

Package: greta.hmm (via r-universe)

February 18, 2025

Version 0.0.0.9000
Title Hidden Markov Modelling in greta
Description Extends greta by providing functionality for fitting Hidden Markov Models (HMMs).
License Apache License 2.0
Encoding UTF-8
LazyData true
ByteCompile true
Depends greta (>= 0.3.0)
Imports R6, tensorflow
Suggests testthat, covr
Roxygen list(markdown = TRUE)
RoxygenNote 6.1.1
Config/pak/sysreqs libpng-dev python3
Repository <https://greta-dev.r-universe.dev>
RemoteUrl <https://github.com/greta-dev/greta.hmm>
RemoteRef HEAD
RemoteSha 86352444e9b0ded2446835186c7f5a9c6349ea25

Contents

hmm	2
simulate_hmm	3
Index	5

Description

`hmm()` create a variable greta array following a Hidden Markov Model (HMM) 'distribution'. That is a probability distribution whose density is given by the probability of observing a sequence of observed states (the variable greta array), given a transition and an emission matrix as its parameters. This can be viewed as a compound distribution consisting of a categorical distribution over observed states, conditional on hidden states which (sequentially) follow another categorical distribution of states. The log density is calculated by analytically integrating out the hidden states using the forward algorithm.

This is a discrete, multivariate distribution, and is most likely to be used with `distribution()` to define a complete HMM, as in the example.

The transition and emission matrices should represent simplices; having rows summing to 1. These can be create with e.g. `imultilogit()`.

Usage

```
hmm(initial, transition, emission, n_timesteps)
```

Arguments

<code>initial</code>	a length K row vector (ie. 1 x K matrix) of probabilities of initial hidden states
<code>transition</code>	a K x K matrix of transition probabilities between hidden states
<code>emission</code>	a K x N matrix of emission probabilities between hidden and observed states
<code>n_timesteps</code>	the number of timesteps (length of the observed state matrix) - must be a positive scalar integer

Examples

```
## Not run:

# simulate data
n_hidden <- 2
n_observable <- 2
timesteps <- 20
initial <- random_simplex_matrix(1, n_hidden)
transition <- random_simplex_matrix(n_hidden,
                                   n_hidden)
emission <- random_simplex_matrix(n_hidden,
                                  n_observable)
hmm_data <- simulate_hmm(initial,
                        transition,
                        emission,
                        timesteps)
obs <- hmm_data$observed
```

```

# create simplex variables for the parameters
initial <- dirichlet(ones(1, n_hidden))
transition <- dirichlet(ones(n_hidden, n_hidden))
emission <- dirichlet(ones(n_hidden, n_observable))

# define the HMM over the observed states
distribution(obs) <- hmm(initial, transition, emission, timesteps)

# build the model
m <- model(transition, emission, initial)

## End(Not run)

```

simulate_hmm	<i>Simulate Sequences of Hidden and Observed States from a Hidden Markov Model</i>
--------------	--

Description

Generate corresponding sequences of hidden and observed states from a Hidden Markov Model (HMM) with specified transition and emission matrices. The simulation is carried out directly in R and is provided only to generate datasets for demonstrating or testing HMMs.

Usage

```

simulate_hmm(initial = random_simplex_matrix(1, 3),
             transition = random_simplex_matrix(3, 3),
             emission = random_simplex_matrix(3, 5), n_timesteps = 10)

random_simplex_matrix(nrow, ncol)

```

Arguments

initial	a length K row vector (ie. 1 x K matrix) of probabilities of initial hidden states
transition	a K x K matrix of transition probabilities between hidden states
emission	a K x N matrix of emission probabilities between hidden and observed states
n_timesteps	the number of timesteps (length of the observed state matrix) - must be a positive scalar integer
nrow, ncol	the number of rows and columns in the matrix

Examples

```

# numbers of hidden and observable states
n_hidden <- 3
n_observable <- 5

```

```
# generate a square matrix of transition probabilities in the hidden states
transition <- simulate_simplex_matrix(n_hidden, n_hidden)

# and a rectangular matrix of probabilities of each observable state,
# given each hidden state
emission <- simulate_simplex_matrix(n_hidden, n_observable)

# simulate an HMM for 10 timesteps
hmm_data <- simulate_hmm(transition, emission, 10)

# pull out the observed states
hmm_data$observed
```

Index

distribution, [2](#)

hmm, [2](#)

imultilogit, [2](#)

random_simplex_matrix (simulate_hmm), [3](#)

simulate_hmm, [3](#)