CPP1002 Coding Assignment 5

馬可夫鏈-存貨問題

電子檔 2011/05/04 09:00 前上傳至 moodle
書面檔 2011/05/04 09:15 前於課堂繳交

問題描述

在日常生活中，我們或許不知道今天下雨（或晴天）的機率，但卻可能會知道已知今天下雨（或晴天）的情況下，明天會下雨（或晴天）的機率。以表一為例，假設今天下雨，則明天下雨的機率=P(明雨|今雨)=0.4；反之，如果今天晴天，則明天晴天的機率=P(明晴|今晴)=0.8。

明天

	t \ t+1	晴天	雨天
今天	晴天	0.8	0.2
	雨天	0.6	0.4

表一：相鄰兩天之晴雨天組合機率

如上例，將每期所探討的現象分成數種可能的狀態（state，以表一為例，共有「晴天」、「雨天」兩種狀態），給定所有狀態在前後相鄰兩期間的「轉移機率」（transition probability，譬如表一的 P(明晴|今晴)、P(明雨|今晴)、P(明晴|今雨)、P(明雨|今雨)四種機率）之後，假設第 t 期各狀態的機率僅以「轉移機率」與第 t-1 期各狀態的機率相關，而與第 t-2 期及其以前各狀態的機率無關的話(亦即今天的天氣狀態如何，僅與昨天的天氣狀態直接相關)，若已知最初的狀態機率(譬如第一天為晴天的機率為 0.3)，則我們可以推算出之後連續數天的天氣變化情況之機率。舉例來說，假如今天有70%機率會下雨，則今天開始（包括今天）接連四天之天氣狀態為「雨雨晴雨」的機率如下：

P(雨雨晴雨)=P(今雨)* P(明雨|今雨)* P(後晴|明雨)* P(大後雨|後晴)=0.7*0.4*0.6*0.2=3.36%。

以上即為「作業研究」(Operations Research，OR)理論中著名的「馬可夫鏈」(Markov Chain)[1]，其應用著重於面對一個不確定的未來事件，可依其初始狀態機率及相鄰兩期間之轉移機率，來推算出連續幾期的狀態組合之發生機率。馬可夫鏈之應用很廣，除了可用來分析連續幾天之天氣變化機率外，也可以用在推導產業的連續多期之存貨變化機率，本次作業便是利用馬可夫鏈針對在不確定的環境下採取既定的存貨策略，分析該策略對存貨數量產生的變化情形。

假設現有一販賣相機的商家採取(s,Q)補貨策略，亦即當他保有的相機存貨數量低於某一設定之安全存量門檻值 s 時，必須立即向其上游補訂 Q 台（譬如 s=2，Q=3 代表存貨若為 0 或 1 台時，必須立即向其上游訂購 3 台）。由於不同 s,Q 值的補貨策略會導致不同的結果，該商家想要知道在其所訂定的某種(s,Q)補貨策略下，經過數期後其存貨狀況為何。欲求解此問題前，我們應先思考存貨的各種變動情形及其對應之發生機率，以製作成「轉移矩陣」(Transition Matrix)。

[1] 更多有關馬可夫鏈的介紹，可自行於網際網路上搜尋，或閱讀以下兩網頁：http://function1122.blogspot.com/2010/08/markov-chains.html 及 http://episte.math.ntu.edu.tw/articles/mm/mm_09_3_08/index.html

假設已知每週的各種需求大小及其發生機率，以及店家的(s,Q)補貨策略皆為已知，舉例如下

需求大小及其發生機率：例如：需求只有四種可能的需求大小(D=0,1,2, or 3)且其機率如下：

P(D = 0) = 0.2　　表示每週需求為 0 的機率是 0.2

P(D = 1) = 0.4　　表示每週需求為 1 的機率是 0.4

P(D = 2) = 0.3　　表示每週需求為 2 的機率是 0.3

P(D = 3) = 0.1　　表示每週需求為 2 的機率是 0.1

本題將假設 D 為整數，不過其範圍不見得從 0 開始，且亦不見得連續，譬如若讀取之 D 為 5,3,2,5,9 時，可依其最大之需求值(9)來設立一個大小為 9+1=10 之需求機率陣列 P，以存取各需求大小之出現機率，譬如 P[2]=P[3]=P[9]=0.2, P[5]=0.4, 而其它之 P[0]=P[1]=P[4]=P[6]=P[7]=P[8]=0。

(s,Q)補貨策略：以(s,Q)=(2,2)為例

當期期末存貨小於安全存量 2(不包含 2)時，隨即在下期期初訂購 2 台相機(並假設立即到貨)

轉移矩陣 T：其所有狀態個數共有 s+Q 個，以上述為例，s+Q=4，因此 T 為一 4×4 矩陣

根據上述假設，以下舉例解釋轉移矩陣 T 之(1,0)、(1,2)、(0,0)、(2,3)項數值 (存貨狀態的時間點皆為期末)：

例:(1,0) ➔當期存貨為 1，下期存貨為 0

1. 此時存貨=1，因為小於安全存量 2 所以要訂購 2 台相機，下期期初將共有 1+2 = 3 台相機
2. 下期期初時有 3 台相機，而期末存貨卻變成 0 台，可見下期之顧客需求總量為 3-0=3 台，其發生機率 P(D = 3) = 0.1。因此，本期期末存貨為 1，下期期末存貨為 0 的機率為 0.1

例: (1,2) ➔當期存貨為 1，下期存貨為 2

1. 此時存貨=1，因為小於安全存量 2 所以要訂購 2 台相機，下期期初將共有 1+2=3 台相機
2. 下期期初時有 3 台相機，而期末存貨卻變成 2 台，可見下期之顧客需求總量為 3-2=1 台，其發生機率 P(D = 1) = 0.4。因此，本期期末存貨為 1，下期期末存貨為 2 的機率為 0.4

例: (0,0) ➔當期存貨為 0，下期存貨為 0

1. 此時存貨=0，因小於安全存量 2 所以要訂購 2 台相機，下期期初將共有 0+2=2 台相機
2. 下期期初時有 2 台相機，而期末存貨卻變成 0 台，可見下期之顧客需求總量可能為 2 或 3 台，(其中需求為 3 時會造成缺貨，由於本題不探討缺貨的額外損失，因此會將「缺貨」視為與「零存貨」有相同效果) 其發生機率為 P(D = 2) + P(D = 3) = 0.3 + 0.1 = 0.4。因此，本期期末存貨為 0，下期期末存貨為 0 的機率為 0.4

例: (2,3) ➔當期存貨為 2，下期存貨為 3

1. 此時存貨=2，因為未小於安全存量 2 所以不訂購相機，下期期初將共有 0+2=2 台相機
2. 下期期初時有 2 台相機，而期末存貨卻變成 3 台，這是不可能的(因為經歷一期的存貨頂多維持不變或變少，不可能無中生有而增加)，因此其發生機率為 0

轉移矩陣 T 之任一(i,j)項可依類似方式及步驟一一推導而得。亦即，首先要讀入給定的資料檔內含的各期需求，整理所有可能的需求大小並統計出不同的需求大小所發生之機率(請注意需求大小不見得一定從 0 開始，且其值不見得連續)，接著依資料檔內含的 s 及 Q 值來設定 T 矩陣，並判斷造成 T 矩陣第(i,j)項的各種需求可能性，計算其機率。

以上述四種需求大小(0,1,2,3)及其機率，可依照上述步驟計算出轉移矩陣 T 如下:

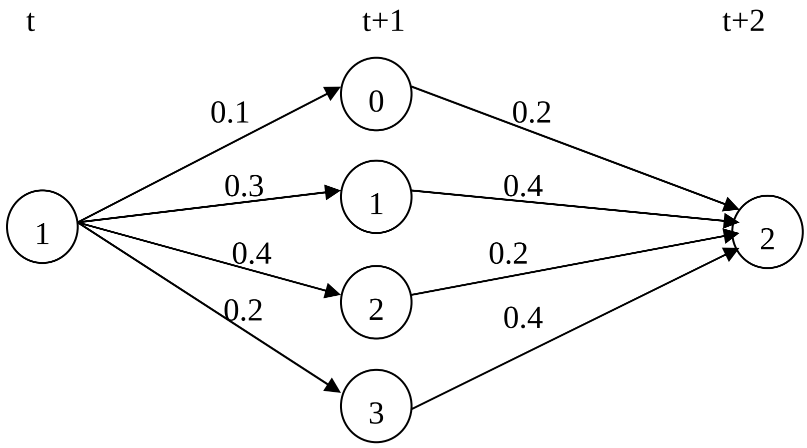
t \ t+1	0	1	2	3
0	0.4	0.4	0.2	0
1	0.1	0.3	0.4	0.2
2	0.4	0.4	0.2	0
3	0.1	0.3	0.4	0.2

轉移矩陣 T

$$\begin{bmatrix} 0.4 & 0.4 & 0.2 & 0 \\ 0.1 & 0.3 & 0.4 & 0.2 \\ 0.4 & 0.4 & 0.2 & 0 \\ 0.1 & 0.3 & 0.4 & 0.2 \end{bmatrix}$$

以轉移矩陣 T 計算跨 k+1 期之轉移機率:

以上方法除了可以知道下期(t + 1)的存貨狀況外，還可以知道下下期(t + 2)的存貨狀況，例如本期存貨為 1 而下下期存貨為 2 的機率之計算方式，可用圖一來說明：



圖一：已知期初及期末狀態之連續三期狀態變化可能性

因此 $P(D_{t+2}=2 \mid D_t=1)=0.1*0.2+0.3*0.4+0.4*0.2+0.2*0.4=0.3$。再仔細觀察，我們可以發現上述的計算剛好為轉移矩陣的 row2(當期存貨為 1)乘以 colum3(下下期存貨為 2)所得之值。因此，若要得知 t→t+2 的各種存貨機率表，可將 t→t+1 與 t+1→t+2 的轉移矩陣相乘而得。

$$\underset{t \,\backslash\, t+1}{\begin{bmatrix} 0.4 & 0.4 & 0.2 & 0 \\ \boxed{0.1 \quad 0.3 \quad 0.4 \quad 0.2} \\ 0.4 & 0.4 & 0.2 & 0 \\ 0.1 & 0.3 & 0.4 & 0.2 \end{bmatrix}} * \underset{t+1 \,\backslash\, t+2}{\begin{bmatrix} 0.4 & 0.4 & 0.2 & 0 \\ 0.1 & 0.3 & 0.4 & 0.2 \\ 0.4 & 0.4 & 0.2 & 0 \\ 0.1 & 0.3 & 0.4 & 0.2 \end{bmatrix}} = \underset{t \,\backslash\, t+2}{\begin{bmatrix} 0.28 & 0.36 & 0.28 & 0.08 \\ 0.25 & 0.35 & \boxed{0.3} & 0.1 \\ 0.28 & 0.36 & 0.28 & 0.08 \\ 0.25 & 0.35 & 0.3 & 0.1 \end{bmatrix}}$$

由於任相鄰兩期之轉移矩陣 T 皆相同，所以若要計算 t→t+k 期的連續 k+1 期之各狀態間變換機率的話，可藉由計算 T^k 而得。

程式要求及作法

一、讀檔、設定需求可能數值並計算其機率：

本次作業希望同學由讀入之輸入檔(e.g. n3.txt)去統計出所有曾出現過的需求大小(D)及其機率(P(D=d))。 讀檔技巧與前幾次作業不太相同，必須使用 switch 依每行不同的開頭作不同的處理。請注意讀檔部分(尤其在處理 c 開頭的忽略 comment 行)可能會讓大部分同學在一開始隨即卡住，老師希望同學自己要能變通，譬如你可以先把 c 開頭的部分刪除，先確定其它部分讀入無誤後，再想辦法處理那些 c 開頭的部分。

每行之開頭大致可分成以下 5 類：

case'c': //comment，忽略不讀取

case't': //title，問題的名稱，讀取並印出

case'w': //需求表的總期(週)數，以 w 表示，可由此設一維陣列以存取各期需求

case's': //補貨策略，後面接著 s Q 兩個數值，可由此設$(s+Q)\times(s+Q)$之二維轉移陣列 T

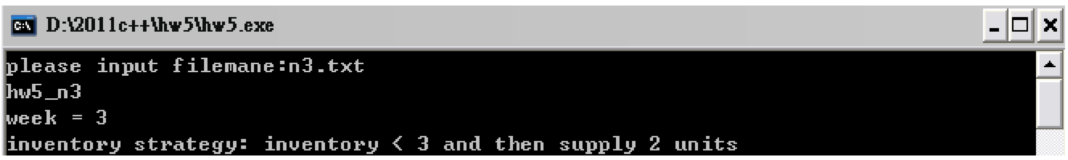
case'd': //每期需求

以 3 週的需求表為例

…………(前面 c 開頭省略)…………

```
t hw5_n3      //問題的名稱，讀取並印出，如圖二
w   3         //歷史需求的總期數，w=3，代表有三期需求 D[0],D[1],D[2]
s   3   2     //補貨策略，安全存量 s=3、訂購量 Q=2，亦即當存貨小於 3 時會訂購 2 台
d   2         //此期需求為 2，即 D[0]=2
d   0         //此期需求為 0，即 D[1]=0
d   1         //此期需求為 1，即 D[2]=1
```

在讀檔後，必須印出如圖二結果:



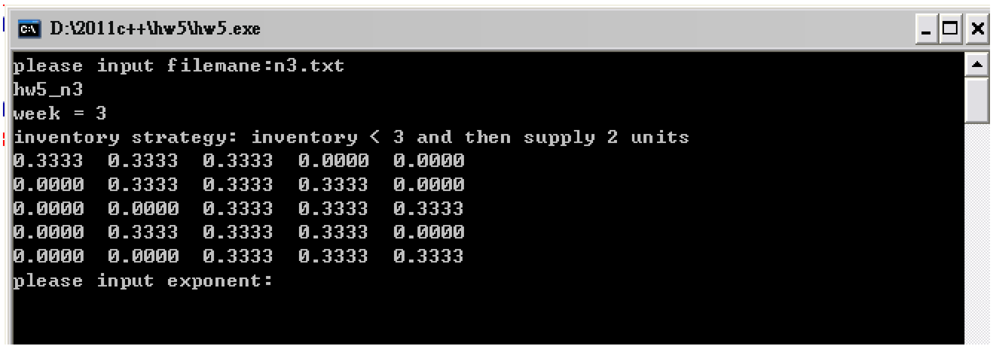
圖二

由於此一歷史資料檔中並未記錄各期曾經出現的最大需求量(最小需求量設為 0，代表沒有需求)，因此一定要先讀完所有需求，才能得知全部的需求大小範圍(假設最大為 D_{max})、次數並進而計算其機率。舉例來說，可用 int D[w]來存取 w 期歷史資料，並使用 double P[D_{max}+1]來計算某期需求量 q 之機率 P[q]。此外，應根據 s+Q 之值設置一個$(s+Q)\times(s+Q)$之二維轉移陣列 T，初始化其值為 0。

以 n3.txt 為例，讀檔後可知 w=3，因此可設定 int D[3]；得知 s=3,Q=2 後可設定一個 5×5 之 int T[5][5]轉移矩陣，並初始化其值為 0。在讀完各期需求後得知 D_{max}=2，因此可設定 double P[3]，並進而計算出 P[0]= P[1]= P[2]=0.3333….

二、計算轉移矩陣 T:

經過上述步驟後，可算出 P[0],⋯, P[D_{max}]，再把計算轉移矩陣 T 的的步驟套入，要特別注意什麼時候該補貨以及究竟該補多少量的貨。以 n3.txt 為例，本步驟計算後的結果如圖三所示:



```
D:\2011c++\hw5\hw5.exe
please input filemane:n3.txt
hw5_n3
week = 3
inventory strategy: inventory < 3 and then supply 2 units
0.3333  0.3333  0.3333  0.0000  0.0000
0.0000  0.3333  0.3333  0.3333  0.0000
0.0000  0.0000  0.3333  0.3333  0.3333
0.0000  0.3333  0.3333  0.3333  0.0000
0.0000  0.0000  0.3333  0.3333  0.3333
please input exponent:
```

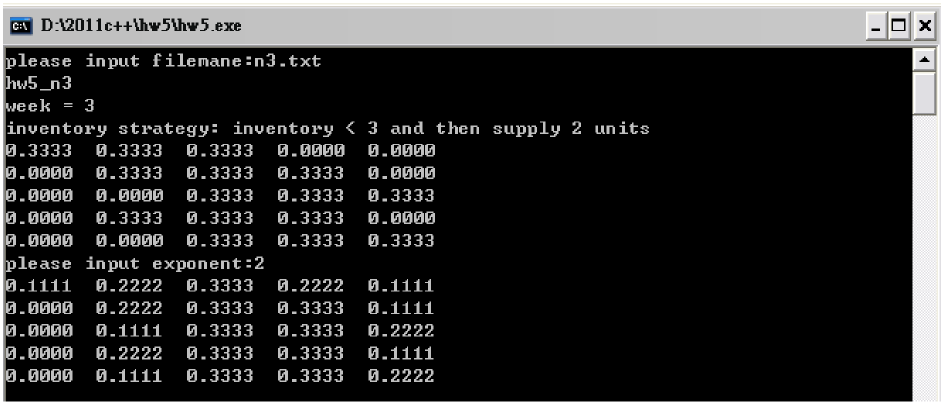
圖三

例: (1,2) ➔ 當期存貨為 1，下期存貨為 2

1. 此時存貨=1，因為小於安全存量 s=3 所以要訂購 Q=2 台相機，下期期初將共有 1+2=3 台相機
2. 下期期初時有 3 台相機，而期末存貨卻變成 2 台，可見下期之顧客需求總量為 3-2=1 台，其發生機率 P[1]=1/3。因此，本期期末存貨為 1，下期期末存貨為 2 的機率為 1/3

三、以轉移矩陣 T 之 k 次方計算跨 k+1 期之轉移機率矩陣:

這個部分必須以一函式 void matrix_power() 來實作。先將原始之 T 複製至另一陣列 Tk，再將 Tk 及 k 當該函式之輸入值，以計算 T^k 之值存入 Tk 矩陣再回傳至主函式。其中，矩陣相乘的次方數(exponent) k 則由使用者輸入，若輸入為 1 則表示為矩陣的 1 次方，輸入 2 則為矩陣的 2 次方⋯以此類推，並印出結果如圖四所示。這部分可參考上次 hw4 將陣列傳入函式的技巧，與 hw4 不同的是，本次作業要傳入 2 維的陣列，這部分請同學自行變通。



```
D:\2011c++\hw5\hw5.exe
please input filemane:n3.txt
hw5_n3
week = 3
inventory strategy: inventory < 3 and then supply 2 units
0.3333  0.3333  0.3333  0.0000  0.0000
0.0000  0.3333  0.3333  0.3333  0.0000
0.0000  0.0000  0.3333  0.3333  0.3333
0.0000  0.3333  0.3333  0.3333  0.0000
0.0000  0.0000  0.3333  0.3333  0.3333
please input exponent:2
0.1111  0.2222  0.3333  0.2222  0.1111
0.0000  0.2222  0.3333  0.3333  0.1111
0.0000  0.1111  0.3333  0.3333  0.2222
0.0000  0.2222  0.3333  0.3333  0.1111
0.0000  0.1111  0.3333  0.3333  0.2222
```

圖四

綜合流程說明與輸出結果

首先，先由使用者輸入檔名，這次作業必須分別測試檔案"n8.txt"、 "n32.txt"以及"n64.txt"，讀檔後，必須印出問題名稱(hw5_n8)、需求表總期數(w)、補貨策略，而後印出轉移矩陣 T，再由使用者輸入 k，並輸出 T^k 的結果。

書面部分必須印出n8.txt,在k=3 和k=6 時的結果。為了使畫面美觀，統一要求**保留小數點以下四位**，方法不拘，數值不要差太多即可。

***作業繳交應注意事項**

1. 作業需要繳交電子檔以及書面(列印程式檔及 n8.txt,在 k=3 和 k=6 的結果)
2. 電子檔請於作業繳交截止時間以前上傳至http://moodle.ncku.edu.tw
   2.1 請同學先建立一個**資料夾**，資料夾名稱為 "**組別_學號_hw5**"，例如組別為第 4 組，學號為 hxxxx，則資料夾名稱則為 **g4_hxxxx_hw5**
   2.2 將程式檔案名稱存為 "**hw5.cpp**"，並將之存於上述設立之**組別_學號_hw5** 資料夾中
   2.3 最後將整個**組別_學號_hw5** 資料夾壓縮成zip檔(**組別_學號_hw5.zip**)，再上傳至moodle
   (！注意！：請勿將cpp檔copy/paste至word檔而上傳之)
3. 書面作業請於 **2011/05/04 上課 5 分鐘內(09:15 前)**繳交至講台，其中需要註明程式是否能被編譯與執行、撰寫人、程式之目的、如何編譯及執行等資料(詳見 http://ilin.iim.ncku.edu.tw/ilin/course/CPP1002/programming.html)。