**33. 马尔可夫预测**

马尔可夫预测，是一种预测事件发生的概率的方法。它是基于马尔可夫链，根据事件的目前状况预测其将来各个时刻（或时期）变动状况的一种预测方法。

马尔可夫预测法的基本要求是状态转移概率矩阵必须具有一定的稳定性。因此，必须具有足够的统计数据，才能保证预测的精度与准确性。换句话说，马尔可夫预测模型必须建立在大量的统计数据的基础之上。

**（一）经典马尔可夫模型**

**一、几个概念**

**状态**：指某一事件在某个时刻（或时期）出现的某种结果；

**状态转移：**事件的发展，从一种状态转变为另一种状态；

**马尔可夫过程：**在事件的发展过程中，若每次状态的转移都仅与前一时刻的状态有关，而与过去的状态无关，或者说状态转移是无后效性的，则这样的状态转移过程就称为马尔可夫过程。

**状态转移概率**：在事件的发展变化过程中，从某一种状态出发，下一时刻转移到其它状态的可能性，称为状态转移概率。由状态转为状态的状态转移概率



**状态转移概率矩阵**：假定某一个事件的发展过程有*n*个可能的状态，即，则矩阵



其中，为从状态转为状态的状态转移概率，称为状态转移概率矩阵。

状态转移矩阵满足：

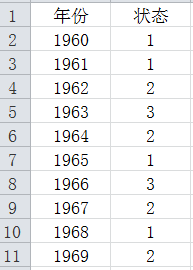
(i) 

(ii) 

**二、状态转移矩阵的计算**

即求出从每个状态转移到其它任何一个状态的状态转移概率，一般采用频率近似概率的思想进行计算。

**例1**某地区农业收成变化的三个状态，即E1“丰收”、E2“平收”和E3“欠收”。下表给出了该地区1960~1999年期间农业收成的状态变化情况（部分）。



计算该地区农业收成变化的状态转移概率矩阵。

datas=xlsread('Agriculture.xlsx');

E=datas(:,2)';

for i=1:3

for j=1:3

f(i,j)=length(findstr([i j],E));

end

end

f %输出状态转移矩阵

fs=sum(f,2);

for i=1:3

P(i,:)=f(i,:)/fs(i);

end

p %输出状态转移概率矩阵

**运行结果：**

f = 3 7 5 %3个E1到E1, 7个E1到E2, 5个E1到E3

7 2 4

4 5 2

P = 0.2000 0.4667 0.3333

0.5385 0.1538 0.3077

0.3636 0.4545 0.1818

**三、状态概率**

用表示事件在第k个时刻（时期）处于状态的概率。

显然，。根据马尔可夫过程的无后效性及Bayes条件概率公式，有



记为第k个时刻（时期）的状态概率向量。由上式可得到计算状态概率向量的递推公式：



其中，为初始状态概率向量。

于是，若事件在某个时刻（时期）的状态已知，则利用状态转移概率矩阵和递推公式，就可以求得它经过k次状态转移后，在第个时刻（时期）处于各种可能的状态的概率，从而就得到该事件在第个时刻（时期）的状态概率预测。

将**例1**中1999年的农业收成状态记为，利用状态转移概率矩阵及递推公式，预测2000—2009年可能出现的各种状态的概率。

S{1}=[0 1 0];

for i=1:10

S{i+1}=S{i}\*P;

end

S{2:end}

运行结果：

ans = 0.5385 0.1538 0.3077

ans = 0.3024 0.4148 0.2828

ans = 0.3867 0.3335 0.2799

ans = 0.3587 0.3590 0.2824

ans = 0.3677 0.3510 0.2813

ans = 0.3648 0.3535 0.2817

ans = 0.3657 0.3527 0.2816

ans = 0.3654 0.3529 0.2816

ans = 0.3655 0.3528 0.2816

ans = 0.3655 0.3529 0.2816

**四、终极状态概率预测**

经过无穷多次状态转移后所得到的状态概率称为终极状态概率。



终极状态概率应满足：，即，即



该齐次方程组的系数行列式为0，有无穷多个解，为得到唯一的正确解，需要另一个限制条件：

此时是n个未知数，n+1个方程，去掉前n个中的任意一个，求解即可得到正确解，即终极状态概率向量。

求**例1**的终极状态概率向量。

n=3; %状态数目

A=P'-eye(n);

A(end,:)=ones(1,n);

%将最后一个方程替换为限制条件sum(pi)=1

b=[zeros(n-1,1);1];

S=inv(A)\*b %解方程组Ax=b得到终极状态概率向量

运行结果：

S = 0.3655 0.3529 0.2816

结果说明，该地区农业收成的变化过程，在无穷多次状态转移后，“丰收”和“平收”状态出现的概率基本相当，都大于“欠收”状态出现的概率。

**例2**设某药品共有1000家购买对象，买A、B、C三药厂的各有400家、300家、300家，即A、B、C三药厂目前的市场占有份额分别为：40%、30%、30%，故初始市场占有状态向量为[0.4, 0.3, 0.3].

为预测A、B、C三个药厂生产的该药品在未来的市场占有情况，收集顾客订货情况如下表：

表 顾客订货情况表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 下季度订货情况 | | | | | 合计 | |
| 来  自 |  | A | B | C | |  |
| A | 160 | 120 | 120 | | 400 |
| B | 180 | 90 | 30 | | 300 |
| C | 180 | 30 | 90 | | 300 |
| 合计 |  | 520 | 240 | 240 | | 1000 |

假设在未来的时期内，订货流向如上表保持不变，即状态转移概率稳定。预测未来3年，A、B、C厂的该药品市场占有率；计算经过若干时期形成稳定之后， A、B、C厂的该药品市场占有率。

f=[160 120 120; 180 90 30; 180 30 90]; %状态转移矩阵

fs=sum(f,2);

for i=1:3

P(i,:)=f(i,:)/fs(i);

end

P %输出状态转移概率矩阵

S0=[0.4 0.3 0.3];

S1=S0\*P

S2=S1\*P

S3=S2\*P

n=3; %状态数目: 3个药厂

A=P'-eye(n);

A(end,:)=ones(1,n); %将最后一个方程替换为限制条件sum(pi)=1

b=[zeros(n-1,1);1];

S=inv(A)\*b %解方程组Ax=b得到终极状态概率向量

运行结果：

P = 0.4000 0.3000 0.3000

0.6000 0.3000 0.1000

0.6000 0.1000 0.3000

S1 = 0.5200 0.2400 0.2400

S2 = 0.4960 0.2520 0.2520

S3 = 0.5008 0.2496 0.2496

S = 0.5000 0.2500 0.2500

**（二）带利润的马尔可夫模型**

经典马尔可夫模型可以应用到，某商品的销售状态的预测。例如，销售状态有畅销和滞销两种，在时间变化过程中，有时呈连续畅销或连续滞销，有时由畅销转为滞销或由滞销转为畅销，每次转移不是盈利就是亏本。

假定连续畅销时盈利r11元，连续滞销时盈利r22元，由畅销转为滞销盈利r12元，由滞销转为畅销盈利r21元，其中，r22和r12为负数，即亏本。

这种随着系统的状态转移，赋予一定利润的马尔可夫模型，称为带利润的马尔可夫模型。

设状态转移概率矩阵为



当系统从状态Ei转到Ej时，赋以利润rij，则系统利润矩阵为：



rij>0表示盈利，rij<0表示亏本，rij= 0表示不亏不盈。

随着时间的变化，系统的状态不断地转移，从而可得到一系列利润。由于状态的转移是随机的，因而一系列的利润是随机变量，其概率关系由马尔可夫链的转移概率决定。

第1期的各利润随机变量记为，其概率分布为：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | … |  |
|  |  | … |  |

, 注意到. 从而第1期的各期望利润为：



第2期的各利润随机变量记为，其概率分布为：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | … |  |
|  |  | … |  |

. 从而第2期的各期望利润为：



依次做下去……，得到第k期的各利润随机变量记为，其概率分布为：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | … |  |
|  |  | … |  |

. 从而第k期的各期望利润为：



当k=1时，规定边界条件。

**例3** 设，求第1至5期的期望利润。

P=[0.5 0.5; 0.4 0.6]; %状态转移矩阵

R=[9 3; 3 -7]; %利润矩阵

T=5; %预测5期的期望利润

n=length(P);

v=zeros(n,T); %初始化期望利润矩阵

for i=1:n

v(i,1)=R(i,:)\*P(i,:)'; %第1期的n个期望利润

end

for k=2:T %计算第2至第T期的各期望利润

for i=1:n

v(i,k)=v(i,k-1)+P(i,:)\*v(:,k-1);

end

end

v

运行结果：

v = 6.0000 7.5000 10.0500 14.6550 23.3205

-3.0000 -2.4000 -0.8400 2.6760 10.1436