Lab 9 – Łańcuchy Markowa DMCH

Wprowadzenie

DTMC – Discrete Time Markov Chain, to ciągi zmiennych losowych

$X_1, X_2, X_3, \ldots : \Omega \to \mathrm{S} = \{\mathrm{s}_1, \ldots, \mathrm{s}_r\}$,

które przyjmują wartości w skończonym zbiorze stanów S.

Warunek Markowa:

$$\Pr\{X_{n+1} = x_{n+1} \,|X_1 = x_1, \ldots, X_n = x_n\} = \Pr\{X_{n+1} = x_{n+1} \,|X_n = x_n\} \;\; \forall x_1, \ldots, x_{n+1} \in S$$

Warunek shiftu, jednorodne łańcuchy Markowa (time-homogeneous MC)

$$\Pr\{X_{n+1+k} = s_i \,|X_{n+k} = s_j\} = \Pr\{X_{n+1} = s_i \,|X_n = s_j\} \quad \forall s_i, s_j \in S, \; k \in \mathbb{N}$$

Klasyfikacja stanów łańcucha jednorodnego:

Stany nieistotne – można z nich wyjść, ale nie można do nich wrócić

Stany pochłaniające – można do nich wejść, ale nie można wyjść.

Stany komunikujące się – można pomiędzy nimi przechodzić w skończonym czasie. Relacja korespondencji dzieli istotne stany łańcucha na klasy równoważności według relacji komunikowania się. Klasy abstrakcji stanów pochłaniających są jednoelementowe.

Klasę komunikujących się stanów nazwiemy domkniętą, jeżeli z żadnego jej stanu nie da się wyjść do stanu z innej klasy.

Stan posiada period *k*, jeżeli może do niego wrócić z niego samego po co najwyżej *k* epokach. Stany o periodzie *k*=1 nazywamy aperiodyczne.

Jeżeli dwa stany należą do tej samej klasy komunikacji, to maja ten sam period.

Łańcuch, który ma tylko jedną klasę komunikacji nazywamy nieredukowalnym (irreducible).

Jeżeli łańcuch jednorodny jest nieredukowalny i aperiodyczny oraz nie ma stanów nieistotnych, to ma graniczny rozkład prawdopodobieństwa, który jest punktem stałym równania Kołmogorowa.

Ważne funkcje działające na obiekcie typu markovchain

Method returns:	
absorbingStates	- the absorbing states of the transition matrix, if any.
conditionalDistribution	- the conditional distribution of the subsequent state s_j, given actual state s_i.
canonicForm	- the transition matrix into canonic form.
is.accessible	- verification if a state *j* is reachable from state *i*.
is.irreducible	- verification whether a DTMC is irreducible.

period	- the period of an irreducible DTMC.
steadyStates	- the vector(s) of steady state(s) in matrix form.
summary	- DTMC summary.
transientStates	- the transient states of the transition matrix, if any.

Definicja łańcucha w R

Macierz przejścia, etykiety stanów, obiekt łańcucha stacjonarnego,

```
library('markovchain')
#
# usatawiamy dane dla pakietu
#
weatherStates <- c("sunny", "cloudy", "rain")
byRow <- TRUE
data = c(0.70, 0.2, 0.1,
        + 0.3, 0.4, 0.3,
        + 0.2, 0.45, 0.35)
data
weatherStates
byrow = byRow
nrow = 3
ncol = 3
dimnames = list(weatherStates, weatherStates)
weatherMatrix <- matrix(data, nrow, ncol, byrow, dimnames)
weatherMatrix
states = weatherStates
transitionMatrix = weatherMatrix
name = "Weather"
mcWeather <- new("markovchain", states, byRow, transitionMatrix, name)
#
# tworzymy obiekt typu markovchain
#
defaultMc <- new("markovchain")
```

obiekt łańcucha niestacjonarnego

Prognozy Markowowskie (rozkłady prawdopodobieństwa stanów w kolejnych krokach)

Wyznaczanie prognozy wieloetapowej dla procesu opisanego DMCH,

```
# ustawiamy stan początkowy
#
initialState <- c(0, 1, 0)
#
# liczymy prognozy po 2 i 7 dniach
#
after2Days <- initialState * (mcWeather * mcWeather)
```

```
after7Days <- initialState * (mcWeather ^ 7)
after2Days
round(after7Days, 3)
```

Symulacja według rozkładów stanów łańcucha:

Dla obiektu z poprzedniego zadania (łańcuch pogody o 3 stanach) oraz określonej macierzy przejścia (łańcuch stacjonarny) można wygenerować ciąg losowany zgodnie z tymi prawdopodobieństwami:

```
weathersOfDays <- rmarkovchain(n=365, object=mcWeather, t0="sunny")
weathersOfDays[1:30]
```

Obliczanie trajektorii oczekiwanej, obliczanie trajektorii kwantylowej dla dyskretyzacji procesów o rozkładach ciągłych.

Zadanie X. Skonstruuj macierz przejścia dla prognozy opadów, w oparciu o szereg rozdzielczy złożony z 10 stanów

Łańcuchy ergodyczne, rozkłady stacjonarne

Wyznaczanie rozkładu stacjonarnego.

```
weatherStates <- c("sunny", "cloudy", "rain")
byRow <- TRUE
data = c(0.70, 0.2, 0.1,
        + 0.3, 0.4, 0.3,
        + 0.2, 0.45, 0.35)
data
weatherStates
byrow = byRow
nrow = 3
ncol = 3
dimnames = list(weatherStates, weatherStates)
weatherMatrix <- matrix(data, nrow, ncol, byrow, dimnames)
weatherMatrix
states = weatherStates
transitionMatrix = weatherMatrix
name = "Weather"
mcWeather <- new("markovchain", states, byRow, transitionMatrix, name)
#
# tworzymy obiekt typu markovchain
#
defaultMc <- new("markovchain")
#
# obliczamy stacjonarny rozkład prawdopodobieństwa
```

```
#
steadyStates(mcWeather)
```

Rola rozkładów stacjonarnych w ergodycznym procesie poszukiwania ewolucyjnego – uczenia maszynowego

Stany pochłaniające, stany osiągalne

Wyznaczanie stanów pochłaniających.

```
# definiujemy nowy łańcucg stacjonarny
#
abstractStates <- c("first", "second", "thrid")
byRow <- TRUE
data = c(1.0, 0.0, 0.0,
        + 0.3, 0.4, 0.3,
        + 0.2, 0.45, 0.35)
byrow = byRow
nrow = 3
ncol = 3
dimnames = list(abstractStates, abstractStates)
abstractMatrix <- matrix(data, nrow, ncol, byrow, dimnames)
abstractMatrix
states = abstractStates
transitionMatrix = abstractMatrix
name = "abstract"
mcabstract <- new("markovchain", states, byRow, transitionMatrix, name)
#
# próba (udana) oblicznia stanów absorbujących
#
absorbingStates(mcabstract)
#
```

Rola stanów pochłaniających w poszukiwaniach ewolucyjnych – uczeniu maszynowym. Wyznaczanie stanów osiągalnych z danego stanu.

Estymacja łańcuchów Markowa na podstawie realizacji szeregów czasowych

markovchainFit - Function to return fitted Markov chain for a given sequence.

Możemy estymować macierz (lub macierze) przejścia łańcucha Markowa na podstawie obserwowanego szeregu czasowego jego stanów. Można stosować różne estymatory macierzy przejścia.

Dla estymatora MLE (Maximum Likelihood Estimator) mamy

$$\hat{p}_{ij}^{MLE} = \frac{n_{ij}}{\sum_{u=1}^{k} n_{iu}}$$

gdzie n_{ij} jest ilością podciągów $(X_n = s_i, X_{n+1} = s_j)$ odnalezionych w obserwowanym szeregu.

Dla wykonania przykładu wykorzystamy obiekt z zadania (symulacja,łańcuch pogody o 3 stanach) oraz określonej macierzy przejścia (łańcuch stacjonarny) i wygenerowany w ten sposób ciąg stanów:

```
weathersOfDays <- rmarkovchain(n=365, object=mcWeather, t0="sunny")
weathersOfDays[1:30]
```

Estymacja wykonana będzie poprzez:

```
weatherFittedMLE <- markovchainFit(data = weathersOfDays, method = "mle",
+ name = "Weather MLE")
 weatherFittedMLE$estimate
```

Możemy użyć estymatora Laplace'a ze stabilizatorem α bardzo małą wartością

$$\hat{p}_{ij}^{LS} = \frac{n_{ij} + \alpha}{\sum_{u=1}^{k}(n_{iu} + \alpha)}$$

```
weatherFittedLAPLACE <- markovchainFit(data = weathersOfDays,
+ method = "laplace", laplacian = 0.01,
+ name = "Weather LAPLACE")
weatherFittedLAPLACE$estimate
```

Predykcja (prognoza modalna)

predict	- Method to calculate predictions from markovchain or markovchainList objects.

Funkcja ta daje możliwość predykcji trajektorii modalnej stanów na n.ahead kroków czasowych do przodu, zakładając że zaszła sekwencja newdata stanów do stanu aktualnego .

Przykład predykcji wykorzystuje fitowany stacjonarny łańcuch Markowa stanów pogody z poprzedniego ćwiczenia:

```
predict(object = weatherFittedMLE$estimate, newdata = c("cloudy", "sunny"), n.ahead = 3)
```