is

A flat fee of C for $D \leq a_1$
A cost of c_1 per unit for $a_1 < D \leq a_2$
A cost of c_2 per unit for $a_2 < D \leq a_3$
A cost of c_3 per unit for $a_3 < D$

Then (see figure)

$$g(t) = C + c_1 I_{M_1}(t)(t - a_1) + (c_2 - c_1) I_{M_2}(t)(t - a_2) + (c_3 - c_2) I_{M_3}(t)(t - a_3) \quad \text{where } M_i = [a_i, \infty) \tag{1}$$

NOTE: Since $(t - a_i) = 0$ at $t = a_i$, we could as well have $M_i = (a_i, \infty)$.

Special Case. If D is Poisson (μ), we may obtain an exact solution for $E[g(D)]$ by using the fact that $E\left[I_{[m,\infty)}(D)\right] = P(D \geq m)$ and

$$E\left[I_{[m,\infty)}(D)D\right] = e^{-\mu} \sum_{k=m}^{\infty} k \frac{\mu^k}{k!} = \mu e^{-\mu} \sum_{k=m-1}^{\infty} \frac{\mu^k}{k!} = \mu P(D \geq m - 1) \tag{2}$$

As an alternate approach, we approximate D by an appropriate number of values.

Exercise 1: Example

A residential College is planning a camping trip over Spring Recess. The number D of persons planning to go is assumed to be Poisson (15). Arrangements for transportation and food have been made as follows:

If $D \leq 4$, a minimal flat fee of \$450 will be charged.
If $4 < D \leq 19$, the additional charge is \$100 for each person more than 4 (but less than 20).
If $19 < D$, the charge is \$80 for each person more than 19.

Thus, $C = 450, a_1 = 4, a_2 = 19, c_1 = 100$, and $c_2 = 80$. The distribution for D is Poisson (15). Then

$$Y = g(D) = 450 + 100 I_{[4,\infty)}(D)(D - 4) + (80 - 100) I_{[19,\infty)}(D)(D - 19) \tag{3}$$

$$= 450 + 100 I_{[4,\infty)}(D) D - 20 I_{[19,\infty)}(D) D - 400 I_{[4,\infty)}(D) + 380 I_{[19,\infty)}(D) \tag{4}$$

a. Obtain the "exact" solution to $E[g(D)]$.
b. Approximate D by terms up to 40 and obtain $E[g(D)]$ and $P(g(D) \geq v)$ for $v = 1000$, 1500, 2000, 2500.

[3]This content is available online at <http://cnx.org/content/m31064/1.4/>.

Order Statistics[4]

In sampling statistics (see Sec 18.7), we deal with an iid class $\{X_i : 1 \le i \le n\}$ of random variables, where n is a prescribed positive integer known as the sample size. An observation of this class gives an n-tuple of numbers $(t_1, t_2, \cdots, t_n)$. As an extension of the extreme values in the case of two variables (Ex 2.11), it is often useful to define a random variable Y_1 whose value for any ω is the smallest of the $X_i(\omega)$; a second random variable Y_2 whose value at ω is the next smallest of the $X_i(\omega)$, and so on through Y_n whose value at ω is the largest of the $X_i(\omega)$. We would like to be able to obtain the distributions for these new random variables in terms of the common distribution for the X_i. We formulate the problem as follows.

Example : Order Statistics
Suppose $\{X_i : 1 \le i \le n\}$ is iid, with common distribution function F. Let

- $Y_1 =$ smallest of $X_1, X_2, ..., X_n$
- $Y_2 =$ next larger of $X_1, X_2, ..., X_n$
- . . .
- $Y_n =$ largest of $X_1, X_2, ..., X_n$

Then Y_k is called the kth **order statistic** for the class $\{X_i : 1 \le i \le n\}$. We wish to determine the distribution functions $F_k(t) = P(Y_k \le t)\, 1 \le k \le n$. Now, $Y_k \le t$ iff k or more of the X_i have values no greater than t. We may view the process as a Bernoulli sequence of n trials. There is a success on the ith trial iff $X_i \le t$. The probability p of a success is $p = P(X \le t) = F(t)$. Hence

$$F_k(t) = P(Y_k \le t) = P(k \text{ or more of the } X_i \text{ lie in } (-inf, t]) = \sum_{j=k}^{n} C(n,j)\, F^j(t)\, [1 - F(t)]^{n-j} \quad (1)$$

Remark. Once the common distribution function F for the X_i is known, then the F_k are calculated in a straightforward manner. For that purpose we may use the MATLAB function cbinom.

Example
Suppose the X_i are exponential (2). Then $F_X(t) = 1 - e^{-2t}$ for positive t. Suppose $n = 5$. We calculate $F_k(t)$ for $t = 0.1, 0.3, 0.5, 0.7, 0.9$.

```
n = 5;
t = 0.1:0.2:0.9;
m = length(t);
F = 1 - exp(-2*t);
for i = 1:m
FK(i,:) = cbinom(n,F(i),1:n);
end
disp([t' F' FK])    % k = 1    k = 2     k = 3     k = 4     k = 5
0.1000    0.1813    0.6321    0.2249    0.0445    0.0046    0.0002
```

[4]This content is available online at <http://cnx.org/content/m31071/1.3/>.

```
0.3000    0.4512    0.9502    0.7456    0.4091    0.1324    0.0187
0.5000    0.6321    0.9933    0.9354    0.7364    0.3946    0.1009
0.7000    0.7534    0.9991    0.9852    0.9000    0.6400    0.2427
0.9000    0.8347    0.9999    0.9968    0.9653    0.8064    0.4052
```

The following special case is important in characterizing the Poisson process (see Sec 21.1).

Example

Order statistics for uniformly distributed random variables

Suppose $\{U_i:\ 1 \le i \le n\}$ is iid, uniform on $(0, T]$. Determine the distribution functions for the order statistics.

SOLUTION

The common distribution function for the U_i is given by $F(t) = t/T, \quad 0 \le t \le T$. According to the result in Ex 2.16, the kth order statistic Y_k has the distribution function

$$F_k(t) = P(Y_k \le t) = \sum_{j=k}^{n} C(n, j) \left(\frac{t}{T}\right)^j \left(\frac{T-t}{T}\right)^{n-j} \quad 0 < t < T \tag{2}$$

Chapter 1

Martingale Sequences

1.1 Martingale Sequences: The Concept, Examples, and Basic Patterns[1]

1.1.1 The concept, examples, and basic patterns

1.1.1.1 A classical example

The notion of martingales and related concepts seem to have originated in studies of games of chance similar to the following. Suppose

$Y_n =$ a gambler's "gain" on the nth play of a game
$Y_0 =$ the original capital or "bankroll"

Set $X_n = 0$ for $n < 0, X_n = \sum_{k=0}^{n} Y_k \forall n \geq 0$. Thus, X_n is the capital after n plays, and

$$Y_0 = X_0 \qquad Y_n = X_n - X_{n-1} \ \forall\, n \geq 0 \tag{1.1}$$

Put $U_n = (X_0,\, X_1,\, \cdots,\, X_n)$ and $V_n = (Y_0,\, Y_1,\, \cdots,\, Y_n)$. For any $n \in \mathbf{N}$, $U_n = g_n(V_n)$ and $V_n = h_n(U_n)$ or, equivalently, $\sigma(U_n) = \sigma(V_n)$. Hence $E[H|U_n] = E[H|V_n]$ *a.s.*

If Y_N is an independent class with $E[Y_n] = 0 \forall n \geq 1$, the game is considered *fair*. In this case, we have by (CE5), (CE7), and hypothesis

$$E[X_{n+1}|U_n] = E[Y_{n+1}|V_n] + E[X_n|U_n] = E[Y_{n+1}] + X_n = X_n \ \ a.s. \tag{1.2}$$

Also $E[X_{n+1} - X_n|U_n] = E[Y_{n+1}|V_n] = 0$ *a.s.*

Gamblers seek to develop a "system" to improve expected earnings. We examine some typical approaches and show their futility. To keep the analysis simple, consider a simple coin-flipping game. Let

$H_k =$ event of a "head" on the kth component trial
$T_k = H_k^c =$ event of a "tail" on the kth component trial

The player has a *system*. He decides how much to bet on each play from the pattern of previous events. Let $B_n[I_{H_n} - I_{T_n}] = B_n Z_n$ be the result of the nth play, where $|B_n|$ is the *amount* of the bet; $B_n > 0$ indicates a bet on a head; $B_n < 0$ indicates a bet on a tail; $B = 0$ indicates a decision *not* to bet.

Systems take various forms; here we consider two possibilities.

[1]This content is available online at <http://cnx.org/content/m31076/1.3/>.

1. The amount of the bet is determend by the pattern of outcomes of previous tosses

$$B_n = g_{n-1}\left(I_{H_1},\, I_{H_2},\, \cdots,\, I_{H_{n-1}}\right) \quad Y_n = B_n Z_n \quad Z_n = I_{H_n} - I_{T_n} = 2I_{H_n} - 1 \tag{1.3}$$

2. The amount bet is determined by the pattern of previous payoffs

$$B_n = g_{n-1}\left(Y_1,\, Y_2,\, \cdots,\, Y_{n-1}\right) = h_{n-1}\left(B_1,\, I_{H_1},\, \cdots,\, B_{n-1}I_{H_{n-1}}\right) \quad Y_n = B_n Z_n \tag{1.4}$$

Let $Y_0 = X_0 = C$, a constant. Since C is independent of any random variable, $E[H|C] = E[H]$. In either scheme, by (CE8), (CI5), and the fact $E[Z_k] = 0$

$$E[Y_{n+1}|V_n] = E[B_{n+1}Z_{n+1}|\phi_n(B_1,\, I_{H_1},\, \cdots,\, B_n I_{H_n})] = B_{n+1}E[Z_{n+1}] = 0 \ a.s. \tag{1.5}$$

It follows that

$$E[X_{n+1}|U_n] = E[Y_{n=1}|v_n] + E[X_n|U_n] = 0 + X_n \ a.s. \tag{1.6}$$

The "fairness" of the game is not altered by the betting scheme, since decisions must be based on past performance. In spite of simple beginnings, the extension and analysis of these patterns form a major thrust of modern probability theory.

1.1.2 Formulation of the concept

In the following treatment,

$X_{\mathbf{N}} = \{X_n :\ n \in \mathbf{N}\}$ is the **basic sequence** $\quad \mathbf{N} = \{0,\, 1,\, \cdots\}$
$Y_{\mathbf{N}} = \{Y_n :\ n \in \mathbf{N}\}$ is the **incremental sequence**

$$Y_n = X_n - X_{n-1} \qquad X_n = \sum_{k=0}^{n} Y_k \ \ n \geq 0, \quad [X_n = 0 \ \ n < 0] \tag{1.7}$$

We suppose Z_N is a **decision sequence** and $X_{\mathbf{N}} \sim Z_{\mathbf{N}}$; that is, $X_n = g_n(W_n) = g_n(Z_0,\, Z_1,\, \cdots,\, Z_n)$.

- $X_{\mathbf{N}} \sim Z_{\mathbf{N}}$ iff $Y_{\mathbf{N}} \sim Z_{\mathbf{N}}$
- If $X_{\mathbf{N}} \sim H_{\mathbf{N}}$ and $H_{\mathbf{N}} \sim Z_{\mathbf{N}}$, then $X_{\mathbf{N}} \sim Z_{\mathbf{N}}$. In particular, if $H_n = k_n(U_n) = k_n(X_0,\, X_1,\, \cdots,\, X_n)$, then $X_{\mathbf{N}} \sim H_{\mathbf{N}}$.

Definition. If X_N is integrable and Z_N is a decision sequence, then

1. X_N is a **martingale** (MG) relative to Z_N iff

$$X_{\mathbf{N}} \sim Z_{\mathbf{N}} \quad \text{and} \quad E[X_{n+1}|W_n] = X_n \ a.s. \quad \forall\, n \in \mathbf{N} \tag{1.8}$$

2. X_N is a **submartingale** (SMG) relative to Z_N iff

$$X_{\mathbf{N}} \sim Z_{\mathbf{N}} \quad \text{and} \quad E[X_{n+1}|W_n] \geq X_n \ a.s. \quad \forall\, n \in \mathbf{N} \tag{1.9}$$

3. X_N is a **supermartingale** (SRMG) relative to Z_N iff

$$X_{\mathbf{N}} \sim Z_{\mathbf{N}} \quad \text{and} \quad E[X_{n+1}|W_n] \leq X_n \ a.s. \quad \forall\, n \in \mathbf{N} \tag{1.10}$$

Notation. When we write $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a martingale (submartingale, supermartingale), we are asserting X_N is integrable, Z_N is a decision sequence, $X_{\mathbf{N}} \sim Z_{\mathbf{N}}$, and X_N is a MG (SMG, SRMG) relative to Z_N.

Definition. If Y_N is integrable and Z_N is a decision sequence, then

1. Y_N is **absolutely fair** relative to Z_N iff
$$Y_{\mathbf{N}} \sim Z_{\mathbf{N}} \quad \text{and} \quad E\left[Y_{n+1}|W_n\right] = 0 \; a.s. \quad \forall\, n \in \mathbf{N} \tag{1.11}$$
2. Y_N is **favorable** relative to Z_N iff
$$Y_{\mathbf{N}} \sim Z_{\mathbf{N}} \quad \text{and} \quad E\left[Y_{n+1}|W_n\right] \geq 0 \; a.s. \quad \forall\, n \in \mathbf{N} \tag{1.12}$$
3. Y_N is **unfavorable** relative to Z_N iff
$$Y_{\mathbf{N}} \sim Z_{\mathbf{N}} \quad \text{and} \quad E\left[Y_{n+1}|W_n\right] \leq 0 \; a.s. \quad \forall\, n \in \mathbf{N} \tag{1.13}$$

Notation. When we write $(Y_{\mathbf{N}}, Z_{\mathbf{N}})$ is absolutely fair (favorable, unfavorable), we are asserting Y_N is integrable, Z_N is a decision sequence, $Y_{\mathbf{N}} \sim Z_{\mathbf{N}}$, and Y_N is absolutely fair (favorable, unfavorable) relative to Z_N. IXA2-2

Theorem 1.1: IXA2-1
If X_N is a basic sequence and Y_N is the corresponding incremental sequence, then

1. $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a martingale iff $(Y_{\mathbf{N}}, Z_{\mathbf{N}})$ is absolutely fair.
2. $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a submartingale iff $(Y_{\mathbf{N}}, Z_{\mathbf{N}})$ is favorable.
3. $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a supermartingale iff $(Y_{\mathbf{N}}, Z_{\mathbf{N}})$ is unfavorable.

Proof:
Let * be any one of the symbols $=$, $\geq$, or $\leq$. Then by linearity and (CE7)

$$E\left[X_{n+1}|W_n\right] = E\left[Y_{n+1}|W_n\right] + E\left[X_n|W_n\right] = E\left[Y_{n+1}|W_n\right] + X_n \; * \; X_n \; a.s. \quad \text{iff} \quad E\left[Y_{n+1}|W_n\right] \; * \; 0 \; a.s. \tag{1.14}$$

Remarks

1. $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a SMG iff $(-X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a SRMG
2. We write (S)MG to indicate the same statement can be made for a MG or a SMG with the appropriate choice of $=$ or $\geq$
3. We write $(\geq)$ to indicate simultaneously two cases:
   - $(\geq)$ read as $=$ in all places (for a MG)
   - $(\geq)$ read as $\geq$ in all places (for a SMG)

1.1.3 Some Basic Patterns

Theorem 1.2: IXA3-1
If $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a (S)MG and $X_{\mathbf{N}} \sim H_{\mathbf{N}}$, with $H_{\mathbf{N}} \sim Z_{\mathbf{N}}$, then $(X_{\mathbf{N}}, H_{\mathbf{N}})$ is a (S)MG.
Proof:
Let $K_n = (H_0, H_1, \cdots, H_n)$. By (CE9), the (S)MG definition, monotonicity, and (CE7)

$$E\left[X_{n+1}|K_n\right] = E\{E\left[X_{n+1}|W_n\right]|K_n\} \; (\geq) \; E\left[X_n|K_n\right] = X_n \; a.s. \tag{1.15}$$

Theorem 1.3: IXA3-2
For integrable $X_{\mathbf{N}} \sim Z_{\mathbf{N}}$, the following are *equivalent*

a	$(X_{\mathbf{N}},\, Z_{\mathbf{N}})$	is a (S)MG
b	$E\left[X_{n+k} \vert W_n\right] \;(\geq)\; X_n \;\; a.s. \quad \forall$	$n,\, k \in \mathbf{N}$
c	$E\left[I_C X_{n+1}\right] \;(\geq)\; E\left[I_C X_n\right] \quad \forall$	$C \in \sigma\left(W_n\right)$
$\forall$	$n \in \mathbf{N}$	
d	$E\left[I_C X_{n+k}\right] \;(\geq)\; E\left[I_C X_n\right] \quad \forall$	$C \in \sigma\left(W_n\right)$
$\forall$	$n,\, k \in \mathbf{N}$	

Proof:

b $\Rightarrow$ **a:** as a special case
a $\Rightarrow$ **b:** By (CE9), (a), and monotonicity

$$E\left[X_{n+k} \vert W_n\right] = E\{E\left[X_{n+k} \vert W_{n+k-1}\right] \vert W_n\} \;\; (\geq) \;\; E\left[X_{n+k-1} \vert W_n\right] \;\; a.s. \tag{1.16}$$

$k-1$ iterations yield $E\left[X_{n+k} \vert W_n\right] \;\; (\geq) \;\; X_n \;\; a.s.$
d $\Rightarrow$ **c:** as a special case
c $\Rightarrow$ **a:** By (CE1) and (c), $E\left[I_C X_{n+1}\right] = E\{I_C E\left[X_{n+1} \vert W_n\right]\} \;(\geq)\; E\left[I_C X_n\right] \;\; a.s.$. Since $X_n \sim W_n \;\; a.s.$ and $E\left[X_{n+1} \vert W_n\right] \sim W_n \;\; a.s.$, the result follows from the uniqueness property (E5)
b $\Rightarrow$ **d:** By (CE1), (b), and monotonicity $E\left[I_C X_{n+k}\right] = E\{I_C E\left[X_{n+k} \vert W_n\right]\} \;\; (\geq) \;\; E\left[I_C X_n\right]$

We thus have $d \Rightarrow c \Rightarrow a \Leftrightarrow b \Rightarrow d$

Corollary 1.1: IXA3-3
If $(X_{\mathbf{N}},\, Z_{\mathbf{N}})$ is a (S)MG, then $E\left[X_{n+k}\right] \;\; (\geq) \;\; E\left[X_n\right] \;\; (\geq) \;\; E\left[X_0\right]$

Theorem 1.4: IXA3-4
$(X_{\mathbf{N}},\, Z_{\mathbf{N}})$ is a (S)MG iff $E\left[X_q - X_p \vert W_n\right] \;\; (\geq) \;\; 0 \;\; a.s. \quad \forall\, n \leq p < q \in \mathbf{N}$
Proof:
EXERCISE. Note $X_q - X_p = Y_{p+1} + \;\cdots\; + Y_q$

IXA3-2

Theorem 1.5: IXA3-5
If $(X_{\mathbf{N}},\, Z_{\mathbf{N}})$ is an $\mathbf{L}^2$ MG, then

$$\begin{aligned}
E\left[X_q - X_p\right] &= 0 & \forall \quad & p < q \in \mathbf{N} \\
E\left[X_n\left(X_q - X_p\right)\right] &= 0 & \forall \quad & n \leq p < q \in \mathbf{N} \\
E\left[\left(X_n - X_m\right)\left(X_q - X_p\right)\right] &= 0 & \forall \quad & m < n \leq p < q \in \mathbf{N} \\
E\left[X_p X_q\right] &= E\left[X_{p \wedge q}^2\right] & \forall \quad & p,\, q \in \mathbf{N} \\
E\left[\left(X_q - X_p\right)^2\right] &= E\left[X_q^2\right] - E\left[X_p^2\right] \geq 0 & \forall \quad & p < q \in \mathbf{N} \\
E\left[X_p^2\right] &= \textstyle\sum_{k=0}^{p} E\left[Y_k^2\right] & \forall \quad & p \in \mathbf{N}
\end{aligned}$$

Proof:

a. $E\left[X_q - X_p\right] = E\{E\left[X_q - X_p \vert W_n\right]\} = 0$ by (CE1b) and Thm IXA3-4
b. $E\left[X_n\left(X_q - X_p\right)\right] = E\{X_n E\left[X_q - X_p \vert W_n\right]\} = 0$ by (CE1b), (CE8), and Thm IXA3-4
c. As in b, since $X_n - X_m \sim W_n$

d. Suppose $p < q$. Then, since $X_p \sim W_p$, $E[X_p X_q] = E\{X_p E[X_q|W_p]\} = E[X_p^2]$ by definition of MG. For $q < p$, interchange p, q in the argument above.
e. $E\left[(X_q - X_p)^2\right] = E[X_q^2] - 2E[X_p X_q] + E[X_p^2] = E[X_q^2] - 2E[X_p^2] + E[X_p^2]$ by d, above
f. By c, $E[Y_j Y_k] = 0$ for $j \neq k$. Hence, $E[X_p^2] = E\left[(\sum_{k=0}^p Y_k)^2\right] = \sum_j \sum_k E[Y_j Y_k] = \sum_{k=0}^p E[Y_k^2]$

A variety of weighted sums of increments are useful.

Theorem 1.6: IXA3-6
Suppose ($X_{\mathbf{N}}$, $Z_{\mathbf{N}}$) is a (S)MG and Y_N is the incremental sequence. Let H_0 be a (nonnegative) constant and let $H_n \sim W_{n-1}$, $n \geq 1$, be bounded (nonnegative). Set

$$X_n^* = \sum_{k=0}^{n} H_k Y_k = \sum_{k=0}^{n} Y_k^* \quad \forall\, n \in \mathbf{N} \tag{1.17}$$

Then ($X_{\mathbf{N}}^*$, $Z_{\mathbf{N}}$) is a (S)MG.
Proof:
$E[Y_{n+1}^*|W_n] = E[H_{n+1} Y_{n+1}|W_n] = H_{n+1} E[Y_{n+1}|W_n]$ $a.s.$ by (CE8)
For MG case: $E[Y_{n+1}^*|W_n] = 0$ $a.s.$ for arbitrary bounded H_n
For SMG case: $E[Y_{n+1}^*|W_n] \geq 0$ $a.s.$ for $H_n \geq 0$, bounded
The conclusion follows from Theorem 1.1, IXA2-1, p. 9

Remark. This result extends the pattern in the introductory gambling example. Theorem 1.4, IXA3-4, p. 10 IXA3-3

Theorem 1.7: IXA3-7
In Theorem IXA3-6 (Theorem 1.6, IXA3-6, p. 11), if $E[X_0] \geq 0$ and $0 \leq H_n \leq 1 a.s. \forall n \in \mathbf{N}$, then $0 \leq E[X_n^*] \leq E[X_n] \forall n \in \mathbf{N}$
Proof:
$E[Y_{n+1}|W_n] \geq H_{n+1} E[Y_{n+1}|W_n]$ $(\geq)$ 0 $a.s.$, by hypothesis, and $H_{n+1} E[Y_{n+1}|W_n] = E[Y_{n+1}^*|W_n]$ $a.s.$, by (CE8). Thus, by monotonicity and (CE1b)

$$E[Y_{n+1}] \ (\geq) \ E[Y_{n+1}^*] \ (\geq) \ 0 \quad \forall\, n \in \mathbf{N} \quad \text{and} \quad E[Y_0] = E[X_0] \geq H_0 E[Y_0] = E[Y_0^*] \tag{1.18}$$

Hence

$$E[X_n] = \sum_{k=0}^{n} E[Y_k] \geq \sum_{k=0}^{n} E[Y_k^*] = E[X_n^*] \geq 0 \tag{1.19}$$

Some important special cases

Theorem 1.8: IXA3-8
Suppose integrable $X_{\mathbf{N}} \sim Z_{\mathbf{N}}$. If $X_{n+1} - X_n \, (\geq) \, 0 a.s. \forall n \in \mathbf{N}$, then $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a (S)MG.
Proof:
Apply monotonicity and Theorem IXA3-4 (Theorem 1.4, IXA3-4, p. 10)

Theorem 1.9: IXA3-9
Suppose X_N has independent increments.

a. If $E[X_n] = c$, invariant with n, then X_N is a MG.
b. If $E[X_{n+1} - X_n] \, (\geq) \, 0, \forall n \in \mathbf{N}$, then ($X_{\mathbf{N}}$ is a (S)MG.

Proof:

b. For any n, consider any $C \in \sigma(U_n)$. By independent increments, $\{I_C, (X_{n+1} - X_n)\}$ is independent. Hence, $E[I_C X_{n+1}] - E[I_C X_n] = E[I_C(X_{n+1} - X_n)] = E[I_C] E[(X_{n+1} - X_n)] (\geq) 0$. The desired result follows from Theorem IXA3-2(c) (Theorem 1.3, IXA3-2, p. 10).

Theorem 1.10: IXA3-10
Suppose g is a *convex* Borel function on an interval I which contains the range of all X_n and $E[|g(X_n)|] < \infty \forall n \in \mathbf{N}$, Let $H_n = g(X_n) \forall n \in \mathbf{N}$,

a. If $(X_\mathbf{N}, Z_\mathbf{N})$ is a MG, then $(H_\mathbf{N}, Z_\mathbf{N})$ is a SMG.
b. If g is nondecreasing and $(X_\mathbf{N}, Z_\mathbf{N})$ is a SMG, then so is $(H_\mathbf{N}, Z_\mathbf{N})$

Proof:

a. : By Jensen's inequality and the definition of a MG

$$E[g(X_{n+1})|W_n] \geq g(E[X_{n+1}|W_n]) = g(X_n) \ \ a.s. \tag{1.20}$$

b. : By Jensen's inequality

$$E[g(X_{n+1})|W_n] \geq g(E[X_{n+1}|W_n]) \ \ a.s. \tag{1.21}$$

Since $E[X_{n+1}|W_n] \geq X_n$ *a.s.* and g is nondecreasing, we have

$$g(E[X_{n+1}|W_n]) \geq g(X_n) \ \ a.s. \tag{1.22}$$

Some commonly utilized convex functions

1. $g(t) = |t|$
2. $g(t) = t^2 g$ is increasing for $t \geq 0$
3. $g(t) = u(t)t \quad g(X_n) = X_n^+ g$ nondecreasing for all t
4. $g(t) = \ - u(-t)tg(X_n) = X_n^- g$ nonincreasing for all t
5. $g(t) = e^{at}, \ \ a > 0 g$ is increasing for all t

Theorem 1.11: IXA3-11
Consider integrable $X_\mathbf{N} \sim Z_\mathbf{N}$.

a. If $E[X_{n+1}|W_n] = aX_n a.s. \forall n$ and $X_n^* = \frac{1}{a^n} X_n \forall n$, then $(X_\mathbf{N}^*, Z_\mathbf{N})$ is a MG
b. If $E[X_{n+1}|W_n] \geq aX_n a.s., a > 0, \forall n$ and $X_n^* = \frac{1}{a^n} X_n \forall n$, then $(X_\mathbf{N}^*, Z_\mathbf{N})$ is a SMG

Proof:

$$E\left[X_{n+1}^*|W_n\right] = \frac{1}{a^{n+1}} E[X_{n+1}|W_n] \ \ (\geq) \ \frac{1}{a^{n+1}} aX_n = X_n^* \ \ a.s. \tag{1.23}$$

The restrictionl $a > 0$ is needed in the $\geq$ case.

1.2 Martingale Sequences: Examples and Further Patterns[2]

1.2.1 Examples and further patterns

Theorem 1.12: A4-1 Sums of Independent Random Variables
Suppose Y_N is an independent, integrable sequence. Set $X_n = \sum_{k=0}^{n} Y_k \quad \forall\, n \geq 0$.
If $E\left[Y_n\right]\;(\geq)\;0\;\;\forall\, n \geq 1$, then X_N is a (S)MG.

Theorem 1.13: A4-2 Products of nonnegative random variables
Suppose $Y_{\mathbf{N}} \sim Z_{\mathbf{N}}, Y_n \geq 0 a.s. \forall n$. Consider $X_{\mathbf{N}} : X_n = c\prod_{k=0}^{n} Y_k, c > 0$.
If $E\left[Y_{n+1}|W_n\right]\;(\geq)\;1\;a.s.\;\forall\, n$, then $(X_{\mathbf{N}},\, Z_{\mathbf{N}})$ is a (S)MG
Proof:
$X_n \sim W_n$ and $X_{n+1} = Y_{n+1}X_n$. Hence, $E\left[X_{n+1}|W_n\right] = X_n\;E\left[Z_{n+1}|W_n\right]\;(\geq)\;X_n\;a.s.\;\forall\, n$

Theorem 1.14: A4-3 Discrete random walk
Consider $Y_0 = 0$ and $\{Y_n : 1 \leq n\}$ iid. Set $X_n = \sum_{k=0}^{n} Y_k \forall n \geq 0$. Suppose $P\left(Y_n = k\right) = p_k$. Let

$$g_Y\left(s\right) = E\left[s^{Y_n}\right] = \sum_k p_k s^k, \quad s > 0 \tag{1.24}$$

Now $g_Y\left(1\right) = 1,\;\; g_Y^{'}\left(1\right) = E\left[Y_n\right],\;\; g_Y^{''}\left(s\right) = \sum_k k\left(k-1\right)p_k s^{k-2} > 0$ for $s > 0$. Hence, $g_Y\left(s\right) = 1$ has at most two roots, one of which is $s = 1$.

a. $s = 1$ is a minimum point iff $E\left[Y_n\right] = 0$, in which case X_N is a MG (see A4-1 (Theorem 1.12, A4-1 Sums of Independent Random Variables, p. 13))
b. If $g_Y\left(r\right) = 1$ for $0 < r < 1$, then $E\left[r^{Y_n}\right] = 1 \forall n \geq 1$. Let $Z_0 = 1, Z_n = r^{X_n} = \prod_{k=1}^{n} r^{Y_k}$. By A4-2, Z_N is a MG

For the MG case in Theorem IXA3-6, the Y_n are centered at conditional expectation; that is $E\left[Y_{n+1}|W_n\right] = 0\;a.s.$ The following is an extension of that pattern.

Theorem 1.15: A4-4 More general sums
Consider integrable $Y_{\mathbf{N}} \sim Z_{\mathbf{N}}$ and bounded $H_{\mathbf{N}} \sim Z_{\mathbf{N}}$. Let $W_n =$ a constant for $n < 0$ and $H_n = 1$ for $n < 0$. Set

$$X_n = \sum_{k=0}^{n} \{Y_k - E\left[Y_k|W_{k-1}\right]\}H_{k-1} \;\;\forall\, n \geq 0 \tag{1.25}$$

Then $(X_{\mathbf{N}},\, Z_{\mathbf{N}})$ is a MG.
Proof:
$X_n \sim W_n; \forall n \geq 0$ and $E\left[X_{n+1}|W_n\right] = X_n + H_n E\{Y_{n+1} - E\left[Y_{n+1}|W_n\right]|W_n\} = X_n + 0 a.s.$

IXA4-2

Theorem 1.16: A4-5 Sums of products
Suppose Y_N is absolutely fair relative to Z_N, with $E\left[|Y_n|^k\right] < \infty \forall n$, fixed $k > 0$. For $n \geq k$, set

$$X_n = \sum_{0 \leq i_1 < \cdots \leq n} Y_{i_1} Y_{i_2} \cdots Y_{i_k} \;\;\sim \mathbf{G}_n \tag{1.26}$$

Then $(X_{\mathbf{N}_k},\, Z_{\mathbf{N}_k})\;\;\mathbf{N}_k = \{k,\, k+1,\, k+2,\;\cdots\}$ is a MG,

[2]This content is available online at <http://cnx.org/content/m31067/1.3/>.

Proof:
$X_{n+1} = X_n + K_{n+1}$, where

$$K_{n+1} = Y_{n+1} \sum_{0 \leq i_1 < \cdots \leq n} Y_{i_1} Y_{i_2} \cdots Y_{i_{k-1}} = Y_{n+1} K_n^* \quad K_n^* \sim W_n \tag{1.27}$$

$$E\left[K_{n+1}|W_n\right] = K_n^* E\left[Y_{n+1}|W_n\right] = 0 \ a.s. \quad \forall\, n \geq k \tag{1.28}$$

We consider, next, some relationships with **homogeneous Markov sequences**.

Suppose $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a homogeneous Markov sequence with finite state space $E = \{1, 2, \cdots, M\}$ and transition matrix $P = [p(i, j)]$. A function f on E is represented by a *column matrix*$f = [f(1), f(2), \cdots, f(M)]^T$. Then $f(X_n)$ has value $f(k)$ when $X_n = k$. Pf is an $m \times 1$ column matrix and $Pf(j)$ is the jth element of that matrix. Consider $E[f(X_{n+1})|W_n] = E[f(X_{n+1})|X_n]$ $a.s.$. Now

$$E\left[f(X_{n+1})|X_n = j\right] = \sum_{k \in E} f(k)\, p(j, k) = Pf(j) \quad \text{so that} \quad E\left[f(X_{n+1})|W_n\right] = Pf(X_n) \tag{1.29}$$

A nonnegative function f on E is called **(super)harmonic** for P iff $Pf \ (\leq)\ f$.

Theorem 1.17: A4-6 Positive supermartingales and superharmonic functions.
Suppose $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a homogeneous Markov sequence with finite state space $E = \{1, 2, \cdots, M\}$ and transition matrix $P = [p(i, j)]$. For nonnegative f on E, let $Y_n = f(X_n) \ \forall\, n \in \mathbf{N}$. Then $(Y_{\mathbf{N}}, Z_{\mathbf{N}})$ is a positive (super)martingale P(SR)MG iff f is (super)harmonic for P.
Proof:
As noted above $E[f(X_{n+1})|W_n] = Pf(X_n)$.

1. If f is (super)harmonic $Pf(X_n) \ (\leq)\ f(X_n) = Y_n$, so that

$$E\left[Y_{n+1}|W_n\right] \ (\leq)\ Y_n \ a.s. \tag{1.30}$$

2. If $(Y_{\mathbf{N}}, Z_{\mathbf{N}})$ is a P(SR)MG, then

$$Y_n = f(X_n) \ (\geq)\ E\left[f(X_{n+1})|W_n\right] = Pf(X_n) \ a.s., \ \text{so that } f \text{ is (super)harmonic} \tag{1.31}$$

IX A4-3
An **eigenfunction** f and associated **eigenvalue** λ for P satisfy $Pf = \lambda f$ (i.e., $(\lambda I - P)f = 0$). In most cases, $|\lambda| < 1$. For real λ, $0 < \lambda < 1$, the eigenfunctions are superharmonic functions. We may use the construction of Theorem IXA3-12 to obtain the associated MG.

Theorem 1.18: A4-7 Martingales induced by eigenfunctions for homogeneous Markov sequences
Let $(Y_{\mathbf{N}}, Z_{\mathbf{N}})$ be a homogenous Markov sequence, and f be an eigenfunction with eigenvalue λ. Put $X_n = \lambda^{-n} f(Y_n)$. Then, by Theorem IAXA3-12, $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a MG.

Theorem 1.19: A4-8 A dynamic programming example.
We consider a *horizon of N stages* and a finite state space $\mathbf{E} = \{1, 2, \cdots, M\}$.

- *Observe* the system at prescribed instants
- Take *action* on the basis of previous states and actions.

Suppose the *observed state* is j and the *action* is $a \in A$. Two results ensue:

1. A return $r(j, a)$ is realized

2. The system moves to a new state

Let:

Y_n = state in nth period, $0 \le n \le N - 1$

A_n = action taken on the basis of $Y_0, A_0, \cdots, Y_{n-1}, A_{n-1}, Y_n$

[A_0 is the initial action based on the initial state Y_0]

A *policy*π is a set of functions $(\pi_0, \pi_1, \cdots, \pi_{N-1})$, such that

$$A_n = \pi_n (Y_0, A_0, \cdots, Y_{n-1}, A_{n-1}, Y_n) \quad 0 \le n \le N-1 \tag{1.32}$$

The *expected return* under policy π, when $Y_0 = j_0$ is

$$R(\pi, j_0) = E\left[\sum_{k=0}^{N-1} r(Y_k, A_k)\right] \tag{1.33}$$

The *goal* is to determine π to maximize $R(\pi, j_0)$.

Let $Z_k = (Y_k, A_k)$ and $W_n = (Z_0, Z_1, \cdots, Z_n)$. If $\{Y_k : 0 \le k \le N-1\}$ is Markov, then use of (CI9) and (CI11) shows that for any policy the Z-process is Markov. Hence

$$E[I_M(Y_{n+1}) \mid W_n] = E[I_M(Y_{n+1}) \mid Z_n] \ \ a.s. \quad \forall\, n:\ 0 \le n \le N-1, \ \ \forall \text{ Borel sets } M \tag{1.34}$$

We assume *time homogeneity* in the sense that

$$P(Y_{n+1} = j | Y_n = i, A_n = a) = p(j|i, a), \ \text{ invariant with } n, \ \forall\, i, j \in \mathbf{E}, \ \forall\, a \in A \tag{1.35}$$

We make a dynamic programming approach

Define recursively $f_N, f_{N-1}, \cdots, f_0$ as follows:

$f_N(j) = 0, \forall j \in \mathbf{E}$. For $n = N, N-1, \cdots, 2, 1$, set

$$f_{n-1}(j) = max\{r(j, a) + \sum_{k \in \mathbf{E}} f_n\ (k)\, p(k|j, a) :\ a \in A\} \tag{1.36}$$

Put

$$X_n = \sum_{k=1}^{n} \{f_k(Y_k) - E[f_k(Y_k) | W_{k-1}]\} \tag{1.37}$$

Then, by A4-2 (Theorem 1.13, A4-2 Products of nonnegative random variables, p. 13), $(X_\mathbf{N}, Z_\mathbf{N})$ is a MG, with $E[X_n] = 0, \ 0 \le n \le N$ and

$$f_{n-1}(Y_{n-1}) \ge r(Y_{n-1}\ , A_{n-1}) + \sum_{k \in \mathbf{E}} f_n(k)\, p(k|Z_{n-1}) = r(Y_{n-1}\ , A_{n-1}) + E[f_n(Y_n) | W_{n-1}] \tag{1.38}$$

IX A4-4

We may therefore assert

$$0 = E[X_N] = E\left(\sum_{k=1}^{N}\{f_k\ (Y_k) - E[f_k(Y_k) | W_{k-1}]\}\right) \ge E\left(\sum_{k=1}^{N}\{f_k\ (Y_k) + r(Y_{k-1}, A_{k-1}) - f_{k-1}\ (Y_{k-1})\}\right) \tag{1.39}$$

$$= E\left[\sum_{k=0}^{N-1} r(Y_k, A_k\ ) + f_N(Y_N) - f_0(Y_0)\right] = E\left[\sum_{k=0}^{N-1} r(Y_k, A_k\ )\right] - E[f_0\ (Y_0)] \tag{1.40}$$

Hence, $R(\pi, Y_0) \leq E[f_0(Y_0)]$. For $Y_0 = j_0$, $R(\pi, j_0) \leq f_0(j_0)$. If a policy π^* can be found which yields equality, then π^* is an optimal policy.

The following procedure leads to such a policy.

- For each $j \in \mathbf{E}$, let $\pi^*_{n-1}(Y_0, A_0, Y_1, A_1, \cdots, A_{n-2}, j) = \pi^*_{n-1}(j)$ be the action which *maximizes*

$$r(j, a) + \sum_{k \in \mathbf{E}} f_n(k)\, p(k|j, a) = r(j, a) + E[f_n(Y_n)\,|Y_{n-1} = j, A_{n-1} = a] \tag{1.41}$$

  Thus, $A^*_n = \pi^*_n(Y_n)$.
- Now, $f_{n-1}(Y_{n-1}) = r\left(Y_{n-1}, A^*_{n-1}\right) - E\left[f_n(Y_n)\,|Z^*_{n-1}\right]$, which yields equality in the argument above. Thus, $R(\pi^*, j) = f_0(j)$ and π^* is optimal.

Note that π^* is a Markov policy, $A^*_n = \pi^*_n(Y_n)$. The functions f_n depend on the future stages, but once determined, the policy is Markov.

Theorem 1.20: A4-9 Doob's martingale
Let X be an integrable random variable and Z_N an arbitrary sequence of random vectors. For each n, let $X_n = E[X|W_n]$. Then $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a MG.
Proof:

$$E[|X_n|] = E\{|E[X|W_n]|\} \leq E\{E[|X|\,|W_n]\} = E[|X|] < \infty \tag{1.42}$$

$$E[X_{n+1}|W_n] = E\{E[X|W_{n+1}]\,|W_n\} = E[X|W_n] = X_n \ \ a.s. \tag{1.43}$$

Theorem 1.21: A4-9a Best mean-square estimators
If $X \in \mathbf{L}^2$, then $X_n = E[X|W_n]$ is the best mean-square estimator of X, given $W_n = (Z_0, Z_1, \cdots, Z_n)$. $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a MG.

Theorem 1.22: A4-9b Futures pricing
Let X_N be a sequence of "spot" prices for a commodity. Let t_0 be the present and $t_0 + T$ be a fixed future. The agent can be expected to know the past history $U_{t_0} = (X_0, X_1, \cdots, X_{t_0})$, and will update as t increases beyond t_0. Put $Y_k = E[X_{t_0+T}|U_{t_0+k}]$, the expected futures price, given the history up to $t_0 + k$. Then $\{Y_k : 0 \leq k \leq T\}$ is a Doob's MG, with $Y = X_{t_0+T}$, relative to $\{Z_k : 0 \leq k \leq T\}$, where $Z_0 = U_{t_0}$ and $Z_k = X_{t_0+k}$ for $1 \leq k \leq T$.

Theorem 1.23: A4-9c Discounted futures
Assume rate of return is r per unit time. Then $\alpha = 1/(1+r)$ is the *discount factor*. Let

$$V_k = E\left[\alpha^{T-k} X_{t_0+T}|U_{t_0+k}\right] = \alpha^{T-k} Y_k \tag{1.44}$$

Then

$$E[V_{k+1}|U_{t_0+k}] = \alpha^{T-k} E[Y_{k+1}|U_{t_0+k}] = \alpha^{T-k-1} Y_k > \alpha^{T-k} Y_k = V_k \ \ a.s. \tag{1.45}$$

Thus $\{V_k : 0 \leq k \leq T\}$ is a SMG relative to $\{Z_k : 0 \leq k \leq T\}$.

Implication from martingale theory is that all methods to determine profitable patterns of prediction from past history are doomed to failure.

IX A4-5

Theorem 1.24: A4-10 Present discounted value of capital
If $\alpha = 1/(1+r)$ is the discount factor, X_n is the dividend at time n, and V_n is the present value, at time n, of all future returns, then

$$V_n = \sum_{k=1}^{\infty} \alpha^k X_{n+k} \text{ so that } V_{n+1} = \sum_{k=1}^{\infty} \alpha^k X_{n+k+1} = \sum_{k=2}^{\infty} \alpha^{k-1} X_{n+k} \tag{1.46}$$

$$= \frac{1}{\alpha} \sum_{k=1}^{\infty} \alpha^k X_{n+k} - X_{n+1} = (1+r) V_n - X_{n+1} \tag{1.47}$$

Note that V_{n+1} $(\geq)$ V_n iff r $(\geq)$ X_{n+1}/V_n. Set $Y_n = E[V_n|U_n]$. Then $Y_{n+1} = (1+r) E[V_n|U_{n+1}] - X_{n+1}$ *a.s.* so that

$$E[Y_{n+1}|U_n] = (1+r) Y_n - E[X_{n+1}|U_n] \tag{1.48}$$

Thus, $(Y_{\mathbf{N}}, X_{\mathbf{N}})$ is a (S)MG iff

$$r \ (\geq) \ \frac{E[X_{n+1}|U_n]}{E[V_n|U_n]} = \frac{\text{Expected return next period, given } U_n}{\text{Expected present value, given } U_n} \tag{1.49}$$

1.2.2 Summary: Convergence of Submartingales

The submartingale convergence theorem

Theorem 1.25:
If $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a SMG with $\underset{n}{lim} E[X_n^+] < \infty$, then there exists $X_\infty \sim W_\infty$ such that $X_n \to X_\infty$ *a.s.*

Uniform integrability and some convergence conditions

Definition. The class $\{X_t : t \in T\}$ is *uniformly integrable* iff

$$sup\{E\left[I_{\{|X_t|>a\}}|X_t|\right] : t \in T\} \to 0 \text{ as } a \to \infty \tag{1.50}$$

Theorem 1.26:
Any of the following conditions ensures uniform integrability:

1. The class is dominated by an integrable random variable Y.
2. The class is finite and integrable.
3. There is a u.i. class $\{Y_t : t \in T\}$ such that $|X_t| \leq |Y_t|$ *a.s.* for all $t \in T$.
4. X integrable implies Doob's MG $\{X_n = E[X|W_n] : n \in \mathbf{N}\}$ is u.i.

Definition. The class $\{X_t : t \in T\}$ is *uniformly absolutely continuous* iff for each $\varepsilon > 0$ there is a $\delta > 0$ such that $P(A) < \delta$ implies $\underset{T}{sup}\{E[I_A|X_t|] : t \in T\} < \varepsilon$.

Theorem 1.27:
$X_T = \{X_t : t \in T\}$ is u.i. iff both (i) X_T is u.a.c., and (ii) $\underset{T}{sup}\{E[|X_t|]\} < \infty$.

Definition. $X_n \xrightarrow{P} X$ iff $P(|X_n - X| > \varepsilon) \to$ as $n \to \infty$ for all $\varepsilon > 0$.

$$X_n \xrightarrow{\mathbf{L}^p} X \quad \text{iff} \quad E[|X_n - X|^p] \to 0 \text{ as } n \to \infty \tag{1.51}$$

Theorem 1.28:
$X_n \to X$ *a.s.* implies $X_n \xrightarrow{P} X$

Theorem 1.29:
If (i) $X_n \overset{\mathbf{L}^p}{\to} X$, (ii) $X_n \to X \;\; a.s.$ and (iii) $\underset{n}{lim} E\left[X_n|Z\right]$ exists a.s., then

$$\underset{n}{lim} E\left[X_n|Z\right] = E\left[X|Z\right] \;\; a.s. \tag{1.52}$$

Theorem 1.30:
Suppose $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a (S)MG. Consider the following

(A) : $\underset{n}{lim} E\left[X_n^+\right] < \infty$ or, equivalently, $\underset{n}{sup} E\left[|X_n|\right] < \infty$ - - - - - - - - - - - -
(a) : X_N is uniformly integrable.
(a$^+$) : $X_{\mathbf{N}}^+$ is uniformly integrable.
(b) : $X_n \overset{\mathbf{L}^1}{\to} X$.
(b$^+$) : $X_n^+ \overset{\mathbf{L}^1}{\to} X$.
(c) : There is an integrable $X_\infty \sim W_\infty$ such that

$$X_n \to X_\infty \;\; a.s. \;\; \text{and} \;\; E\left[X_\infty|W_n\right] \;\; (\geq) \;\; X_n \;\; a.s. \;\; \forall\, n \in \mathbf{N} \tag{1.53}$$

(c') : Condition (c) with $\geq$ even for a MG.
(d) : There is an integrable X with $E\left[X|W_n\right] \;\; (\geq) \;\; X_n \;\; a.s. \;\; \forall\, n \in \mathbf{N}$
(d') : Condition (d) with $\geq$ even for a MG.

Then

1. Each of the propositions (a) through (d') implies (A), hence SMG convergence.
2. (a) $\Rightarrow$ (a$^+$)
3. (a) $\Leftrightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (d)
4. (a$^+$) $\Leftrightarrow$ (b$^+$) $\Leftrightarrow$ (c') $\Leftrightarrow$ (d')
5. For a MG, (d) $\Rightarrow$ (a), so that (a) $\Leftrightarrow$ (b) $\Leftrightarrow$ (c) $\Leftrightarrow$ (d)

The notion of **regularity** is characterized in terms of the conditions in the theorem.
Definition. A martingale $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is said to be **martingale regular** iff the equivalent conditions (a), (b), (c), (d) in the theorem hold.
A submartingale $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is said to be **submartingale regular** iff the equivalent conditions (a$^+$), (b$^+$), (c'), (d') in the theorem hold.
Remarks

1. Since a MG is a SMG, a martingale regular MG is also submartingale regular.
2. It is *not* true, in general that a submartingale regular SMG is martingale regular. We do have for SMG (a) $\Leftrightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (d).
3. Regularity may be viewed in terms of membership of X_∞ in the (S)MG. The condition $E\left[X_\infty|W_n\right](\geq) X_n a.s.$ is indicated by saying X_∞ **belongs** to the (S)MG or by saying the (S)MG is *closed* (on the right) by X_∞.

Summary
For a *martingale*$(X_{\mathbf{N}}, Z_{\mathbf{N}})$

1. If martingale regular, then $X_n \to X_\infty \sim W_\infty a.s.$ and $X_n = E\left[X_\infty|W_n\right] a.s. \forall n \in \mathbf{N} E\left[X_{n+k}|W_n\right] = E\{E\left[X_\infty|W_{n+k}\right]|W_n\} = E\left[X_\infty|W_n\right] = X_n a.s.$ and $E\left[X_0\right] = E\left[X_n\right] = E\left[X_\infty\right] \forall n \in \mathbf{N}$
2. If submartingale regular, but not martingale regular, then $X_n \to X_\infty \sim W_\infty a.s.$ but $E\left[X_\infty|W_n\right] \geq X_n a.s. \forall n \in \mathbf{N}$ and $E\left[X_0\right] = E\left[X_n\right] \leq E\left[X_\infty\right] < \infty \forall n \in \mathbf{N}$

For a *submartingale*$(X_{\mathbf{N}},\, Z_{\mathbf{N}})$

Either martingale regularity or submartingale regularity implies

$X_n \to X_\infty \sim W_\infty \;a.s.$ and $X_n \le E\left[X_{n+1}|W_n\right] \le E\left[X_\infty|W_n\right] \;\; a.s. \;\; \forall\, n \in \mathbf{N}$

and $E\left[X_0\right] \le E\left[X_n\right] \le E\left[X_\infty\right] < \infty \;\; \forall\, n \in \mathbf{N}$

If X_N is *uniformly integrable*, then $E\left[X_n\right] \to E\left[X_\infty\right]$.

Theorem 1.31:

If $(X_{\mathbf{N}}, Z_{\mathbf{N}})$ is a MG with $E\left[X_n^2\right] < K \forall n ß \mathbf{N}$, then the proceess is MG regular, with

$$X_n \to X_\infty \;\; a.s. \qquad E\left[(X_\infty - X_n)^2\right] \to 0 \quad \text{and} \quad E\left[X_n\right] = E\left[X_\infty\right] \;\; \forall\, n \in \mathbf{N} \tag{1.54}$$

Chapter 2

Markov Procedures for Markov Decision Processes

2.1 A Reorder problem– Electronic Store[1]

2.1.1 A reorder problem

Example 2.1
An electronic store stocks a certain type of DVD player. At the end of each week, an order is placed for early delivery the following Monday. A maximum of four units is stocked. Let the states be the number of units on hand at the end of the sales week: $\mathbf{E} = \{0, 1, 2, 3, 4\}$. Two possible actions:

- Order two units, at a cost of \$150 each
- Order four units, at a cost of \$120 each

Units sell for \$200. If demand exceeds the stock in hand, the retailer assumes a penalty of \$40 per unit (in losses due to customer dissatisfaction, etc.). Because of turnover, return on sales is considered two percent per week, so that discount is $\alpha = 1/1.02$ on a weekly basis.

In state 0, there are three possible actions: order 0, 2, or 4. In states 1 and 2 there are two possible actions: order 0 or 2. In states 3 and 4, the only action is to order 0. Customer demand in week $n+1$ is represented by a random variable D_{n+1}. The class is iid, uniformly distributed on the values 0, 1, 2, 3, 4. If X_n is the state at the end of week n, then $\{X_n, D_{n+1}\}$ is independent for each n.

Analyze the system as a Markov decision process with type 3 gains, depending upon current state, action, and demand. Determine the transition probability matrix PA (properly padded) and the gain matrix (also padded). *Sample calculations* are as follows:

State 0, action 0: $p_{00}(0) = 1(\text{all other} p_{0k}(0) = 0$

State 0, action 2: $p_{00}(2) = P(D \geq 2) = 3/5, \;\; p_{01}(2) = P(D = 1) = 1/5$, etc.

State 2, action 2: $p_{2j}(k) = 1/5, \;\; k = 0, 1, 2, 3, 4$

For state = i, action = a, and demand = k, we seek $g(i, a, k)$

$g(0,0,k) = -40k$	0 -40 -80 -120 -160
$g(0,2,k) = -300 + 200min\{k,2\} - 40max\{k-2,0\}$	-300 -100 100 60 20
$g(0,4,k) = -480 + 200k$	-480 -280 -80 120 320

[1] This content is available online at <http://cnx.org/content/m31073/1.4/>.

1. Complete the transition probability table and the gain table.
2. Determine an optimum infinite-horizon strategy with no discounting.
3. Determine an optimum infinite-horizon strateby with discounting (alpha = 1/1.02).
4. The manager decides to set up a six-week strategy, after which new sales conditions may be established. Determine an optimum strategy for the six-week period.

Data file

```
    % file orderdata.m
% Version of 4/5/94
% Data organized for computation
type = 3;
states = 0:4;
= [0  2  4 ...  % Actions (padded)
 0  2 02 ...
 0  2 02 ...
 0 00 00 ...
 0 00 00];
C  = [0 -300 -480 ... % Order costs (padded)
0 -300 -300 ...
0 -300 -300 ...
0  0  0 ...
0  0  0];
SP = 200; % Selling price
BP = 40;  % Backorder penalty
PD = 0.2*ones(1,5);  % Demand probabilities
```

2.1.2 Transition Probabilities and Gains

The procedure

```
    % file reorder.m
% Version of 4/11/94
% Calculates PA and GA for reorder policy
states = input('Enter row vector of states ');
A = input('Enter row vector A of actions (padded)  ');
C = input('Enter row vector C of order costs (padded) ');
D = input('Enter row vector D of demand values ');
PD = input('Enter row vector PD of demand probabilities ');
SP = input('Enter unit selling price SP ');
BP = input('Enter backorder penalty cost BP ');
m = length(states');
q = length(A);
na = q/m;
N = length(D);
S = ones(na,1)*states;
S = S(:)';
[d,s] = meshgrid(D,S);
a = A'*ones(1,N);
ca = C'*ones(1,N);
TA = (s + a - d).*(s + a - d >= 0);
```

```
for i = 1:q
PA(i,:) = tdbn(states,TA(i,:),PD);
end
PA
GA = ca + SP*d - (SP + BP)*(d -s -a).*(d > s+a)
```

The calculations

```
    orderdata
reorder
Enter row vector of states states
Enter row vector A of actions (padded) A
Enter row vector C of order costs (padded) C
Enter row vector D of demand values D
Enter row vector PD of demand probabilities PD
Enter unit selling price SP SP
Enter backorder penalty cost BP BP
PA =
1.0000  0       0       0       0
0.6000  0.2000  0.2000  0       0
0.2000  0.2000  0.2000  0.2000  0.2000
0.8000  0.2000  0       0       0
0.4000  0.2000  0.2000  0.2000  0
0.4000  0.2000  0.2000  0.2000  0
0.6000  0.2000  0.2000  0       0
0.2000  0.2000  0.2000  0.2000  0.2000
0.2000  0.2000  0.2000  0.2000  0.2000
0.4000  0.2000  0.2000  0.2000  0
0.4000  0.2000  0.2000  0.2000  0
0.4000  0.2000  0.2000  0.2000  0
0.2000  0.2000  0.2000  0.2000  0.2000
0.2000  0.2000  0.2000  0.2000  0.2000
0.2000  0.2000  0.2000  0.2000  0.2000
GA =
 0 -40 -80 -120 -160
 -300 -100 100 60 20
 -480 -280 -80 120 320
 0 200 160  120  80
-300 -100 100 300 260
-300 -100 100 300 260
0     200 400 360 320
-300 -100 100 300 500
-300 -100 100 300 500
0     200 400 600 560
0     200 400 600 560
0     200 400 600 560
0     200 400 600 800
0     200 400 600 800
0     200 400 600 800
```

Infinite-horizon strategy (no discounting)

```
    polit
```

```
Data needed:
- - - - - - - - - - - - - - - -
Enter case number to show gain type case
Enter row vector of states states
Enter row vector A of possible actions A
Enter value of alpha (= 1 for no discounting) 1
Enter matrix PA of transition probabilities PA
Enter matrix GA of gains GA
Enter row vector PD of demand probabilities PD
Index Action Value
0 -80
2 2 -44
3 4 -80
4 0 112
5 2 52
6 2 52
7 0 256
8 2 100
9 2 100
10 0 352
11 0 352
12 0 352
13 0 400
14 0 400
15 0 400
Initial policy: action numbers
2 1 1 1 1
Policy: actions
2 0 0 0 0

    New policy: action numbers
3 2 2 1 1
Policy: actions
4 2 2 0 0

Long-run distribution
0.2800 0.2000 0.2000 0.2000 0.1200
Test values for selecting new policy
Index Action Test Value
1.0000    0         -248.0000
2.0000    2.0000    -168.8000
3.0000    4.0000    -41.6000
4.0000    0         -48.8000
5.0000    2.0000    -5.6000
6.0000    2.0000    -5.6000
7.0000    0         131.2000
8.0000    2.0000    138.4000
9.0000    2.0000    138.4000
10.0000   0         294.4000
11.0000   0         294.4000
12.0000   0         294.4000
13.0000   0         438.4000
```

```
14.0000   0         438.4000
15.0000   0         438.4000
Optimum policy
State Action Value
0         4.0000    -168.0000
1.0000    2.0000    -132.0000
2.0000    2.0000    12.0000
3.0000    0         168.0000
4.0000    0         312.0000
Long-run expected gain per period G
126.4000
```

Infinite-horizon strategy (with discounting)

```
    polit
Data needed:
- - - - - - - - - - - - - - - -
Enter type number to show gain type
Enter row vector of states states
Enter row vector A of possible actions A
Enter value of alpha (= 1 for no discounting) 1/1.02
Enter matrix PA of transition probabilities PA
Enter matrix GA of gains GA
Enter row vector PD of demand probabilities PD

    Index Action Value
1 0 -80
2 -44
3 -80
4 0 112
5 2 52
6 2 52
7 0 256
8 2 100
9 2 100
10 0 352
11 0 352
12 0 352
13 0 400
14 0 400
15 0 400
Initial policy: action numbers
2 1 1 1 1
Policy: actions
2 0 0 0 0

New policy: action numbers
3 2 2 1 1
Policy: actions
4 2 2 0 0
Test values for selecting policy
Index Action Test Value
```

```
1.0e+03 *
0.0010 0  6.0746
0.0020 0.0020 6.1533
0.0030 0.0040 6.2776
0.0040 0      6.2740
0.0050 0.0020 6.3155
0.0060 0.0020 6.3155
0.0070 0      6.4533
0.0080 0.0020 6.4576
0.0090 0.0020 6.4576
0.0100 0      6.6155
0.0110 0      6.6155
0.0120 0      6.6155
0.0130 0      6.7576
0.0140 0      6.7576
0.0150 0      6.7576
Optimum policy
State Action Value
1.0e+03 *
0      0.0040 6.2776
0.0010 0.0020 6.3155
0.0020 0.0020 6.4576
0.0030 0      6.6155
0.0040 0      6.7576
```

Finite-horizon calculations

```
    dpinit
Initialize for finite horizon calculations
Matrices A, PA, and GA, padded if necessary
Enter case number to show gain type case
Enter vector of states states
Enter row vector A of possible actions A
Enter matrix PA of transition probabilities PA
Enter matrix GA of gains GA
Enter row vector PD of demand probabilities  PD
Call for dprog
dprog
States and expected total gains
0    1    2    3    4
-44  112  256  352  400
States Actions
0  2
1  0
2  0
3  0
4  0
dprog
States and expected total gains
0          1.0000    2.0000    3.0000   4.0000
135.2000   178.4000  315.2000  478.4000 615.2000
States Actions
0  4
```

```
1  2
2  2
3  0
4  0
dprog
States and expected total gains
0          1.0000     2.0000      3.0000     4.0000
264.4800   300.4800   444.4800    600.4800   744.4800
States Actions
0  4
1  2
2  2
3  0
4  0
dprog
States and expected total gains
1.0000  2.0000  3.0000  4.0000
390.8800 426.8800 570.8800 726.8800  870.8800
States  Actions
    0     4
    1     2
    2     2
    3     0
    4     0


    dprog
States and expected total gains
        0    1.0000    2.0000    3.0000    4.0000
 517.2800  553.2800  697.2800  853.2800  997.2800
States  Actions
    0     4
    1     2
    2     2
    3     0
    4     0
dprog
States and expected total gains
  1.0e+03 *
        0    0.0010    0.0020    0.0030    0.0040
   0.6437    0.6797    0.8237    0.9797    1.1237
States  Actions
    0     4
    1     2
    2     2
    3     0
    4     0
```

2.2 Markov Decision – Type 3 Gains[2]

Example 2.2

An electronic store stocks a certain type of VCR. At the end of each week, an order is placed for early delivery the following Monday. A maximum of four units is stocked. Let the states be the number of units on hand at the end of the sales week: $Eb = \{0,\ 1,\ 2,\ 3,\ 4\}$. Two possible actions:

- Order two units, at a cost of \$150 each
- Order four units, at a cost of \$120 each

Units sell for \$200. If demand exceeds the stock in hand, the retailer assumes a penalty of \$40 per unit (in losses due to customer dissatisfaction, etc.). Because of turnover, return on sales is considered two percent per week, so that discount is $\alpha = 1/1.02$ on a weekly basis.

In state 0, there are three possible actions: order 0, 2, or 4. In states 1 and 2 there are two possible actions: order 0 or 2. In states 3 and 4, the only action is to order 0. Customer demand in week $n+1$ is represented by a random variable D_{n+1}. The class is iid, uniformly distributed on the values 0, 1, 2, 3, 4. If X_n is the state at the end of week n, then $\{X_n, D_{n+1}\}$ is independent for each n.

Analyze the system as a Markov decision process with case 3 gains, depending upon current state, action, and demand. Determine the transition probability matrix PA (properly padded) and the gain matrix (also padded). *Sample calculations* are as follows:

- State 0, action 0: $p_{00}(0) = 1$ (all other $p_{0k}(0) = 0$)
- State 0, action 2: $p_{00}(2) = P(D \geq 2) = 3/5, p_{01}(2) = P(D = 1) = 1/5$, etc.
- State 2, action 2: $p_{2j}(k) = 1/5, k = 0,1,2,3,4$

For state = i, action = a, and demand = k, we seek $g(i,a,k)$

$g(0,0,k) = -40k$	0	−40	−80	−120	−160
$g(0,2,k) = -300 + 200min\{k,2\} - 40max\{k-2,0\}$	−300	−100	100	60	20
$g(0,4,k) = -480 + 200k$	−480	−280	−80	120	320

1. Complete the transition probability table and the gain table.
2. Determine an optimum infinite-horizon strategy with no discounting.
3. Determine an optimum infinite-horizon strateby with discounting (alpha = 1/1.02).
4. The manager decides to set up a six-week strategy, after which new sales conditions may be established. Determine an optimum strategy for the six-week period.

Data file

```
% file orderdata.m

% Version of 4/5/94

% Data organized for computation

type = 3;

states = 0:4;

A   = [0   2   4 ...                    % Actions (padded)
```

[2]This content is available online at <http://cnx.org/content/m31065/1.3/>.

```
       0   2  02 ...
       0   2  02 ...
       0  00  00 ...
       0  00  00];
C   = [0 -300 -480 ...           % Order costs (padded)
       0 -300 -300 ...
       0 -300 -300 ...
       0    0    0 ...
       0    0    0];
SP = 200;                                  % Selling price
BP = 40;                                    % Backorder penalty
PD = 0.2*ones(1,5);              % Demand probabilities
```

2.2.1 Transition Probabilities and Gains

The procedure

```
     % file reorder.m
% Version of 4/11/94
% Calculates PA and GA for reorder policy
states = input('Enter row vector of states   ');
A   = input('Enter row vector A of actions (padded)   ');
C   = input('Enter row vector C of order costs (padded)   ');
D   = input('Enter row vector D of demand values   ');
PD = input('Enter row vector PD of demand probabilities     ');
SP = input('Enter unit selling price SP   ');
BP = input('Enter backorder penalty cost BP   ');
m   = length(states');
q   = length(A);
na = q/m;
N   = length(D);
S   = ones(na,1)*states;
S   = S(:)';
[d,s] = meshgrid(D,S);
a   = A'*ones(1,N);
ca = C'*ones(1,N);
TA = (s + a - d).*(s + a - d >= 0);
for i = 1:q
PA(i,:) = tdbn(states,TA(i,:),PD);
```

```
end
PA
GA = ca + SP*d - (SP + BP)*(d -s -a).*(d > s+a)
```

The calculations

```
    orderdata
 reorder
Enter row vector of states   states
Enter row vector A of actions (padded)   A
Enter row vector C of order costs (padded)   C
Enter row vector D of demand values   D
Enter row vector PD of demand probabilities     PD
Enter unit selling price SP   SP
Enter backorder penalty cost BP   BP
PA =
      1.0000            0            0            0            0
      0.6000       0.2000       0.2000            0            0
      0.2000       0.2000       0.2000       0.2000       0.2000
      0.8000       0.2000            0            0            0
      0.4000       0.2000       0.2000       0.2000            0
      0.4000       0.2000       0.2000       0.2000            0
      0.6000       0.2000       0.2000            0            0
      0.2000       0.2000       0.2000       0.2000       0.2000
      0.2000       0.2000       0.2000       0.2000       0.2000
      0.4000       0.2000       0.2000       0.2000            0
      0.4000       0.2000       0.2000       0.2000            0
      0.4000       0.2000       0.2000       0.2000            0
      0.2000       0.2000       0.2000       0.2000       0.2000
      0.2000       0.2000       0.2000       0.2000       0.2000
      0.2000       0.2000       0.2000       0.2000       0.2000
GA =
           0          -40          -80         -120         -160
        -300         -100          100           60           20
        -480         -280          -80          120          320
           0          200          160          120           80
        -300         -100          100          300          260
        -300         -100          100          300          260
           0          200          400          360          320
        -300         -100          100          300          500
        -300         -100          100          300          500
           0          200          400          600          560
           0          200          400          600          560
           0          200          400          600          560
           0          200          400          600          800
           0          200          400          600          800
           0          200          400          600          800
```

2.2.2 Infinite-horizon strategy (no discounting)

```
    polit
Data needed:

- - - - - - - - - - - - - - - -

Enter type number to show gain type   type
Enter row vector of states   states
Enter row vector A of possible actions   A
Enter value of alpha (= 1 for no discounting)   1
Enter matrix PA of transition probabilities   PA
Enter matrix GA of gains   GA
Enter row vector PD of demand probabilities   PD
Index    Action   Value
    1         0     -80
    2         2     -44
    3         4     -80
    4         0     112
    5         2      52
    6         2      52
    7         0     256
    8         2     100
    9         2     100
   10         0     352
   11         0     352
   12         0     352
   13         0     400
   14         0     400
   15         0     400
Initial policy: action numbers
        2         1         1         1         1
Policy: actions
        2         0         0         0         0

New policy: action numbers
        3         2         2         1         1
Policy: actions
        4         2         2         0         0
Long-run distribution
      0.2800       0.2000       0.2000       0.2000       0.1200
Test values for selecting new policy
      Index         Action   Test Value
      1.0000             0    -248.0000
      2.0000        2.0000    -168.8000
      3.0000        4.0000     -41.6000
      4.0000             0     -48.8000
      5.0000        2.0000      -5.6000
      6.0000        2.0000      -5.6000
      7.0000             0     131.2000
      8.0000        2.0000     138.4000
      9.0000        2.0000     138.4000
     10.0000             0     294.4000
     11.0000            0     294.4000
     12.0000            0     294.4000
     13.0000            0     438.4000
     14.0000            0     438.4000
     15.0000            0     438.4000
    Optimum policy
      State         Action        Value
          0        4.0000    -168.0000
      1.0000       2.0000    -132.0000
      2.0000       2.0000      12.0000
      3.0000            0     168.0000
      4.0000            0     312.0000
Long-run expected gain per period G
  126.4000
```

2.2.3 Infinite-horizon strategy (with discounting)

```
    polit
Data needed:
- - - - - - - - - - - - - - - -
Enter case number to show gain type   type
Enter row vector of states   states
Enter row vector A of possible actions   A
Enter value of alpha (= 1 for no discounting)   1/1.02
Enter matrix PA of transition probabilities   PA
Enter matrix GA of gains   GA
Enter row vector PD of demand probabilities   PD
 Index    Action     Value
     1        0      -80
     2        2      -44
     3        4      -80
     4        0      112
     5        2       52
     6        2       52
     7        0      256
     8        2      100
     9        2      100
     10        0     352
     11        0     352
     12        0     352
     13        0     400
     14        0     400
     15        0     400
Initial policy: action numbers
        2         1         1         1         1
Policy: actions
        2         0         0         0         0
New policy: action numbers
        3         2         2         1         1
Policy: actions
```

```
        4          2          2          0          0
Test values for selecting policy
Index         Action     Test Value
1.0e+03 *
0.0010             0         6.0746
0.0020        0.0020         6.1533
0.0030        0.0040         6.2776
0.0040             0         6.2740
0.0050        0.0020         6.3155
0.0060        0.0020         6.3155
0.0070             0         6.4533
0.0080        0.0020         6.4576
0.0090        0.0020         6.4576
0.0100             0         6.6155
0.0110             0         6.6155
0.0120             0         6.6155
0.0130             0         6.7576
0.0140             0         6.7576
0.0150             0         6.7576
Optimum policy
State       Action       Value
1.0e+03 *
     0       0.0040       6.2776
0.0010       0.0020       6.3155
0.0020       0.0020       6.4576
0.0030            0       6.6155
0.0040            0       6.7576
```

2.2.4 Finite-horizon calculations

```
    dpinit
Initialize for finite horizon calculations
Matrices A, PA, and GA, padded if necessary
Enter type number to show gain type   type
Enter vector of states   states
Enter row vector A of possible actions   A
Enter matrix PA of transition probabilities   PA
Enter matrix GA of gains   GA
Enter row vector PD of demand probabilities   PD
Call for dprog
dprog
States and expected total gains
      0       1       2       3       4
    -44     112     256     352     400
States   Actions
     0         2
     1         0
     2         0
     3         0
     4         0
dprog
```

```
States and expected total gains
         0     1.0000     2.0000     3.0000     4.0000
  135.2000   178.4000   315.2000   478.4000   615.2000
States   Actions
     0         4
     1         2
     2         2
     3         0
     4         0
dprog
States and expected total gains
         0     1.0000     2.0000     3.0000     4.0000
  264.4800   300.4800   444.4800   600.4800   744.4800
States   Actions
     0         4
     1         2
     2         2
     3         0
     4         0
dprog
States and expected total gains
         0     1.0000     2.0000     3.0000     4.0000
  390.8800   426.8800   570.8800   726.8800   870.8800
States   Actions
     0         4
     1         2
     2         2
     3         0
     4         0
 dprog
States and expected total gains
         0     1.0000     2.0000     3.0000     4.0000
  517.2800   553.2800   697.2800   853.2800   997.2800
States   Actions
     0         4
     1         2
     2         2
     3         0
     4         0
dprog
States and expected total gains
    1.0e+03 *
           0       0.0010       0.0020       0.0030       0.0040
      0.6437       0.6797       0.8237       0.9797       1.1237
States   Actions
     0         4
     1         2
     2         2
     3         0
     4         0
```

2.3 MATLAB Calculations for Decision Models[3]

2.3.1 Data

There are three types. In all types, we need the following:

$A =$ the vector of actions	$(1 \times m)$	m	= the number of actions
$PH: \quad PH(i) = P(H = u_i)$	$(1 \times s)$	s	= the number of values of H
$PXH: \quad PXH(i, j) = P(X = x_j \vert H = u_i)$	$(s \times q)$	q	= the number of values of X

Type 1: The usual type. In addition to the above, we need

$L = [L(a, y_k)]$	$(m \times n)$	m	= the number of actions
$PYH: PYH(i, k) = P(Y = y_k \vert H = u_i)$	$(s \times n)$	n	= the number of values of Y

Type 2: The matrix $RH = [r(a, i)]$ is given. L and PYH are not needed.
Type 3: Sometimes $Y = H$. In this case $RH = L$, which we need, in addition to the above.

2.3.2 Calculated quantities

1. $RH = [r(a, i)]$ $(m \times s)$ [Risk function = expected loss, given H] $r(a, i) = E[L(a, Y) \vert H = u_i] = \sum_k L(a, k) P(Y = y_k \vert H = u_i)$ MATLAB: RH = L*PYH'
2. PX $(1 \times q)$ $PX(j) = P(X = x_j) = \sum_i P(H = u_i) P(X = j \vert H = u_i)$ MATLAB: `PX = PH*PXH`
3. PHX $(q \times s)$ $PHX(i,) = P(H = u_j \vert X = x_i) = P(X = x_i \vert H = u_i) P(H = u_j) / P(X = X_i)$ MATLAB: [a,b] = meshgrid(PH,PX) PHX = PXH'.*a./b
4. $RX = [R(a, j)]$ $(m \times q)$ [Expected risk, given X] $R(a, j) = E[r(a, H) \vert X = x_j] = \sum_i r(a, i) P(H = u_i \vert X = x_j)$ MATLAB: RX = RH*PHX'
5. Select d^* from RX: $d^*(j)$ is the action a (row number) for minimum expected loss, given $X = j$. Set $D = [d^*(1), d^*(2), \cdots d^*(q)]$.
6. Calculate the Bayesian risk BD for d^*. $BD = E[R(d^*(X), X)] = \sum_j RX(D(j), j) PX(j)$ MATLAB: RD*PX'

NOTE: Actions are represented in calculations by action number (position in the matrix). In some cases, each action has a value other than its position number. The actual values can be presented in the final display.

file dec.m

```
    %~file~dec.m
%~Version~of~12/12/95
disp('Decision~process~with~experimentation')
disp('There~are~three~types,~according~to~the~data~provided.')
disp('In~all~types,~we~need~the~row~vector~A~of~actions,')
disp('the~row~vector~PH~with~PH(i)~=~P(H~=~u_i),')
disp('the~row~vector~X~of~test~random~variable~values,~and')
disp('the~matrix~PXH~with~PXH(i,j)~=~P(X~=~x_j|H~=~u_i).')
disp('Type~1.~~Loss~matrix~L~of~L(a,k)')
disp('~~~~~~~~~Matrix~PYH~with~PYH(i,k)~=~P(Y~=~y_k|H~=~u_i)')
disp('Type~2.~~Matrix~RH~of~r(a,i)~=~E[L(a,Y)|H~=~u_i].')
```

[3] This content is available online at <http://cnx.org/content/m31068/1.4/>.

```
disp('~~~~~~~~~L~and~PYH~are~not~needed~for~this~type.')
disp('Type~3.~~Y~=~H,~so~that~only~RH~=~L~is~needed.')
c~~~=~input('Enter~type~number~~');
A~~~=~input('Enter~vector~A~of~actions~');
PH~~=~input('Enter~vector~PH~of~parameter~probabilities~~');
PXH~=~input('Enter~matrix~PXH~of~conditional~probabilities~~');
X~~~=~input('Enter~vector~X~of~test~random~variable~values~~');
s~=~length(PH);
q~=~length(X);
if~c~==~1
~L~~~=~input('Enter~loss~matrix~L~~');
~PYH~=~input('Enter~matrix~PYH~of~conditional~probabilities~~');
~RH~~=~L*PYH';
elseif~c~==~2
~RH~~=~input('Enter~matrix~RH~of~expected~loss,~given~H~~');
else
~L~~~=~input('Enter~loss~matrix~L~~');
~RH~~=~L;
end
PX~~~=~PH*PXH;~~~~~~~~%~(1~x~s)(s~x~q)~=~(1~x~q)
[a,b]~=~meshgrid(PH,PX);
PHX~=~PXH'.*a./b;~~~~~%~(q~x~s)
RX~~=~RH*PHX';~~~~~~~~%~(m~x~s)(s~x~q)~=~(m~x~q)
[RD~D]~=~min(RX);~~~~~%~determines~min~of~each~col
~~~~~~~~~~~~~~~~~~~~~~ %~and~row~on~which~min~occurs
S~=~[X;~A(D);~RD]';
BD~=~RD*PX';~~~~~~~~~~%~Bayesian~risk
h~~=~['~~Optimum~losses~and~actions'];
sh~=~['~~Test~value~~Action~~~~~Loss'];
disp('~')
disp(h)
disp(sh)
disp(S)
disp('~')
disp(['Bayesian~risk~~B(d*)~=~',num2str(BD),])
```

Example 2.3: General case

```
    %~file~dec1.m
%~Data~for~Problem~22-11
type~=~1;
A~=~[10~15];~~~~~~~~~~%~Artificial~actions~list
PH~=~[0.3~0.2~0.5];~~~%~PH(i)~=~P(H~=~i)
PXH~=~[0.7~0.2~0.1;~~~%~PXH(i,j)~=~P(X~=~j|H=~i)
~~~~~~0.2~0.6~0.2;
~~~~~~0.1~0.1~0.8];
X~=~[-1~0~~1];
L~=~[1~~0~-2;~~~~~~~~~%~L(a,k)~=~loss~when~action~number~is~a,~outcome~is~k
~~~~3~-1~-4];
PYH~=~[0.5~0.3~0.2;~~~%~PYH(i,k)~=~P(Y~=~k|H~=~i)
~~~~~~0.2~0.5~0.3;
~~~~~~0.1~0.3~0.6];
```

```
~
dec1
dec
Decision~process~with~experimentation
There~are~three~types,~according~to~the~data~provided.
In~all~types,~we~need~the~row~vector~A~of~actions,
the~row~vector~PH~with~PH(i)~=~P(H~=~i),
the~row~vector~X~of~test~random~variable~values,~and
the~matrix~PXH~with~PXH(i,j)~=~P(X~=~j|H~=~i).
Type~1.~~Loss~matrix~L~of~L(a,k)
~~~~~~~~Matrix~PYH~with~PYH(i,k)~=~P(Y~=~k|H~=~i)
Type~2.~~Matrix~RH~of~r(a,i)~=~E[L(a,Y)|H~=~i].
~~~~~~~~L~and~PYH~are~not~needed~in~this~case.
Type~3.~~Y~=~H,~so~that~only~RH~=~L~is~needed.
Enter~type~number~~type
Enter~vector~A~of~actions~A
Enter~vector~PH~of~parameter~probabilities~~PH
Enter~matrix~PXH~of~conditional~probabilities~~PXH
Enter~vector~X~of~test~random~variable~values~~X
Enter~loss~matrix~L~~L
Enter~matrix~PYH~of~conditional~probabilities~~PYH
~
~Optimum~losses~and~actions
~Test~value~~Action~~~~~Loss
~~-1.0000~~~15.0000~~~-0.2667
~~~~~~~~0~~~15.0000~~~-0.9913
~~~1.0000~~~15.0000~~~-2.1106
~
Bayesian~risk~~B(d*)~=~-1.3
```

Intermediate steps in solution of Example 1, to show results of various operations

```
    RH
RH~~=~~0.1000~~~-0.4000~~~-1.1000
~~~~~~0.4000~~~-1.1000~~~-2.4000
PX
PX~~=~~0.3000~~~~0.2300~~~~0.4700
a
a~~~=~~0.3000~~~~0.2000~~~~0.5000
~~~~~~0.3000~~~~0.2000~~~~0.5000
~~~~~~0.3000~~~~0.2000~~~~0.5000
b
b~~~=~~0.3000~~~~0.3000~~~~0.3000
~~~~~~0.2300~~~~0.2300~~~~0.2300
~~~~~~0.4700~~~~0.4700~~~~0.4700
PHX
PHX~=~~0.7000~~~~0.1333~~~~0.1667
~~~~~~0.2609~~~~0.5217~~~~0.2174
~~~~~~0.0638~~~~0.0851~~~~0.8511
RX
RX~~=~-0.1667~~~-0.4217~~~-0.9638
~~~~~-0.2667~~~-0.9913~~~-2.1106
```

Example 2.4: *RH* given

```
    %~file~dec2.m~~
%~Data~for~type~in~which~RH~is~given
type~=~2;
A~=~[1~2];
X~=~[-1~1~3];
PH~=~[0.2~0.5~0.3];
PXH~=~[0.5~0.4~0.1;~~~%~PXH(i,j)~=~P(X~=~j|H~=~i)
~~~~~~0.4~0.5~0.1;
~~~~~~0.2~0.4~0.4];
RH~=~[-10~~~5~-12;
~~~~~~~5~-10~~-5];~~~~%~r(a,i)~=~expected~loss~when
~~~~~~~~~~~~~~~~~~~~~~%~~~action~is~a,~given~H~=~i
~
dec2
dec
Decision~process~with~experimentation
-------------------~Instruction~lines~edited~out
Enter~type~number~~type
Enter~vector~A~of~actions~A
Enter~vector~PH~of~parameter~probabilities~~PH
Enter~matrix~PXH~of~conditional~probabilities~~PXH
Enter~vector~X~of~test~random~variable~values~~X
Enter~matrix~RH~of~expected~loss,~given~H~~RH
~
~Optimum~losses~and~actions
~Test~value~~Action~~~~~Loss
~~-1.0000~~~~2.0000~~~-5.0000
~~~1.0000~~~~2.0000~~~-6.0000
~~~3.0000~~~~1.0000~~~-7.3158
~
Bayesian~risk~~B(d*)~=~-5.89
```

Example 2.5: Example 3

Carnival example (type in which $Y = H$)

A carnival is scheduled to appear on a given date. Profits to be earned depend heavily on the weather. If rainy, the carnival loses \$15 (thousands); if cloudy, the loss is \$5 (thousands); if sunny, a profit of \$10 (thousands) is expected. If the carnival sets up its equipment, it must give the show; if it decides not to set up, it forfeits \$1,000. For an additional cost of \$1,000, it can delay setup until the day before the show and get the latest weather report.

Actual weather $H = Y$ is 1 rainy, 2 cloudy, or 3 sunny.

The weather report X has values 1, 2, or 3, corresponding to predictions rainy, cloudy, or sunny respectively.

Reliability of the forecast is expressed in terms of $P(X = j|H = i)$– see matrix PXH

Two actions: 1 set up; 2 no set up.

Possible losses for each action and weather condition are in matrix L.

```
    %~file~dec3,m
%~Carnival~problem
type~=~3;~~~~~~~~~~~~~%~Y~=~H~~(actual~weather)
```

```
A~=~[1~~2];~~~~~~~~~~~%~1:~setup~~2:~no~setup
X~=~[1~~2~~3];~~~~~~~~%~1;~rain,~~2:~cloudy,~3:~sunny
L~=~[16~6~-9;~~~~~~~~~%~L(a,k)~=~loss~when~action~number~is~a,~outcome~is~k
~~~~~2~2~~2];~~~~~~~ ~%~--with~premium~for~postponing~setup
PH~=~0.1*[1~3~6];~~~~~%~P(H~=~i)
PXH~=~0.1*[7~2~1;~~~~~%~PXH(i,j)~=~P(X~=~j|H~=~i)
~~~~~~~~~~2~6~2;
~~~~~~~~~~1~2~7];
~
dec3
dec
Decision~process~with~experimentation
-------------------~Instruction~lines~edited~out
Enter~case~number~~case
Enter~vector~A~of~actions~A
Enter~vector~PH~of~parameter~probabilities~~PH
Enter~matrix~PXH~of~conditional~probabilities~~PXH
Enter~vector~X~of~test~random~variable~values~~X
Enter~loss~matrix~L~~L
~
~Optimum~losses~and~actions
~Test~value~~Action~~~~~Loss
~~~1.0000~~~~2.0000~~~~2.0000
~~~2.0000~~~~1.0000~~~~1.0000
~~~3.0000~~~~1.0000~~~-6.6531
~
Bayesian~risk~~B(d*)~=~-2.56
```

2.4 Matlab Procedures for Markov Decision Processes[4]

In order to provide the background for Matlab procedures for Markov decision processes, we first summarize certain essentials in the analysis of homogeneous Markov chains with costs and rewards associated with states, or with transitions between states. References are to Pfeiffer: PROBABILITY FOR APPLICATIONS.

1. Some cost and reward patterns
   Consider a finite, ergodic (i.e., recurrent, positive, aperiodic, irreducible) homogeneous Markov chain X_N, with state space $\mathbf{E} = \{1, 2, \cdots, M\}$. To facilitate use of matrix computation programs, we number states from one, except in certain examples in which state zero has a natural meaning. Associated with this chain is a cost or reward structure belonging to one of the three general classes described below. The one-step transition probability matrix is $\mathbf{P} = [p(i,j)]$, where $p(i,j) = P(X_{n+1} = j | X_n = i)$. The distribution for X_n is represented by the row matrix $\pi(n) = [p_1(n), p_2(n), \cdots, p_M(n)]$, where $p_i(n) = P(X_n = i)$. The long-run distribution is represented by the *row matrix* $\pi = [\pi_1, \pi_2, \cdots, \pi_M]$.

   a. **Type 1. Gain associated with a state**. A reward or gain in the amount $g(i)$ is realized in the next period if the current state is i. The **gain sequence**$\{G_n : 1 \leq n\}$ of random variables $G_{n+1} = g(X_n)$ is the sequence of gains realized as the chain X_N evolves. We represent the gain function g by the column matrix $\mathbf{g} = [g(1)\, g(2) \cdots g(M)]^T$.

[4]This content is available online at <http://cnx.org/content/m31095/1.7/>.

b. **Type 2. One-step transition gains**. A reward or gain in the amount $g(i,j) = g_{ij}$ is realized in period $n+1$ if the system goes from state i in period n to state j in period $n+1$. The corresponding one-step transition probability is $p(i,j) = p_{ij}$. The gain matrix is $\mathbf{g} = [g(i,j)]$. The **gain sequence**$\{G_n : 1 \leq n\}$ of random variables $G_{n+1} = g(X_n, X_{n+1})$ is the sequence of gains experienced as the chain X_N evolves.

c. **Type 3. Gains associated with a demand**. In this case, the gain random variables are of the form

$$G_{n+1} = g(X_n, D_{n+1}) \tag{2.1}$$

If n represents the present, the random vector $U_n = (X_0, X_1, \cdots, X_n)$ represents the "past" at n of the chain X_N. We suppose $\{D_n : 1 \leq n\}$ is iid and each $\{D_{n+1}, U_n\}$ is independent. Again, the **gain sequence**$\{G_n : 1 \leq n\}$ represents the gains realized as the process evolves. Standard results on Markov chains show that in each case the sequence $G_{\mathbf{N}} = \{G_n : 1 \leq n\}$ is Markov.

A recurrence relation. We are interested in the expected gains in future stages, given the present state of the process. For any $n > 0$ and any $m > 1$, the gain $G_n^{(m)}$ in the m periods following period n is given by

$$G_n^{(m)} = G_{n+1} + G_{n+2} + \cdots + G_{n+m} \tag{2.2}$$

If the process is in state i, the expected gain in the next period q_i is

$$q_i = v_i^{(1)} = E[G_{n+1}|X_n = i] \tag{2.3}$$

and the expected gain in the next m periods is

$$v_i^{(m)} = E\left[G_n^{(m)}|X_n = i\right] \tag{2.4}$$

We utilize a recursion relation that allows us to begin with the transition matrix $\mathbf{P}$ and the *next-period expected gain matrix*$\mathbf{q} = [q_1 q_2 \cdots q_m]^T$ and calculate the $v_i^{(m)}$ for any $m > 1$. Although the analysis is somewhat different for the various gain structures, the result exhibits a common pattern. In each case

$$v_i^{(m)} = E\left[G_n^{(m)}|X_n = i\right] = E[G_{n+1}|X_n = i] + \sum_{k=1}^{m-1} E[G_{n+k+1}|X_n = i] \tag{2.5}$$

$$= q_i + \sum_{k=1}^{m-1} \sum_{j \in E} E[G_{n+k+1}|X_{n+1} = j]\, p(i,j) \tag{2.6}$$

$$= q_i + \sum_{j \in E} E\left[\sum_{k=1}^{m-1} G_{n+k}|X_n = j\right] p(i,j) \tag{2.7}$$

We thus have the fundamental recursion relation

$$(*)\; v_i^{(m)} = q_i + \sum_{j \in E} v_j^{(m-1)} p(i,j) \tag{2.8}$$

The recursion relation $(*)$ shows that the transition matrix $\mathbf{P} = [p(i,j)]$ and the vector of next-period expected gains, which we represent by the column matrix $\mathbf{q} = [q_1, q_2, \cdots, q_M]^T$, determine the $v_i^{(m)}$. The difference between the cases is the manner in which the q_i are obtained.

Type 1: . $q_i = E[g(X_n)|X_n = i] = g(i)$
Type 2: . $q_i = E[g(X_n, X_{n+1})|X_n = i] = E[g(i, X_{n+1})|X_n = i] = \sum_{j \in E} g(i,j)\, p(i,j)$
Type 3: . $q_i = E[g(X_n, D_{n+1})|X_n = i] = E[g(i, D_{n+1})] = \sum_k g(i,k)\, P(D = k)$

Matrix form: . For computational purposes, we utilize these relations in matrix form. If

$$\mathbf{v}(n) = \left[v_1^{(n)} v_2^{(n)} \cdots v_M^{(n)}\right]^T \text{ and } \mathbf{q} = [q_1 q_2 \cdots q_M]^T \tag{2.9}$$

then

$$(*)\, \mathbf{v}(m+1) = \mathbf{q} + \mathbf{P}\mathbf{v}(m) \text{ for all } m > 0, \text{ with } \mathbf{v}(0) = 0 \tag{2.10}$$

Examination of the expressions for q_i, above, shows that the next-period expected gain matrix $\mathbf{q}$ takes the following forms. In each case, $\mathbf{g}$ is the gain matrix. In case c, p_D is the column matrix whose elements are $P(D = k)$.

Type 1: $\mathbf{q} = \mathbf{g}$
Type 2: $\mathbf{q} =$ the diagonal of $\mathbf{P}\mathbf{g}^T$
Type 3: $\mathbf{q} = \mathbf{g}\mathbf{p}_D$

2. Some long-run averages
Consider the average expected gain for m periods

$$E\left[\frac{1}{m} G_n^{(m)}\right] = \frac{1}{m}\sum_{k=1}^{m} E[G_{n+k}] = \frac{1}{m}\sum_{k=1}^{m} E\{E[G_{n+k}|X_{n-1}]\} \tag{2.11}$$

Use of the Markov property and the fact that for an ergodic chain

$$\frac{1}{m}\sum_{k=1}^{m} p^k(i,j) \to \pi_j \text{ as } m \to \infty \tag{2.12}$$

yields the result that,

$$\lim_{m\to\infty} E\left[\frac{1}{m} G_n^{(m)}\right] = \sum_i P(X_{n-1} = i) \sum_j q_j \pi_j = \sum_j q_j \pi_j = g \tag{2.13}$$

and for any state i,

$$\lim_{m\to\infty} E\left[\frac{1}{m} G_n^{(m)} | X_n = i\right] = \lim_{m\to\infty} \frac{1}{m} v_i^{(m)} = g \tag{2.14}$$

The expression for g may be put into matrix form. If the long-run probability distribution is represented by the row matrix $\pi = [\pi_1 \pi_2 \cdots \pi_M]$ and $\mathbf{q} = [q_1 q_2 \cdots ; q_M]^T$, then

$$g = \pi\mathbf{q} \tag{2.15}$$

3. Discounting and potentials
Suppose in a given time interval the value of money increases by a fraction a. This may represent the potential earning if the money were invested. One dollar now is worth $1 + a$ dollars at the end of one period. It is worth $(1 + a)^n$ dollars after n periods. Set $\alpha = 1/(1 + a)$, so that $0 < \alpha \leq 1$. It takes α dollars to be worth one dollar after one period. It takes α^n dollars at present to be worth one dollar n periods later. Thus the **present worth** or **discounted value** of one dollar n periods in the future is α^n dollars. This is the amount one would need to invest presently, at interest rate a per unit of time, to have one dollar n periods later. If f is any function defined on the state space $\mathbf{E}$, we designate by $\mathbf{f}$ the column matrix $[f(1) f(2) \cdots f(M)]^T$. We make an exception to this convention in the case of the distributions of the state probabilities $\pi(n) = [p_1(n) p_2(n) \cdots p_M(n)]$, their generating function, and the long-run probabilities $\pi = [\pi_1 \pi_2 \cdots \pi_M]$, which we represent as row matrices. It should be clear that much of the following development extends immediately to infinite state space $\mathbf{E} = \mathbf{N} = \{0, 1, 2, \cdots\}$. We assume one of the gain structures introduced in Sec 1 and the corresponding gain sequence $\{G_n : 1 \leq n\}$. The value of the random variable G_{n+1} is the gain or reward realized

during the period $n+1$. We let each transition time be at the end of one period of time (say a month or a quarter). If n corresponds to the present, then G_{n+k} is the gain in period $n+k$. If we do not discount for the period immediately after the present, then $\alpha^{k-1}G_{n+k}$ is the present value of that gain. Hence

$$\sum_{k=1}^{\infty} \alpha^{k-1} G_{n+k} \text{ is the total discounted future gain} \tag{2.16}$$

Definition. The *α-potential* of the gain sequence $\{G_n : 1 \le n\}$ is the function ϕ defined by

$$\phi(i) = E\left[\sum_{n=0}^{\infty} \alpha^n G_{n+1} | X_0 = i\right] \forall i \in \mathbf{E} \tag{2.17}$$

Remark. $\phi(i)$ is the expected total discounted gain, given the system starts in state i. We next define a concept which is related to the α-potential in a useful way. **Definition**. The *α-potential matrix*$\mathbf{R}^\alpha$ for the process X_N is the matrix

$$\mathbf{R}^\alpha = \sum_{n=0}^{\infty} \alpha^n \mathbf{P}^n \text{ with } \mathbf{P}^0 = \mathbf{I} \tag{2.18}$$

Theorem 2.1: 3.1
Let X_N be an ergodic, homogeneous Markov chain with state space $\mathbf{E}$ and gain sequence $\{G_n : 1 \le n\}$. Let $\mathbf{q} = [q_1 q_1 \cdots q_M]^T$ where $q_i = E[G_{n+1}|X_n = i]$ for$i \in \mathbf{E}$. For $\alpha \in (0,1)$, let ϕ be the α-potential for the gain sequence. That is,

$$\phi(i) = E\left[\sum_{n=0}^{\infty} \alpha^n G_{n+1} | X_0 = i\right] \forall i \in \mathbf{E} \tag{2.19}$$

Then, if $\mathbf{R}^\alpha$ is the α-potential matrix for X_N, we have

$$\phi = \mathbf{R}^\alpha \mathbf{q} \tag{2.20}$$

— □

Theorem 2.2: 3.2
Consider an ergodic, homogeneous Markov chain X_N with gain sequence $\{G_n : 1 \le n\}$ and next-period expected gain matrix $\mathbf{q}$. For $\alpha \in (0,1)$, the α-potential ϕ and the α-potential matrix $\mathbf{R}^\alpha$ satisfy

$$[\mathbf{I} - \alpha\mathbf{P}]\,\mathbf{R}^\alpha = \mathbf{I}\text{and}\,[\mathbf{I} - \alpha\mathbf{P}]\,\phi = \mathbf{q} \tag{2.21}$$

If the state space $\mathbf{E}$ is finite, then

$$\mathbf{R}^\alpha = [\mathbf{I} - \alpha\mathbf{P}]^{-1}\text{and}\phi = [\mathbf{I} - \alpha\mathbf{P}]^{-1}\mathbf{q} = \mathbf{R}^\alpha \mathbf{q} \tag{2.22}$$

— □

Example 2.6: A numerical example
Suppose the transition matrix $\mathbf{P}$ and the next-period expected gain matrix $\mathbf{q}$ are

$$\mathbf{P} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.4 & 0.1 & 0.5 \\ 0.6 & 0.3 & 0.1 \end{bmatrix} \text{ and } \mathbf{q} = \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix} \tag{2.23}$$

For $\alpha = 0.9$, we have

$$\mathbf{R}^{0.9} = (\mathbf{I} - 0.9\mathbf{P})^{-1} = \begin{bmatrix} 4.4030 & 2.2881 & 3.3088 \\ 3.5556 & 3.1356 & 3.3088 \\ 3.6677 & 2.2881 & 4.0441 \end{bmatrix} \text{ and } \phi = \mathbf{R}^{0.9}\mathbf{q} = \begin{bmatrix} 30.17 \\ 32.72 \\ 30.91 \end{bmatrix} \tag{2.24}$$

— □

The next example is of class c, but the demand random variable is Poisson, which does not have finite range. The simple pattern for calculating the q_i must be modified.

Example 2.7: Costs associated with inventory under an (m, M) order policy.
If k units are ordered, the cost in dollars is

$$c(k) = \{ \begin{matrix} 0 & k = 0 \\ 10 + 25k & 0 < k \leq M \end{matrix} \tag{2.25}$$

For each unit of unsatisfied demand, there is a penalty of 50 dollars. We use the (m, M) inventory policy described in Ex 23.1.3. $X_0 = M$, and X_n is the stock at the end of period n, before restocking. Demand in period n is D_n. The cost for restocking at the end of period $n+1$ is

$$g(X_n, D_{n+1}) = \{ \begin{matrix} 10 + 25(M - X_n) + 50max\{D_{n+1} - M, 0\} & 0 \leq X_n < m \\ 50max\{D_{n+1} - X_n, 0\} & m \leq X_n \leq M \end{matrix} \tag{2.26}$$

For $m = 1, M = 3$, we have

$$g(0, D_{n+1}) = 85 + 50 I_{[3,\infty)}(D_{n+1})(D_{n+1} - 3) \tag{2.27}$$

$$g(i, D_{n+1}) = 50 I_{[3,\infty)}(D_{n+1})(D_{n+1} - i)\, 1 \leq i \leq 3 \tag{2.28}$$

We take $m = 1, M = 3$ and assume D is Poisson (1). From Ex 23.1.3 in PA, we obtain

$$\mathbf{P} = \begin{bmatrix} 0.0803 & 0.1839 & 0.3679 & 0.3679 \\ 0.6321 & 0.3679 & 0 & 0 \\ 0.2642 & 0.3679 & 0.3679 & 0 \\ 0.0803 & 0.1839 & 0.3679 & 0.3679 \end{bmatrix} \tag{2.29}$$

The largest eigenvalue $|\lambda| \approx 0.26$, so $n \geq 10$ should be sufficient for convergence. We use $n = 16$.
Taking any row of $\mathbf{P}^{16}$, we approximate the long-run distribution by

$$\pi = [0.2858 \;\; 0.2847 \;\; 0.2632 \;\; 0.1663] \tag{2.30}$$

We thus have

$$q_0 = 85 + 50E\left[I_{[3,\infty)}(D_{n+1})(D_{n+1} - 3)\right] = 85 + 50\sum_{k=4}^{\infty}(k-3)p_k \text{ (the term for } k = 3 \text{ is zero)} \tag{2.31}$$

For the Poisson distribution $\sum_{k=n}^{\infty} kp_k = \lambda \sum_{k=n-1}^{\infty} p_k$.
Hence $q_0 = 85 + 50\left[\sum_{k=3}^{\infty} p_k - 3\sum_{k=4}^{\infty} p_k\right] \approx 85 + 50\,[0.0803 - 3 \times 0.0190] = 86.1668$ $q_1 =$

$50\sum_{k=3}^{\infty}(k-1)\,p_k = 50\left[\sum_{k=2}^{\infty}p_k - \sum_{k=3}^{\infty}p_k\right] = 50p_2 = 9.1970$ $q_2 = 50\sum_{k=3}^{\infty}(k-2)\,p_k = 50\,[0.2642 - 2\times 0.0803] = 5.1809$ $q_3 = q_0 - 85 = 1.1668$ so that

$$\mathbf{q} = \begin{bmatrix} 86.1668 \\ 9.1970 \\ 5.1809 \\ 1.1668 \end{bmatrix} \tag{2.32}$$

Then, for $\alpha = 0.8$ we have

$$\mathbf{R}^{0.8} = (\mathbf{I} - 0.8\mathbf{P})^{-1} = \begin{bmatrix} 1.9608 & 1.0626 & 1.1589 & 0.8178 \\ 1.4051 & 2.1785 & 0.8304 & 0.5860 \\ 1.1733 & 1.2269 & 2.1105 & 0.4893 \\ 0.9608 & 1.0626 & 1.1589 & 1.8178 \end{bmatrix} \quad \text{and} \quad \phi = \mathbf{R}^{0.8}\mathbf{q} = \begin{bmatrix} 185.68 \\ 146.09 \\ 123.89 \\ 100.68 \end{bmatrix} \tag{2.33}$$

Recall that we are dealing with expected discounted costs. Since we usually consider starting with a full stock, the amount $\phi(M) = \phi(3) = 100.68$ is the quantity of interest. Note that this is smaller than other values of $\phi(j)$, which correspond to smaller beginning stock levels. We expect greater restocking costs in such cases.

4. Evolution of chains with costs and rewards
   a. No discounting A generating function analysis of

$$(*) \quad \mathbf{v}(m+1) = \mathbf{q} + \mathbf{P}\mathbf{v}(m) \quad \text{for all } m > 0, \ \text{with } \mathbf{v}(0) = 0 \tag{2.34}$$

shows

$$\mathbf{v}(n) = ng\mathbf{1} + \mathbf{v} + \text{transients}, \quad \text{where } \mathbf{v} = \mathbf{B}_0\mathbf{q} \tag{2.35}$$

Here $g = \pi\mathbf{q}$, is a column matrix of ones, $\mathbf{P}_0 = \mathbf{P}^{\infty}$, and B_0 is a matrix which analysis also shows we may approximate by

$$\mathbf{B}_0 = \mathbf{B}(1) \approx \mathbf{I} + \mathbf{P} + \mathbf{P}^2 + \cdots + \mathbf{P}^n - (n+1)\,\mathbf{P}_0 \tag{2.36}$$

The largest $|\lambda_i| < 1$ gives an indication of how many terms are needed.

Example 2.8: The inventory problem (continued)
We consider again the inventory problem. We have

$$\mathbf{P} = \begin{bmatrix} 0.0803 & 0.1839 & 0.3679 & 0.3679 \\ 0.6321 & 0.3679 & 0 & 0 \\ 0.2642 & 0.3679 & 0.3679 & 0 \\ 0.0803 & 0.1839 & 0.3679 & 0.3679 \end{bmatrix} \quad \text{and} \quad \mathbf{q} = \begin{bmatrix} 86.1668 \\ 9.1970 \\ 5.1819 \\ 1.1668 \end{bmatrix} \tag{2.37}$$

The eigenvalues are $1, 0.0920 + i0.2434, 0.0920 - 0.2434$ and 0. Thus, the decay of the transients is controlled by $|\lambda^*| = \left(0.0920^2 + 0.2434^2\right)^{1/2} = 0.2602$. Since $|\lambda^*|^4 \approx 0.0046$, the transients die out rather rapidly. We obtain the following approximations

$$\mathbf{P}_0 \approx \mathbf{P}^{16} \approx \begin{bmatrix} 0.2852 & 0.2847 & 0.2632 & 0.1663 \\ 0.2852 & 0.2847 & 0.2632 & 0.1663 \\ 0.2852 & 0.2847 & 0.2632 & 0.1663 \\ 0.2852 & 0.2847 & 0.2632 & 0.1663 \end{bmatrix} \text{ so that } \pi \approx [0.2852\ 0.2847\ 0.2632\ 0.1663] \tag{2.38}$$

The approximate value of B_0 is found to be

$$\mathbf{B}_0 \approx \mathbf{I} + \mathbf{P} + \mathbf{P}^2 + \mathbf{P}^3 + \mathbf{P}^4 - 5\mathbf{P}_0 \approx \begin{bmatrix} 0.4834 & -0.3766 & -0.1242 & 0.0174 \\ 0.0299 & 0.7537 & -0.5404 & -0.2432 \\ -0.2307 & -0.1684 & 0.7980 & -0.3989 \\ -0.5166 & -0.3766 & -0.1242 & 1.0174 \end{bmatrix} \tag{2.39}$$

The value $g = \pi\mathbf{q} \approx 28.80$ is found in the earlier treatment. From the values above, we have

$$\mathbf{v} = \mathbf{B}_0\mathbf{q} \approx \begin{bmatrix} 37.6 \\ 6.4 \\ -17.8 \\ -47.4 \end{bmatrix} \text{ so that } \mathbf{v}(n) \approx \begin{bmatrix} 28.80n + 37.6 \\ 28.80n + 6.4 \\ 28.80n - 17.8 \\ 28.80n - 47.4 \end{bmatrix} + \text{ transients} \tag{2.40}$$

The average gain per period is clearly $g \approx 28.80$. This soon dominates the constant terms represented by the entries in $\mathbf{v}$.

b. With discounting Let the discount factor be $\alpha \in (0, 1)$. If n represents the present period, then G_{n+1} = the reward in the first succeeding period G_{n+k} = the reward in the kth succeding period. If we do not discount the first period, then

$$G_n^{(m)} = G_{n+1} + \alpha G_{n+2} + \alpha^2 G_{n+3} + \cdots + \alpha^{m-1} G_{n+m} = G_{n+1} + \alpha G_{n+1}^{(m-1)} \tag{2.41}$$

Thus

$$v_i^{(m)} = E\left[G_n^{(m)} | X_n = i\right] = q_i + \alpha \sum_{j=1}^{M} p(i,\, j)\, v_j^{(m-1)} \tag{2.42}$$

In matrix form, this is

$$\mathbf{v}(n) = \mathbf{q} + \alpha\mathbf{P}\mathbf{v}(n-1) \tag{2.43}$$

A generating function analysis shows that

$$v_i^{(n)} = v_i + \text{transients} \quad 1 \le i \le M \tag{2.44}$$

Hence, the steady state part of

$$v_i^{(m)} = q_i + \alpha \sum_{j=1}^{M} p(i,j)\, v_j^{(m-1)} \text{ is } v_i = q_i + \alpha \sum_{j=1}^{M} p(i,j)\, v_j 1 \le i \le M \tag{2.45}$$

In matrix form,

$$\mathbf{v} = \mathbf{q} + \alpha\mathbf{P}\mathbf{v} \quad \text{which yields} \quad \mathbf{v} = [\mathbf{I} - \alpha\mathbf{P}]^{-1}\mathbf{q} \tag{2.46}$$

Since the $q_i = E\left[G_{n+1}|X_n = i\right]$ are known, we can solve for the v_i. Also, since $v_i^{(m)}$ is the present value of expected gain m steps into the future, the v_i represent the present value for an unlimited future, given that the process starts in state i.

5. Stationary decision policies
We suppose throughout this section that X_N is ergodic, with finite state space $\mathbf{E} = \{1, 2, \cdots, M\}$. For each state $i \in \mathbf{E}$, there is a class A_i of possible **actions** which may be taken when the process is in state i. A **decision policy** is a sequence of decision functions $d_1, d_2 \cdots$ such that

$$\text{The action at stage } n \text{ is } d_n\left(X_0, X_1, ..., X_n\right). \tag{2.47}$$

The action selected is in A_i whenever $X_n = i$. We consider a special class of decision policies. **Stationary decision policy**

$$d_n\left(X_0, X_1, \cdots, X_n\right) = d\left(X_n\right) \quad \text{with } d \text{ invariant with } n \tag{2.48}$$

The possible actions depend upon the current state. That is $d\left(X_n\right) \in A_i$ whenever $X_n = i$. Analysis shows the Markov character of the process is preserved under stationary policies. Also, if $\mathbf{E}$ is finite and every stationary policy produces an irreducible chain, then an optimal stationary policy is optimal. *We suppose each policy yields an ergodic chain.* Since the transition probabilities from any state depend in part on the action taken when in that state, the long-run probabilities will depend upon the policy. In the no-discounting case, we want to determine a stationary policy that maximizes the gain $g = \pi\mathbf{q}$ for the corresponding long-run probabilities. We say "a stationary policy," since we do not rule out the possibility there may be more than one policy which maximizes the gain. In the discounting case, the goal is to maximize the steady state part v_i in the expressions

$$v_i^{(n)} = v_i + \text{ transients} \quad 1 \le i \le M \tag{2.49}$$

In simple cases, with small state space and a small number of possible actions in each state, it may be feasible to obtain the gain or present values for each permissible policy and then select the optimum by direct comparison. However, for most cases, this is not an efficient approach. In the next two sections, we consider iterative procedures for step-by-step optimization which usually greatly reduces the computational burden.

6. Policy iteration method– no discounting
We develop an iterative procedure for finding an optimal stationary policy. The goal is to determine a stationary policy that maximizes the long run expected gain per period g. In the next section, we extend this procedure to the case where discounting is in effect. *We assume that each policy yields an ergodic chain.* Suppose we have established that *under a given policy* (1) $v_i^{(n)} = q_i + \sum_j p(i, j)\, v_j^{(n-1)}$, where $q_i = E\left[G_{n+1}|X_n = i\right]$ In this case, the analysis in Sec 4 shows that for large n, (2) $v_i^{(n)} \approx ng + v_i$, where $g = \sum_i \pi_i q_i$ We note that v_i and g depend upon the entire policy, while q_i and $p(i, j)$ depend on the individual actions a_i. Using (1) and (2), we obtain

$$ng + v_i = q_i + \sum_i p(i, j)\left[(n-1)\,g + v_j\right] = q_i + (n-1)\,g + \sum_j p(i, j)\, v_j \tag{2.50}$$

From this we conclude (3) $g + v_i = q_i + \sum_j p(i, j)\, v_j$ for all $i \in \mathbf{E}$ Suppose a policy d has been used. That is, action $d(i)$ is taken whenever the process is in state i. To simplify writing, we drop the indication of the action and simply write $p(i, j)$ for $p_ij\left(d(i)\right)$, etc. Associated with this policy, there is a gain g. We should like to determine whether or not this is the maximum possible gain, and if it is not, to find a policy which does give the maximum gain. The following two-phase procedure has been found to be effective. It is essentially the procedure developed originally by Ronald Howard, who pioneered the treatment of these problems. The new feature is the method of carrying out the value-determination step, utilizing the approximation method noted above.

a. **Value-determination procedure** We calculate $g = \sum_i \pi_i q_i = \pi\mathbf{q}$. As in Sec 4, we calculate

$$\mathbf{v} = \mathbf{B}_0\mathbf{q} \approx \left[\mathbf{I} + \mathbf{P} + \mathbf{P}^2 + \cdots + \mathbf{P}^s - (s+1)\,\mathbf{P}_0\right]\mathbf{q} \quad \text{where } \mathbf{P}_0 = \underset{n}{lim}\mathbf{P}^n \tag{2.51}$$

b. **Policy-improvement procedure** We suppose policy d has been used through period n. Then, by (3), above,

$$g + v_i = q_i + \sum_j p(i,\, j)\, v_j \tag{2.52}$$

We seek to improve policy d by selecting policy d^*, with $d^*(i) = a^*_{ik}$, to satisfy

$$q_i^* + \sum_j p_{ij}^* v_j = max\{q_i(a_{ik}) + \sum_j p_{ij}(a_{ik})\, v_j \;:\; a_{ik} \in A_i\}, \quad 1 \le i \le M \tag{2.53}$$

Remarks

- In the procedure for selecting d^*, we use the "old" v_j.
- It has been established that $g^* \ge g$, with equality iff g is optimal. Since there is a finite number of policies, the procedure must converge to an optimum stationary policy in a finite number of steps.

We implement this two step procedure with Matlab. To demonstrate the procedure, we consider the following

Example 2.9: A numerical example
A Markov decision process has three states: State space $\mathbf{E} = \{1, 2, 3\}$.
Actions: State 1: $A_1 = \{1, 2, 3\}$ State 2: $A_2 = \{1, 2\}$ State 3: $A_3 = \{1, 2\}$
Transition probabilities and rewards are:

$p_{1j}(1)$: [1/3 1/3 1/3]	$g_{1j}(1)$: [1 3 4]
$p_{1j}(2)$: [1/4 3/8 3/8]	$g_{2j}(2)$: [2 2 3]
$p_{1j}(3)$: [1/3 1/3 1/3]	$g_{3j}(3)$: [2 2 3]
$p_{2j}(1)$: [1/8 3/8 1/2]	$g_{2j}(1)$: [2 1 2]
$p_{2j}(2)$: [1/2 1/4 1/4]	$g_{2j}(2)$: [1 4 4]
$p_{3j}(1)$: [3/8 1/4 3/8]	$g_{3j}(1)$: [2 3 3]
$p_{3j}(2)$: [1/8 1/4 5/8]	$g_{3j}(2)$: [3 2 2]

Table 2.1

Use the policy iteration method to determine the policy which gives the maximum gain g.
A computational procedure utilizing Matlab We first put the data in an m-file. Since we have several cases, it is expedient to include the case number. This example belongs to type 2. Data in file md61.m

```
    type = 2
PA = [1/3 1/3 1/3; 1/4 3/8 3/8; 1/3 1/3 1/3; 0 0 0;
     1/8 3/8 1/2; 1/2 1/4 1/4; 0 0 0;
     3/8 1/4 3/8; 1/8 1/4 5/8]
GA = [1 3 4; 2 2 3; 2 2 3; 0 0 0;        % Zero rows are separators
```

```
     2 1 2; 1 4 4; 0 0 0;
     2 3 3; 3 2 2]
A = [1 2 3 0 1 2 0 1 2]


    md61
type = 2
PA = 0.3333    0.3333    0.3333
    0.2500    0.3750    0.3750
    0.3333    0.3333    0.3333
         0         0         0
    0.1250    0.3750    0.5000
    0.5000    0.2500    0.2500
         0         0         0
    0.3750    0.2500    0.3750
    0.1250    0.2500    0.6250

GA =  1     3     4
     2     2     3
     2     2     3
     0     0     0
     2     1     2
     1     4     4
     0     0     0
     2     3     3
     3     2     2

A =   1     2     3     0     1     2     0     1     2
```

Once the data are entered into Matlab by the call for file “md61,” we make preparation for the policy-improvement step. The procedure is in the file newpolprep.m

```
    % file newpolprep.m
% version of 3/23/92
disp('Data needed:')
disp('  Matrix PA of transition probabilities for states and actions')
disp('  Matrix GA of gains for states and actions')
disp('  Type number to identify GA matrix types')
disp('    Type 1.  Gains associated with a state')
disp('    Type 2.  One-step transition gains')
disp('    Type 3.  Gains associated with a demand')
disp('  Row vector of actions')
disp('  Value of alpha (= 1 for no discounting)')
c  = input('Enter type number to show gain type ');
a  = input('Enter value of alpha (= 1 for no discounting) ');
PA = input('Enter matrix PA of transition probabilities ');
GA = input('Enter matrix GA of gains ');
if c == 3
 PD = input('Enter row vector of demand probabilities ');
end
if c == 1
 QA = GA';
elseif c ==2
```

```
 QA = diag(PA*GA');     % (qx1)
else
 QA = GA*PD';
end

    A  = input('Enter row vector A of possible actions ');  % (1xq)
m  = length(PA(1,:));
q  = length(A);
n  = input('Enter n, the power of P  to approximate P0 ');
s  = input('Enter s, the power of P  in the approximation of V ');
QD = [(1:q)' A' QA];    % First col is index; second is A; third is QA
DIS = ['    Index      Action    Value'];
disp(DIS)
disp(QD)
if a == 1
 call = 'Call for newpol';
else
 call = 'Call for newpold';
end
disp(call)

    newpolprep   % Call for preparatory program in file npolprep.m
Data needed:
 Matrix PA of transition probabilities for states and actions
 Matrix GA of gains for states and actions
 Type number to identify GA matrix types
   Type 1.  Gains associated with a state
   Type 2.  One-step transition gains
   Type 3.  Gains associated with a demand
 Row vector of actions
 Value of alpha (= 1 for no discounting)

Enter type number to show gain type 2
Enter value of alpha (=1 for no discounting) 1
Enter matrix PA of transition probabilities  PA
Enter matrix GA of gains GA
Enter row vector A of possible actions  A
Enter n, the power of P  to approximate P0  16
Enter s, the power of P  in the approximation of V  6
   Index      Action    Value
   1.0000    1.0000    2.6667
   2.0000    2.0000    2.3750
   3.0000    3.0000    2.3333
   4.0000         0         0
   5.0000    1.0000    1.6250
   6.0000    2.0000    2.5000
   7.0000         0         0
   8.0000    1.0000    2.6250
   9.0000    2.0000    2.1250

Call for newpol
```

The procedure has prompted for the procedure newpol (in file newpol.m)

```
    % file:  newpol.m  (without discounting)
% version of 3/23/92
d  = input('Enter policy as row matrix of indices ');
D  = A(d);                % policy in terms of actions
P  = PA(d',:);            % selects probabilities for policy
Q  = QA(d',:);            % selects next-period gains for policy
P0 = P^n;                 % Display to check convergence
PI = P0(1,:);             % Long-run distribution
G = PI*Q                  % Long-run expected gain per period
C = P + eye(P);
for j=2:s
 C = C + P^j;            % C = I + P + P^2 + ... + P^s
end
V = (C - (s+1)*P0 )*Q;  % B = B0*Q
disp(' ')
disp('Approximation to P0; rows are long-run dbn')
disp(P0)
disp('Policy in terms of indices')
disp(d)
disp('Policy in terms of actions')
disp(D)
disp('Values for the policy selected')
disp(V)
disp('Long-run expected gain per period G')
disp(G)
T = [(1:q)' A' (QA + PA*V)];  % Test values for determining new policy
DIS =['    Index     Action   Test Value'];
disp(DIS)
disp(T)
disp('Check current policy against new test values.')
disp('--If new policy needed, call for newpol')
disp('--If not, use policy, values V, and gain G, above')
```

```
    newpol
Enter policy as row matrix of indices [2 6 9]  % A deliberately poor choice

Approximation to P0; rows are long-run dbn
   0.2642    0.2830    0.4528
   0.2642    0.2830    0.4528
   0.2642    0.2830    0.4528

Policy in terms of indices
    2     6     9

Policy in terms of actions
    2     2     2

    Long-run expected gain per period G
   2.2972
   Index     Action   Test Value
   1.0000    1.0000    2.7171
   2.0000    2.0000    2.4168
   3.0000    3.0000    2.3838
   4.0000         0         0
   5.0000    1.0000    1.6220
   6.0000    2.0000    2.5677
   7.0000         0         0
   8.0000    1.0000    2.6479
   9.0000    2.0000    2.0583

Check current policy against new test values.
--If new policy needed, call for newpol
--If not, use policy and gain G, above      % New policy is needed
newpol

Enter policy as row matrix of indices [1 6 8]

Approximation to P0; rows are long-run dbn
   0.3939    0.2828    0.3232
   0.3939    0.2828    0.3232
   0.3939    0.2828    0.3232

Policy in terms of indices
    1     6     8

Policy in terms of actions
    1     2     1

Values for selected policy
   0.0526
  -0.0989
   0.0223

Long-run expected gain per period G
   2.6061

   Index     Action   Test Value
   1.0000    1.0000    2.6587
   2.0000    2.0000    2.3595
   3.0000    3.0000    2.3254
   4.0000         0         0
   5.0000    1.0000    1.6057
   6.0000    2.0000    2.5072
   7.0000         0         0
   8.0000    1.0000    2.6284
   9.0000    2.0000    2.1208

Check current policy against new test values.
--If new policy needed, call for newpol
--If not, use policy, values V, and gain G, above
```

Since the policy selected on this iteration is the same as the previous one, the procedure has converged to an optimal policy. The first of the first three rows is maximum; the second of the next two rows is maximum; and the first of the final two rows is maximum. Thus, we have selected rows 1, 5, 6, corresponding to the optimal policy $d^* \sim (1\ 2\ 1)$. The expected long-run gain per period $g = 2.6061$.
The value matrix is

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0.0527 \\ -0.0989 \\ 0.0223 \end{bmatrix} \tag{2.54}$$

We made a somewhat arbitrary choice of the powers of $\mathbf{P}$ used in the approximation of P_0 and B_0. As we note in the development of the approximation procedures in Sec 4, the convergence of $\mathbf{P}^n$ is governed by the magnitude of the largest eigenvalue other than one. We can always get a check on the reliability of the calculations by checking the eigenvalues for $\mathbf{P}$ corresponding to the presumed optimal policy. For the choice above, we find the eigenvalues to be 1, -0.1250, 0.0833. Since $0.125^4 \approx 0.0002$, the choices of exponents should be quite satisfactory. In fact, we could probably use $\mathbf{P}^8$ as a satisfactory approximation to P_0, if that were desirable. The margin allows for the possibility that for some policies the eigenvalues may not be so small. — □

7. Policy iteration– with discounting
It turns out that the policy iteration procedure is simpler in the case of discounting, as suggested by the evolution analysis in Sec 4. We have the following two-phase procedure, based on that analysis.
   a. **Value-determination procedure**. Given the q_i and transition probabilities $p(i,j)$ for the current policy, solve $\mathbf{v} = \mathbf{q} + \alpha\mathbf{P}\mathbf{v}$ to get for the corresponding v_i

$$\mathbf{v} = [\mathbf{I} - \alpha\mathbf{P}]^{-1}\mathbf{q} \tag{2.55}$$

   b. **Policy-improvement procedure** Given $\{v_i : 1 \le i \le M\}$ for the current policy, determine a new policy d^*, with each $d^*(i) = a_i*$ determined as that action for which

$$q_i^* + \alpha \sum_{j=1}^{M} p^*(i,j)\, v_j = \max_k \{q_i(a_{ik}) + \sum_{j=1}^{M} p_{ij}(a_{ik})\, v_j a_{ik} \in A_i\} \tag{2.56}$$

We illustrate the Matlab procedure with the same numerical example as in the previous case, using discount factor $a = 0.8$. The data file is the same, so that we call for it, as before.

Example 2.10

Assume data in file md61.m is in Matlab; if not, call for md61. We use the procedure newpolprep to prepare for the iterative procedure by making some initial choices.

```
    newpolprep
Data needed:
 Matrix PA of transition probabilities for states and actions
 Matrix GA of gains for states and actions
 Type number to identify GA matrix types
   Type 1.  Gains associated with a state
   Type 2.  One-step transition gains
   Type 3.  Gains associated with a demand
 Row vector of actions
 Value of alpha (= 1 for no discounting)
```

```
    Enter type number to show gain type 2
Enter value of alpha (= 1 for no discounting) 0.8
Enter matrix PA of transition probabilities PA
Enter matrix GA of gains GA
Enter row vector A of possible actions A
Enter n, the power of P  to approximate P0 16
Enter s, the power of P  in the approximation of V 6

   Index      Action  Test Value
   1.0000    1.0000    2.6667
   2.0000    2.0000    2.3750
   3.0000    3.0000    2.3333
   4.0000         0         0
   5.0000    1.0000    1.6250
   6.0000    2.0000    2.5000
   7.0000         0         0
   8.0000    1.0000    2.6250
   9.0000    2.0000    2.1250

Call for newpold
```

The procedure for policy iteration is in the file newpold.m.

```
    % file newpold.m  (with discounting)
% version of 3/23/92
d = input('Enter policy as row matrix of indices ');
D = A(d);
P = PA(d',:);        % transition matrix for policy selected
Q = QA(d',:);        % average next-period gain for policy selected

V = (eye(P) - a*P)\Q;            % Present values for unlimited future
T = [(1:q)' A' (QA + a*PA*V)];   % Test values for determining new policy
disp(' ')
disp('Approximation to P0; rows are long-run dbn')
disp(P0)
disp('    Policy in terms of indices')
disp(d)
disp('    Policy in terms of actions')
disp(D)
disp('  Values for policy')
disp(V)
DIS =['    Index     Action    Test Value'];
disp(DIS)
disp(T)
disp('Check current policy against new test values.')
disp('--If new policy needed, call for newpold')
disp('--If not, use policy and values above')


    newpold
Enter policy as row matrix of indices [3 5 9]  % Deliberately poor choice

Approximation to P0; rows are long-run dbn
```

```
   0.3939    0.2828    0.3232
   0.3939    0.2828    0.3232
   0.3939    0.2828    0.3232

   Policy in terms of indices
    3     5     9

   Policy in terms of actions
    3     1     2

 Values for policy
  10.3778
   9.6167
  10.1722

   Index     Action  Test Value
   1.0000    1.0000   10.7111
   2.0000    2.0000   10.3872
   3.0000    3.0000   10.3778
   4.0000         0         0
   5.0000    1.0000    9.6167
   6.0000    2.0000   10.6089
   7.0000         0         0
   8.0000    1.0000   10.7133
   9.0000    2.0000   10.1722

Check current policy against new test values.
--If new policy needed, call for newpold
--If not, use policy and values above

newpold
Enter policy as row matrix of indices [1 6 8]

Approximation to P0; rows are long-run dbn
   0.3939    0.2828    0.3232
   0.3939    0.2828    0.3232
   0.3939    0.2828    0.3232

   Policy in terms of indices
    1     6     8

   Policy in terms of actions
    1     2     1

 Values for policy
  13.0844
  12.9302
  13.0519

       Index     Action  Test Value
   1.0000    1.0000   13.0844
   2.0000    2.0000   12.7865
```

```
    3.0000    3.0000   12.7511
    4.0000         0         0
    5.0000    1.0000   12.0333
    6.0000    2.0000   12.9302
    7.0000         0         0
    8.0000    1.0000   13.0519
    9.0000    2.0000   12.5455

Check current policy against new test values.
--If new policy needed, call for newpold
--If not, use policy and values above
```

Since the policy indicated is the same as the previous policy, we know this is an optimal policy. Rows 1, 6, 8, indicate the optimal policy to be $d^* \sim (1, 2, 1)$. It turns out in this example that the optimal policies are the same for the discounted and undiscounted cases. That is not always true. The $\mathbf{v}$ matrix gives the present values for unlimited futures, for each of the three possible starting states.

Chapter 3

Queues with Poisson Arrivals, Exponential Servers[1]

A standard model of a queueing system with a single waiting line and one or more servers assumes that "customers" arrive according to a Poisson process with rate (λ). The customer at the head of the line goes to the first available server, if there are more than one, or to the single server as soon as available, if there is only one. The servers operate independently (of each other and the arrival process), each with exponential service time. We suppose each server has the same distribution, exponential (μ). Such a system may be analyzed as a Markov birth-death process. An analysis of the long-run probabilities and expectations of various quantities after the system has settled down to equilibrium yields the results below.

Calculation of these quantities is straightforward, but somewhat tedious if various cases are considered. Matlab procedures for single-server and two-server systems are utilized to make these calculations quickly and to present them in a useful way.

Notation

- N_t = number in system (in service and waiting) at time t
- Q_t = number waiting to be served at time t
- $\pi_j = \lim\limits_{t\to\infty} p_i j\,(t)$ = long-run probability of being in state j
- W_t = waiting time for service for customer who arrives at time t
- D_t = waiting time plus service time for customer who arrives at time t

- A = random variable with distribution of interarrival times
- S = random variable with distribution of service times

Long-run probabilities $\pi_j = P(N_t = j)$ for large t, s servers, $E[A] = 1/\lambda, E[S] = 1/\mu$

For $s = 1$,

- $\rho = E[S]/E[A] = \lambda/\mu$
- $\pi_0 = 1 - \rho\pi_n = (1-\rho)\,\rho^n$
- N_t is approximately geometric $(1-\rho)$

For $s > 1$,

- $\rho = E[S]/sE[A] = \lambda/s\mu$
- $\pi_n = \{ \begin{array}{ll} \pi_0\,(s\rho)^n/n! = \pi_0\,(\lambda/\mu)^n/n! & 0 \le n \le s \\ \pi_0\,(s^s/s!)\,\rho^n = \pi_0\,[(s\rho)^s/s!]\,\rho^{n-s} & s < n \end{array}$

[1]This content is available online at <http://cnx.org/content/m31072/1.3/>.

For $s = 2$

- $\pi_0 = \frac{1-\rho}{1+\rho} = \frac{2\mu-\lambda}{2\mu+\lambda}$

For $s = 3$

- $\pi_0 = \frac{1-\rho}{1+2\rho+\frac{3}{2}\rho^2}$

For $s = 4$

- $\pi_0 = \frac{1-\rho}{1+3\rho+8\rho^2+\frac{8}{3}\rho^3}$

For large t, with the system in equilibrium

$$E[D_t] = E[A]\,E[N_t] \text{ and } E[W_t] = E[A]\,E[Q_t] \approx E[S]\,E[N_t] \tag{3.1}$$

For $s = 1$

- $E[N_t] = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$
- $E[Q_t] = \rho E[N_t]\,P(N_t > 0) = \rho$
- $E[W_t] = E[S]\,E[N_t] = \frac{\lambda/\mu}{\mu-\lambda}$
- D_t is approximately exponential $(\mu-\lambda)$

For $s > 1$

- $C = P(W_t > 0) = \pi_0 \frac{(s\rho)^s}{s!(1-\rho)} = E[Q_t]\,\frac{1-\rho}{\rho} = s\mu(1-\rho)\,E[W_t]$
- $P(W_t > v) = Ce^{-(\mu s-\lambda)v} v \geq 0$
- $P(D_t > v) = e^{-\mu v}\left[1 + C\mu\frac{1-e^{-[\mu(s-1)-\lambda]v}}{\mu(s-1)-\lambda}\right]$ for $\lambda \neq \mu(s-1)$
- $P(D_t > v) = e^{-\mu v}[1+\mu C v]$ for $\lambda = \mu(s-1)$
- $E[Q_t] = \pi_0\frac{(s\rho)^s}{s!}\frac{\rho}{(1-\rho)^2}$
- $E[N_t] \approx E[Q_t] + \frac{\lambda}{\mu} = E[Q_t] + s\rho$

3.1 Matlab calculations for single server queue (in file queue1.m)

```
    L = input('Enter lambda  ');    % Type desired value, no extra space
M = input('Enter mu  ');        % Type desired value, no extra space
a = ['    lambda      mu'];
b = [L M];
disp(a)
disp(b)

r = L/M;                        % Rho

EN = r/(1 - r);                 % E[N]

EQ = r*EN;                      % E[Q]

EW = EQ/L;                      % E[W]

ED = EN/L;                      % E[D]
```

```
A = ['      rho         EN         EQ         EW         ED'];      % Identifies entries in B
B = [r EN EQ EW ED];
disp(A)
disp(B)

v = input('Enter row matrix of values v  ');   % Type matrix of desired values

PD = exp(-M*(1 - r)*v);                       % Calculates P(Dt > v)

S = ['       v      P(D>v)'];
s = [v; PD]';
disp(S)
disp(s)
```

Example 3.1

```
    queue1

Enter lambda  0.1
Enter mu  0.2
   lambda      mu
   0.1000    0.2000

     rho       EN        EQ        EW        ED
   0.5000    1.0000    0.5000    5.0000   10.0000


Enter row matrix of values v  [8 16 24]
      v      P(D>v)
   8.0000    0.4493
  16.0000    0.2019
  24.0000    0.0907
```

3.2 Matlab calculations for two-server queue (in file queue2.m)

Note that the procedure will not calculate $P(D > v)$ if $\lambda = \mu$.

```
    L = input('Enter lambda  ');    % Type desired value, no extra space
M = input('Enter mu  ');        % Type desired value, no extra space
a = ['    lambda      mu'];
b = [L M];
disp(a)
disp(b)

r = L/(2*M);
EQ = (2*r^3)/(1 - r^2);
EN = EQ + 2*r;
EW = EQ/L;
ED = EN/L;
```

```
A = [' rho EN EQ EW ED'];      % Identifies entries in B
B = [r EN EQ EW ED];
disp(A)
disp(B)

v = input('Enter row matrix of values v  ');

t = 2*M*EW*(1 - r)/(1 - 2*r);
PD2 = exp(-M*v).*(1 + t.*(1 - exp(-M*v + L*v)));  % Calculates P(D > v) for L not equal M

S = ['       v      P(D>v)'];
s = [v; PD2]';
disp(S)
disp(s)
```

Example 3.2

```
    queue2

Enter lambda  0.1
Enter mu  0.2
   lambda      mu
   0.1000    0.2000

     rho        EN        EQ        EW        ED
   0.2500    0.5333    0.0333    0.3333    5.3333

Enter row matrix of values v  [4 8 16]
      v      P(D>v)
   4.0000    0.4790
   8.0000    0.2241
  16.0000    0.0473
```

3.3 Comparison of single-server and two-server queues

A queueing system has Poisson arrivals, rate λ and exponential (μ) service times.

a. In system one, there is one server, with expected service time $1/\mu = 1$ minute. Determine

$$[E[N], E[Q], E[W], E[D], \text{ and } P(D > v), v = 1, 3, 5, 10 (3.2)$$

for expected arrival rates $\lambda = 0.6, 0.9, 0.99$ customers per minute.

b. In system two there are two servers, each with expected service time $1/\mu = 2$ minutes. Calculate the same quantities as for system one and compare the results for the two systems.

```
    queue1
Enter lambda  0.6
Enter mu  1
    lambda      mu
```

```
   0.6000    1.0000

     rho        EN        EQ        EW        ED
   0.6000    1.5000    0.9000    1.5000    2.5000

Enter row matrix of values v  [1 3 5 10]
      v      P(D>v)
   1.0000    0.6703
   3.0000    0.3012
   5.0000    0.1353
  10.0000    0.0183
Ov = ones(1,length(v));
R = r*Ov;                    % Row vector with all terms = r
r1 = R;
E11 = B;
v11 = PD;

queue1
Enter lambda  0.9
Enter mu  1
    lambda      mu
   0.9000    1.0000

     rho        EN        EQ        EW        ED
   0.9000    9.0000    8.1000    9.0000   10.0000

Enter row matrix of values v  v  % Calls for previously entered v
      v      P(D>v)
   1.0000    0.9048
   3.0000    0.7408
   5.0000    0.6065
  10.0000    0.3679
R = r*Ov;
r2 = R;
E12 = B;
v12 = PD;


queue1
Enter lambda  0.99
Enter mu  1
   lambda      mu
   0.9900    1.0000

     rho        EN        EQ        EW        ED
   0.9900   99.0000   98.0100   99.0000  100.0000

Enter row matrix of values v  v
      v      P(D>v)
   1.0000    0.9900
   3.0000    0.9704
   5.0000    0.9512
  10.0000    0.9048
R = r*Ov;
r3 = R;
E13 = B;
v13 = PD;

queue2                 % Begin calculations for second system
Enter lambda  0.6
Enter mu   0.5
   lambda      mu
   0.6000    0.5000

     rho        EN        EQ        EW        ED
   0.6000    1.8750    0.6750    1.1250    3.1250

Enter row matrix of values v  v
      v      P(D>v)
   1.0000    0.7501
   3.0000    0.3988
   5.0000    0.2019
  10.0000    0.0328
E21 = B;               % Not necessary to determne r1, r2, r3, since
v21 = PD2;             % they are the same as for system one.

queue2
Enter lambda  0.9
Enter mu  0.5
   lambda      mu
   0.9000    0.5000

     rho        EN        EQ        EW        ED
   0.9000    9.4737    7.6737    8.5263   10.5263

Enter row matrix of values v   v
      v      P(D>v)
   1.0000    0.9245
   3.0000    0.7749
   5.0000    0.6410
  10.0000    0.3916

E22 = B;
v22 = PD2;

queue2
Enter lambda  0.99
Enter mu  0.5
   lambda      mu
   0.9900    0.5000

     rho        EN        EQ        EW        ED
   0.9900   99.4975   97.5175   98.5025  100.5025

 Enter row matrix of values v   v
      v      P(D>v)
   1.0000    0.9920
   3.0000    0.9743
   5.0000    0.9557
  10.0000    0.9094
E23 = B;
v23 = PD2;
C = [E11; E21; zeros(E11); E12; E22; zeros(E11); E13; E23];  % Zeros are spacers

disp(A)
     rho        EN        EQ        EW        ED
disp(C)
   0.6000    1.5000    0.9000    1.5000    2.5000
   0.6000    1.8750    0.6750    1.1250    3.1250
        0         0         0         0         0
   0.9000    9.0000    8.1000    9.0000   10.0000
   0.9000    9.4737    7.6737    8.5263   10.5263
        0         0         0         0         0
   0.9900   99.0000   98.0100   99.0000  100.0000
   0.9900   99.4975   97.5175   98.5025  100.5025

H = ['      rho        v      P(D1>v)   P(D2>v)'];
PDV = [r1 r2 r3; v v v; v11 v12 v13; v21 v22 v23]';

disp(H)
     rho        v      P(D1>v)   P(D2>v)
disp(PDV)
   1.0000    1.0000    0.6703    0.7501
   1.0000    3.0000    0.3012    0.3988
   1.0000    5.0000    0.1353    0.2019
   1.0000   10.0000    0.0183    0.0328
   0.9000    1.0000    0.9048    0.9245
   0.9000    3.0000    0.7408    0.7749
   0.9000    5.0000    0.6065    0.6410
   0.9000   10.0000    0.3679    0.3916
   0.9900    1.0000    0.9900    0.9920
   0.9900    3.0000    0.9704    0.9743
   0.9900    5.0000    0.9512    0.9557
   0.9900   10.0000    0.9048    0.9094
```

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Topics in Applied Probability
This collection includes some m-files for problems supplementary to Pfeiffer: Applied Probability. In addition, the text file collection New Prob mfiles contains both the user defined programs for that work and a collection of mfiles for specific problems with properly formatted data which can be entered into the workspace by calling the appropriate file. These m-files come from a variety of sources (e.g., exams or problem sets, hence the odd names) and may be useful for examples and exercises.