Package 'SAutomata'

November 2, 2018

Type Package

Title Inference and Learning in Stochastic Automata

Version 0.1.0

Maintainer Muhammad Kashif Hanif <mkashifhanif@gcuf.edu.pk>

Description Machine learning provides algorithms that can learn from data and make inferences or predictions. Stochastic automata is a class of input/output devices which can model components. This work provides implementation an inference algorithm for stochastic automata which is similar to the Viterbi algorithm. Moreover, we specify a learning algorithm using the expectation-maximization technique and provide a more efficient implementation of the Baum-Welch algorithm for stochastic automata. This work is based on Inference and learning in stochastic automata was by Karl-Heinz Zimmermann(2017) <doi:10.12732/ijpam.v115i3.15>.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 2.0.0)

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Muhammad Kashif Hanif [cre, aut],
Muhammad Umer Sarwar [aut],
Rehman Ahmad [aut],
Zeeshan Ahmad [aut],
Karl-Heinz Zimmermann [aut]

Repository CRAN

Date/Publication 2018-11-02 18:00:03 UTC

R topics documented:

`BaumWelch` *Inferring the Forward and Backward Probabilities of a Stochastic Automata Model via the Baum-Welch algorithm*

Description

For an initial Stochastic Automata Model (SA) and a given sequence of observations, the Baum-Welch algorithm infers optimal forward and backward probabilities to the SA. Since the Baum-Welch algorithm is a variant of the Expectation-Maximisation algorithm, the algorithm converges to a local solution which might not be the global optimum.

Usage

```
BaumWelch(initsa, x, y, m, error, theta = NULL)
```

Arguments

`initsa`	A Stochastic Automata Model.
`x`	A sequence of inputs.
`y`	A sequence of outputs.
`m`	Maximum length of sequence to create sample set for learning.
`error`	Maximum error rate.
`theta`	Optional Conditional Probabilities.

Value

Returns the conditional probabilities by learning the sample set.

Examples

```
states<-c('s1','s2')
inputSymbols<-c('a','b')
outputSymbols<-c(0,1)
transProb<-matrix(c(0.70,0.50, 0.30,0.50), nrow = 2, ncol = 2,byrow = TRUE)
emissionProb<-matrix(c(0.50,0.30, 0.40,0.60,.50,.70,.60,.40), nrow = 2, ncol = 4, byrow = TRUE)
initsa<-initSA(states,inputSymbols,outputSymbols,emissionProb,transProb)
x<-c('b','a')
y<-c(0,1)
m<-1
error<-10
BaumWelch(initsa, x, y, m, error)
```

initSA *Initialisation of SA's*

Description

This function initialises a general discrete time and discrete space Stochastic Automata(SA). A SA consists of an alphabet of states, input and output symbols. The SA is designed to make inference on the states through the observation of input symbols on output symbols. The stochastics of the SA is fully described by the set of states, input and output symbols and the conditional probablities (i.e. state transition probablity and output symbols emission probablity by inputs symbols on state set).

Usage

```
initSA(states,inputSymbols,outputSymbols,emissionProb,transitionProb)
```

Arguments

states	Vector with names of states.
inputSymbols	Vector with names of input Symbols.
outputSymbols	Vector with names of output Symbols.
emissionProb	Stochastic matrix containing emission probablities of output symbols between states and input symbols.
transitionProb	Stochastic matrix containing probablities between states.

Details

The column sum of transitionProb and emissionProb must be equal to 1. Otherwise this function generates an error message.

Value

This function initSA returns an SA that consists of a list of 5 elements:

States	Vector with names of states.
inputSymbols	Vector with names of input Symbols.
outputSymbols	Vector with names of output Symbols.
outputSymbols	Vector with names of output Symbols.
emissionProb	Annotated matrix containing emission probablities of output symbols between states and input symbols.
transitionProb	Annotated matrix containing probablities between states.

Author(s)

Rehman Ahmad <rehman.ahmad777@gmail.com>

Examples

```
states<-c('s1','s2')
inputSymbols<-c('a','b')
outputSymbols<-c(0,1)
transProb<-matrix(c(0.70,0.50, 0.30,0.50), nrow = 2, ncol = 2,byrow = TRUE)
emissionProb<-matrix(c(0.50,0.30, 0.40,0.60,.50,.70,.60,.40), nrow = 2, ncol = 4, byrow = TRUE)
initsa<-initSA(states,inputSymbols,outputSymbols,emissionProb,transProb)
```

Sbackward *Computes The Backward Probabilities*

Description

The Sbackward function computes the backward probabilities. The backward probabilities for state 'S' up to output observations at time k is defined as the probability of observing the sequance of observations 'Y'(y_1, ... ,y_k) and that state at time 'k' is 'S'. that is:
f[k,X] := Prob(Y_k+1 = y_k+1, ... , Y_k = y_k ,S_k = S).
Where Y_1, ... ,Y_n = y_1, ... , y_n is sequance of observed emissions and S_k is a random variable that represents the state at time k.

Usage

```
Sbackward(initsa, x, y, theta=NULL)
```

Arguments

initsa	A Stochastic Model.
x	A vector of input sequance.
y	A vector of Output sequance.
theta	Optional Conditional Probabilities.

Value

Return Value:
backward A matrix containing the backward probabilities. The probabilities are given on a logarithmic scale (natural logarithm). This first dimension refer to the time and the second dimension to states.

Author(s)

Rehman Ahmad <rehman.ahmad777@gmail.com>

Examples

```
states<-c('s1','s2')
inputSymbols<-c('a','b')
outputSymbols<-c(0,1)
transProb<-matrix(c(0.70,0.50, 0.30,0.50), nrow = 2, ncol = 2,byrow = TRUE)
emissionProb<-matrix(c(0.50,0.30, 0.40,0.60,.50,.70,.60,.40), nrow = 2, ncol = 4, byrow = TRUE)
initsa<-initSA(states,inputSymbols,outputSymbols,emissionProb,transProb)
x<-c('b','a')
y<-c(0,1)
sb<-Sbackward(initsa, x, y)
```

---

scores *Calculation of Probabilities (Not For End User)*

Description

This function is not for end user.

Usage

```
scores(initsa=NULL,theta=NULL)
```

Arguments

initsa	Model SA.
theta	Optional(Conditional Prababilities).

Examples

```
## Not run:
scores(initsa)

## End(Not run)
```

`Sforward` *Computes The Forward Probabilities*

Description

The `Sforward` function computes the forward probabilities. The forward probabilities for state 'S' up to output observations at time k is defined as the probability of observing the sequance of observations 'Y'(y_1, ... ,y_k) and that state at time 'k' is 'S'. that is:
f[k,X] := Prob(Y_1 = y_1, ... , Y_k = y_k ,S_k = S).
Where Y_1, ... ,Y_n = y_1, ... , y_n is sequance of observed emissions and S_k is a random variable that represents the state at time k.

Usage

```
Sforward(initsa, x, y, theta=NULL)
```

Arguments

`initsa`	A Stochastic Model.
`x`	A vector of input sequance.
`y`	A vector of Output sequance.
`theta`	Optional Conditional Probabilities.

Value

Return Value:
`forward` A matrix containing the forward probabilities. The probabilities are given on a logarithmic scale (natural logarithm). This first dimension refer to the time and the second dimension to states.

Author(s)

Rehman Ahmad <rehman.ahmad777@gmail.com>

Examples

```
states<-c('s1','s2')
inputSymbols<-c('a','b')
outputSymbols<-c(0,1)
transProb<-matrix(c(0.70,0.50, 0.30,0.50), nrow = 2, ncol = 2,byrow = TRUE)
emissionProb<-matrix(c(0.50,0.30, 0.40,0.60,.50,.70,.60,.40), nrow = 2, ncol = 4, byrow = TRUE)
initsa<-initSA(states,inputSymbols,outputSymbols,emissionProb,transProb)
x<-c('b','a')
y<-c(0,1)
sf<-Sforward(initsa, x, y)
```

TOC.sampleData *Learning (Not For End User)*

Description

This function is not for end user.

Usage

```
TOC.sampleData(initsa=NULL,n)
```

Arguments

initsa	SA Model.
n	Length of input sample set sequence.

Examples

```
## Not run:
states<-c('s1','s2')
inputSymbols<-c('a','b')
outputSymbols<-c(0,1)
transProb<-matrix(c(0.70,0.50, 0.30,0.50), nrow = 2, ncol = 2,byrow = TRUE)
emissionProb<-matrix(c(0.50,0.30, 0.40,0.60,.50,.70,.60,.40), nrow = 2, ncol = 4, byrow = TRUE)
initsa<-initsa(states,inputSymbols,outputSymbols,emissionProb,transProb)
n<-3
TOC.sampleData(initsa, n)



## End(Not run)
```

Index