Higher, possibly multivariate, Order Markov Chains in markovchain package

Deepak Yadav, Tae Seung Kang, Giorgio Alfredo Spedicato

Abstract

The **markovchain** package contains functions to fit higher (possibly) multivariate order Markov chains. The functions are shown as well as simple exmaples

Keywords: Higher order Markov chains.

1. Higher Order Markov Chains

Continuous time Markov chains are discussed in the CTMC vignette which is a part of the package.

An experimental `fitHigherOrder` function has been written in order to fit a higher order Markov chain ((Ching, Huang, Ng, and Siu 2013, Ching, Ng, and Fung (2008))). `fitHigherOrder` takes two inputs

1. sequence: a categorical data sequence.
2. order: order of Markov chain to fit with default value 2.

The output will be a `list` which consists of

1. lambda: model parameter(s).
2. Q: a list of transition matrices. Q_i is the *ith* step transition matrix stored column-wise.
3. X: frequency probability vector of the given sequence.

Its quadratic programming problem is solved using `solnp` function of the Rsolnp package (Ghalanos and Theussl 2014).

```
R> require(markovchain)
R> library(Rsolnp)

R> data(rain)
R> fitHigherOrder(rain$rain, 2)

$lambda
[1] 0.5 0.5
```

```
$Q
$Q[[1]]
            0       1-5       6+
0   0.6605839 0.4625850 0.1976285
1-5 0.2299270 0.3061224 0.3122530
6+  0.1094891 0.2312925 0.4901186

$Q[[2]]
            0       1-5       6+
0   0.6021898 0.4489796 0.3412698
1-5 0.2445255 0.2687075 0.3214286
6+  0.1532847 0.2823129 0.3373016


$X
        0       1-5       6+
0.5000000 0.2691606 0.2308394
```

```
R> fitHigherOrder(rain$rain, 3)
```

```
$lambda
[1] 0.3333333 0.3333333 0.3333333

$Q
$Q[[1]]
            0       1-5       6+
0   0.6605839 0.4625850 0.1976285
1-5 0.2299270 0.3061224 0.3122530
6+  0.1094891 0.2312925 0.4901186

$Q[[2]]
            0       1-5       6+
0   0.6021898 0.4489796 0.3412698
1-5 0.2445255 0.2687075 0.3214286
6+  0.1532847 0.2823129 0.3373016

$Q[[3]]
            0       1-5       6+
0   0.5693431 0.4455782 0.4183267
1-5 0.2536496 0.2891156 0.2749004
6+  0.1770073 0.2653061 0.3067729


$X
        0       1-5       6+
0.5000000 0.2691606 0.2308394
```

2. Higher Order Multivariate Markov Chains

2.1. Introduction

HOMMC model is used for modeling behaviour of multiple categorical sequences generated by similar sources. The main reference is (Ching *et al.* 2008). Assume that there are s categorical sequences and each has possible states in M. In nth order MMC the state probability distribution of the jth sequence at time $t = r + 1$ depend on the state probability distribution of all the sequences (including itself) at times $t = r, r - 1, ..., r - n + 1$.

$$x_{r+1}^{(j)} = \sum_{k=1}^{s}\sum_{h=1}^{n} \lambda_{jk}^{(h)} P_h^{(jk)} x_{r-h+1}^{(k)}, j = 1, 2, ..., s, r = n - 1, n, ...$$

with initial distribution $x_0^{(k)}, x_1^{(k)}, ..., x_{n-1}^{(k)} (k = 1, 2, ..., s)$. Here

$$\lambda_{jk}^{(h)} \geq 0, 1 \leq j, k \leq s, 1 \leq h \leq n \; and \; \sum_{k=1}^{s}\sum_{h=1}^{n} \lambda_{jk}^{(h)} = 1, j = 1, 2, 3, ..., s.$$

Now we will see the simpler representation of the model which will help us understand the result of `fitHighOrderMultivarMC` method.

Let $X_r^{(j)} = ((x_r^{(j)})^T, (x_{r-1}^{(j)})^T, ..., (x_{r-n+1}^{(j)})^T)^T for \; j = 1, 2, 3, ..., s$. Then

$$\begin{pmatrix} X_{r+1}^{(1)} \\ X_{r+1}^{(2)} \\ . \\ . \\ . \\ X_{r+1}^{(s)} \end{pmatrix} = \begin{pmatrix} B^{11} & B^{12} & . & . & B^{1s} \\ B^{21} & B^{22} & . & . & B^{2s} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ B^{s1} & B^{s2} & . & . & B^{ss} \end{pmatrix} \begin{pmatrix} X_r^{(1)} \\ X_r^{(2)} \\ . \\ . \\ . \\ X_r^{(s)} \end{pmatrix} \text{where}$$

$$B^{ii} = \begin{pmatrix} \lambda_{ii}^{(1)} P_1^{(ii)} & \lambda_{ii}^{(2)} P_2^{(ii)} & . & . & \lambda_{ii}^{(n)} P_n^{(ii)} \\ I & 0 & . & . & 0 \\ 0 & I & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & I & 0 \end{pmatrix}_{mn*mn} \text{and}$$

$$B^{ij} = \begin{pmatrix} \lambda_{ij}^{(1)} P_1^{(ij)} & \lambda_{ij}^{(2)} P_2^{(ij)} & . & . & \lambda_{ij}^{(n)} P_n^{(ij)} \\ 0 & 0 & . & . & 0 \\ 0 & 0 & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & 0 & 0 \end{pmatrix}_{mn*mn} \text{when } i \neq j.$$

2.2. Representation of parameters in the code

$P_h^{(ij)}$ is represented as $Ph(i,j)$ and $\lambda_{ij}^{(h)}$ as Lambdah(i,j). For example: $P_2^{(13)}$ as $P2(1,3)$ and $\lambda_{45}^{(3)}$ as Lambda3(4,5).

2.3. Definition of HOMMC class

```
R> showClass("hommc")
```

```
Class "hommc" [package "markovchain"]

Slots:

Name:       order    states         P    Lambda     byrow      name
Class:    numeric character     array   numeric   logical character
```

Any element of hommc class is comprised by following slots:

1. states: a character vector, listing the states for which transition probabilities are defined.
2. byrow: a logical element, indicating whether transition probabilities are shown by row or by column.
3. order: order of Multivariate Markov chain.
4. P: an array of all transition matrices.
5. Lambda: a vector to store the weightage of each transition matrix.
6. name: optional character element to name the HOMMC

2.4. How to create an object of class HOMMC

```
R> states <- c('a', 'b')
R> P <- array(dim = c(2, 2, 4), dimnames = list(states, states))
R> P[ , , 1] <- matrix(c(1/3, 2/3, 1, 0), byrow = FALSE, nrow = 2, ncol = 2)
R> 
R> P[ , , 2] <- matrix(c(0, 1, 1, 0), byrow = FALSE, nrow = 2, ncol = 2)
R> 
R> P[ , , 3] <- matrix(c(2/3, 1/3, 0, 1), byrow = FALSE, nrow = 2, ncol = 2)
R> 
R> P[ , , 4] <- matrix(c(1/2, 1/2, 1/2, 1/2), byrow = FALSE, nrow = 2, ncol = 2)
R> 
R> Lambda <- c(.8, .2, .3, .7)
R> 
R> hob <- new("hommc", order = 1, Lambda = Lambda, P = P, states = states,
R+            byrow = FALSE, name = "FOMMC")
R> hob
```

```
Order of multivariate markov chain = 1
states = a b

List of Lambda's and the corresponding transition matrix (by cols) :
Lambda1(1,1) : 0.8
P1(1,1) :
          a b
a 0.3333333 1
b 0.6666667 0

Lambda1(1,2) : 0.2
P1(1,2) :
  a b
a 0 1
b 1 0

Lambda1(2,1) : 0.3
P1(2,1) :
          a b
a 0.6666667 0
b 0.3333333 1

Lambda1(2,2) : 0.7
P1(2,2) :
    a   b
a 0.5 0.5
b 0.5 0.5
```

2.5. Fit HOMMC

`fitHighOrderMultivarMC` method is available to fit HOMMC. Below are the 3 parameters of this method.

1. seqMat: a character matrix or a data frame, each column represents a categorical sequence.
2. order: order of Multivariate Markov chain. Default is 2.
3. Norm: Norm to be used. Default is 2.

3. A Marketing Example

We tried to replicate the example found in (Ching *et al.* 2008) for an application of HOMMC. A soft-drink company in Hong Kong is facing an in-house problem of production planning and inventory control. A pressing issue is the storage space of its central warehouse, which often finds itself in the state of overflow or near capacity. The company is thus in urgent needs to study the interplay between the storage space requirement and the overall growing sales demand. The product can be classified into six possible states (1, 2, 3, 4, 5, 6) according to their sales volumes. All products are labeled as 1 = no sales volume, 2 = very slow-moving (very low sales volume), 3 = slow-moving, 4 = standard, 5 = fast-moving or 6 = very fast-moving (very high sales volume). Such labels are useful from both marketing and production planning points of view. The data is cointaind in `sales` object.

```
R> data(sales)
R> head(sales)
```

```
     A   B   C   D   E
[1,] "6" "1" "6" "6" "6"
[2,] "6" "6" "6" "2" "2"
[3,] "6" "6" "6" "2" "2"
[4,] "6" "1" "6" "2" "2"
[5,] "2" "6" "6" "2" "2"
[6,] "6" "1" "6" "3" "3"
```

The company would also like to predict sales demand for an important customer in order to minimize its inventory build-up. More importantly, the company can understand the sales pattern of this customer and then develop a marketing strategy to deal with this customer. Customer's sales demand sequences of five important products of the company for a year. We expect sales demand sequences generated by the same customer to be correlated to each other. Therefore by exploring these relationships, one can obtain a better higher-order multivariate Markov model for such demand sequences, hence obtain better prediction rules.

In (Ching *et al.* 2008) application, they choose the order arbitrarily to be eight, i.e., n = 8. We first estimate all the transition probability matrices P_h^{ij} and we also have the estimates of the stationary probability distributions of the five products:.

$$\widehat{\boldsymbol{x}}^{(1)} = \begin{pmatrix}0.0818 & 0.4052 & 0.0483 & 0.0335 & 0.0037 & 0.4275\end{pmatrix}^{\boldsymbol{T}}$$

$$\widehat{\boldsymbol{x}}^{(2)} = \begin{pmatrix}0.3680 & 0.1970 & 0.0335 & 0.0000 & 0.0037 & 0.3978\end{pmatrix}^{\boldsymbol{T}}$$

$$\widehat{\boldsymbol{x}}^{(3)} = \begin{pmatrix}0.1450 & 0.2045 & 0.0186 & 0.0000 & 0.0037 & 0.6283\end{pmatrix}^{\boldsymbol{T}}$$

$$\widehat{\boldsymbol{x}}^{(4)} = \begin{pmatrix}0.0000 & 0.3569 & 0.1338 & 0.1896 & 0.0632 & 0.2565\end{pmatrix}^{\boldsymbol{T}}$$

$$\widehat{\boldsymbol{x}}^{(5)} = \begin{pmatrix}0.0000 & 0.3569 & 0.1227 & 0.2268 & 0.0520 & 0.2416\end{pmatrix}^{\boldsymbol{T}}$$

By solving the corresponding linear programming problems, we obtain the following higher-order multivariate Markov chain model:

$$\begin{aligned}
\boldsymbol{x}_{r+1}^{(1)} &= \boldsymbol{P}_1^{(12)}\boldsymbol{x}_r^{(2)}\\
\boldsymbol{x}_{r+1}^{(2)} &= 0.6364\boldsymbol{P}_1^{(22)}\boldsymbol{x}_r^{(2)} + 0.3636\boldsymbol{P}_3^{(22)}\boldsymbol{x}_r^{(2)}\\
\boldsymbol{x}_{r+1}^{(3)} &= \boldsymbol{P}_1^{(35)}\boldsymbol{x}_r^{(5)}\\
\boldsymbol{x}_{r+1}^{(4)} &= 0.2994\boldsymbol{P}_8^{(42)}\boldsymbol{x}_r^{(2)} + 0.4324\boldsymbol{P}_1^{(45)}\boldsymbol{x}_r^{(5)} + 0.2681\boldsymbol{P}_2^{(45)}\boldsymbol{x}_r^{(5)}\\
\boldsymbol{x}_{r+1}^{(5)} &= 0.2718\boldsymbol{P}_8^{(52)}\boldsymbol{x}_r^{(2)} + 0.6738\boldsymbol{P}_1^{(54)}\boldsymbol{x}_r^{(4)} + 0.0544\boldsymbol{P}_2^{(55)}\boldsymbol{x}_r^{(5)}
\end{aligned}$$

According to the constructed 8th order multivariate Markov model, Products A and B are closely related. In particular, the sales demand of Product A depends strongly on Product B.

The main reason is that the chemical nature of Products A and B is the same, but they have different packaging for marketing purposes. Moreover, Products B, C, D and E are closely related. Similarly, products C and E have the same product flavor, but different packaging. In this model, it is interesting to note that both Product D and E quite depend on Product B at order of 8, this relationship is hardly to be obtained in conventional Markov model owing to huge amount of parameters. The results show that higher-order multivariate Markov model is quite significant to analyze the relationship of sales demand.

```
R> # fit 8th order multivariate markov chain
R> object <- fitHighOrderMultivarMC(sales, order = 8, Norm = 2)
```

We choose to show only results shown in the paper. We see that λ values are quite close, but not equal, to those shown in the original paper.

```
Order of multivariate markov chain = 8

states = 1 2 3 4 5 6

List of Lambda's and the corresponding transition matrix (by cols) :

Lambda1(1,2) : 0.9999989
P1(1,2) :
           1         2         3         4 5          6
1 0.06060606 0.1509434 0.0000000 0.1666667 0 0.07547170
2 0.44444444 0.4716981 0.4444444 0.1666667 1 0.33018868
3 0.01010101 0.1320755 0.2222222 0.1666667 0 0.02830189
4 0.01010101 0.0754717 0.2222222 0.1666667 0 0.01886792
5 0.01010101 0.0000000 0.0000000 0.1666667 0 0.00000000
6 0.46464646 0.1698113 0.1111111 0.1666667 0 0.54716981

Lambda1(2,2) : 0.4812247
P1(2,2) :
           1          2         3         4 5           6
1 0.40404040 0.20754717 0.0000000 0.1666667 1 0.433962264
2 0.11111111 0.47169811 0.3333333 0.1666667 0 0.132075472
3 0.02020202 0.05660377 0.3333333 0.1666667 0 0.009433962
4 0.00000000 0.00000000 0.0000000 0.1666667 0 0.000000000
5 0.00000000 0.00000000 0.1111111 0.1666667 0 0.000000000
6 0.46464646 0.26415094 0.2222222 0.1666667 0 0.424528302

Lambda3(2,2) : 0.3870364
P3(2,2) :
           1          2         3         4 5          6
1 0.40404040 0.16981132 0.3333333 0.1666667 0 0.44230769
2 0.18181818 0.33962264 0.2222222 0.1666667 0 0.14423077
3 0.03030303 0.05660377 0.0000000 0.1666667 0 0.02884615
```

```
4 0.00000000 0.00000000 0.0000000 0.1666667 0 0.00000000
5 0.00000000 0.00000000 0.1111111 0.1666667 0 0.00000000
6 0.38383838 0.43396226 0.3333333 0.1666667 1 0.38461538

Lambda1(3,5) : 0.676394
P1(3,5) :
          1          2         3         4          5          6
1 0.1666667 0.09473684 0.1515152 0.1639344 0.07142857 0.21538462
2 0.1666667 0.18947368 0.2727273 0.2295082 0.14285714 0.18461538
3 0.1666667 0.04210526 0.0000000 0.0000000 0.00000000 0.01538462
4 0.1666667 0.00000000 0.0000000 0.0000000 0.00000000 0.00000000
5 0.1666667 0.01052632 0.0000000 0.0000000 0.00000000 0.00000000
6 0.1666667 0.66315789 0.5757576 0.6065574 0.78571429 0.58461538

Lambda8(4,2) : 0.2729107
P8(4,2) :
           1          2         3         4 5          6
1 0.00000000 0.00000000 0.0000000 0.1666667 0 0.00000000
2 0.34343434 0.18867925 0.6666667 0.1666667 0 0.42424242
3 0.10101010 0.16981132 0.0000000 0.1666667 1 0.14141414
4 0.20202020 0.22641509 0.1111111 0.1666667 0 0.17171717
5 0.08080808 0.09433962 0.1111111 0.1666667 0 0.03030303
6 0.27272727 0.32075472 0.1111111 0.1666667 0 0.23232323

Lambda1(4,5) : 0.2293906
P1(4,5) :
          1          2          3          4          5          6
1 0.1666667 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
2 0.1666667 0.47368421 0.21212121 0.03278689 0.00000000 0.64615385
3 0.1666667 0.10526316 0.21212121 0.19672131 0.07142857 0.09230769
4 0.1666667 0.00000000 0.24242424 0.54098361 0.57142857 0.03076923
5 0.1666667 0.01052632 0.03030303 0.18032787 0.28571429 0.00000000
6 0.1666667 0.41052632 0.30303030 0.04918033 0.07142857 0.23076923

Lambda2(4,5) : 0.2989146
P2(4,5) :
          1          2          3          4          5          6
1 0.1666667 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
2 0.1666667 0.55319149 0.36363636 0.06557377 0.00000000 0.41538462
3 0.1666667 0.13829787 0.09090909 0.21311475 0.28571429 0.04615385
4 0.1666667 0.05319149 0.24242424 0.40983607 0.64285714 0.06153846
5 0.1666667 0.02127660 0.06060606 0.16393443 0.07142857 0.03076923
6 0.1666667 0.23404255 0.24242424 0.14754098 0.00000000 0.44615385

Lambda8(5,2) : 0.2268906
P8(5,2) :
           1          2         3         4 5          6
```

```
1 0.00000000 0.00000000 0.0000000 0.1666667 0 0.00000000
2 0.35353535 0.20754717 0.6666667 0.1666667 1 0.39393939
3 0.10101010 0.15094340 0.0000000 0.1666667 0 0.13131313
4 0.22222222 0.30188679 0.2222222 0.1666667 0 0.20202020
5 0.09090909 0.03773585 0.0000000 0.1666667 0 0.03030303
6 0.23232323 0.30188679 0.1111111 0.1666667 0 0.24242424

Lambda1(5,4) : 0.2268058
P1(5,4) :
          1          2          3          4          5          6
1 0.1666667 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
2 0.1666667 0.48421053 0.16666667 0.01960784 0.05882353 0.60869565
3 0.1666667 0.10526316 0.16666667 0.15686275 0.05882353 0.11594203
4 0.1666667 0.00000000 0.44444444 0.62745098 0.64705882 0.02898551
5 0.1666667 0.01052632 0.02777778 0.15686275 0.23529412 0.00000000
6 0.1666667 0.40000000 0.19444444 0.03921569 0.00000000 0.24637681

Lambda2(5,5) : 0.5398433
P2(5,5) :
          1          2          3          4          5          6
1 0.1666667 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
2 0.1666667 0.52127660 0.42424242 0.04918033 0.07142857 0.43076923
3 0.1666667 0.12765957 0.03030303 0.19672131 0.21428571 0.07692308
4 0.1666667 0.05319149 0.33333333 0.54098361 0.50000000 0.07692308
5 0.1666667 0.02127660 0.03030303 0.11475410 0.21428571 0.01538462
6 0.1666667 0.27659574 0.18181818 0.09836066 0.00000000 0.40000000
```

References

Ching WK, Huang X, Ng MK, Siu TK (2013). “Higher-order markov chains.” In *Markov Chains*, pp. 141–176. Springer.

Ching WK, Ng MK, Fung ES (2008). “Higher-order multivariate Markov chains and their applications.” *Linear Algebra and its Applications*, **428**(2), 492–507.

Ghalanos A, Theussl S (2014). *Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method.* R package version 1.15.

Affiliation:

Deepak Yadav
B-Tech student, Computer Science and Engineering
Indian Institute of Technology, Varanasi Uttar Pradesh - 221 005, India
E-mail: deepakyadav.iitbhu@gmail.com

Tae Seung Kang
Ph.D student, Computer & Information Science & Engineering
University of Florida Gainesville, FL, USA
E-mail: tskang3@gmail.com

Giorgio Alfredo Spedicato
Ph.D C.Stat ACAS, UnipolSai R&D
Via Firenze 11 Paderno Dugnano 20037 Italy
E-mail: spedicato_giorgio@yahoo.it