Package: proclhmm (via r-universe)

September 4, 2024

Title Latent Hidden Markov Models for Response Process Data

Version 1.0.0

Maintainer Xueying Tang <xueyingtang1989@gmail.com>

Description Provides functions for simulating from and fitting the latent hidden Markov models for response process data (Tang, 2024) <doi:10.1007/s11336-023-09938-1>. It also includes functions for simulating from and fitting ordinary hidden Markov models.

BugReports https://github.com/xytangtang/proclhmm/issues

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

LinkingTo Rcpp

Imports Rcpp (>= 1.0.8.3), statmod (>= 1.4.36)

Repository https://xytangtang.r-universe.dev

RemoteUrl https://github.com/xytangtang/proclhmm

RemoteRef HEAD

RemoteSha b7dff6b9001900a198921806d0a268761b60dd06

Contents

compute_P1_lhmm *Compute LHMM probabilities from parameters*

Description

Compute initial state probability from LHMM parameters; currently, the initial state probability does not depend on latent traits

Usage

```
compute_P1_lhmm(para_P1)
```

Arguments

para_P1	a vector of length K-1. parameters of initial state probability vector

Value

initial state probability vector of length K

See Also

compute_PQ_lhmm for state-transition and state-action probabilities of LHMM, compute_paras_hmm for computing probabilities in HMM.

Examples

```
paras <- sim_lhmm_paras(5, 2)
P1 <- compute_P1_lhmm(paras$para_P1)
```

`compute_paras_hmm` *Compute probabilities from logit scale parameters in HMM*

Description

Compute probabilities from logit scale parameters in HMM

Usage

```
compute_paras_hmm(para_P, para_Q, para_P1)
```

Arguments

`para_P` `K` by `K-1` matrix. parameters of state-transition probability matrix

`para_Q` `K` by `N-1` matrix. parameters of state-action (emission) probability matrix

`para_P1` `K-1` vector. parameters of initial state probability distribution

Value

a list of three elements:

`P` `K` by `K` state-transition probability matrix

`Q` `K` by `N` state-action (emission) probability matrix

`P1` initial state probability vector of length `K`

See Also

`compute_PQ_lhmm`, `compute_P1_lhmm` for computing probabilities in LHMM

Examples

```
paras <- sim_hmm_paras(5, 2, return_prob=FALSE)
prob_paras <- compute_paras_hmm(paras$para_P, paras$para_Q, paras$para_P1)
```

compute_PQ_lhmm *Compute LHMM probabilities from parameters*

Description

Compute state-transition and state-action (emission) probability matrices from LHMM parameters

Usage

```
compute_PQ_lhmm(theta, para_a, para_b, para_alpha, para_beta)
```

Arguments

`theta`	latent trait
`para_a`	`K` by `K-1` matrix. discrimination parameters of state transition probability matrix
`para_b`	`K` by `K-1` matrix. location parameters of state transition probability matrix
`para_alpha`	`K` by `N-1` matrix. discrimination parameters of state-action (emission) probability matrix
`para_beta`	`K` by `N-1` matrix. location parameters of state-action (emission) probability matrix

Value

A list of two elements

`P`	`K` by `K` state-transition probability matrix
`Q`	`K` by `N` state-action probability matrix

See Also

`compute_P1_lhmm` for initial state probabilities of LHMM, `compute_paras_hmm` for computing probabilities in HMM.

Examples

```
paras <- sim_lhmm_paras(5, 2)
prob_paras <- compute_PQ_lhmm(1.5, paras$para_a, paras$para_b, paras$para_alpha, paras$para_beta)
```

`compute_theta` *Estimate latent traits in LHMM*

Description

Compute MAP estimates of latent traits given LHMM parameters

Usage

```
compute_theta(int_seqs, para_a, para_b, para_alpha, para_beta, para_P1, n_pts)
```

Arguments

`int_seqs`	a list of `n` action sequences where actions are coded as integers 0, ..., N-1
`para_a`	`K` by `K-1` matrix. discrimination parameters of state transition probability matrix
`para_b`	`K` by `K-1` matrix. location parameters of state transition probability matrix
`para_alpha`	`K` by `N-1` matrix. discrimination parameters of state-action (emission) probability matrix
`para_beta`	`K` by `N-1` matrix. location parameters of state-action (emission) probability matrix
`para_P1`	a vector of length `K-1`. parameters of initial state probability vector
`n_pts`	number of quadrature points

Value

a vector of length `n`. Estimated latent traits.

`find_state_seq` *Viterbi algorithm for HMM*

Description

Find the most likely hidden state sequence of an observed sequence under HMM

Usage

```
find_state_seq(seq, P1, P, Q)
```

Arguments

`seq`	An action sequence coded in integers
`P1`	initial state probability vector of length `K`
`P`	`K` by `K` state transition probability matrix
`Q`	`K` by `N` state-action (emission) probability matrix

Value

a hidden state sequence coded in integers

---

hmm *MMLE of HMM*

Description

Maximum marginalized likelihood estimation of HMM. Optimization is performed through `optim`.

Usage

```
hmm(action_seqs, K, paras, ...)
```

Arguments

`action_seqs`	a list of `n` action sequences
`K`	number of hidden states
`paras`	a list of elements named `para_P1`, `para_P`, and `para_Q`, providing initial values of model parameters
`...`	additional arguments passed to `optim`

Value

a list containing the following elements

`seqs`	action sequences coded in integers
`K`	number of hidden states
`N`	number of distinct actions
`paras_init`	a list containing initial values of parameters
`paras_est`	a list containing parameter estimates
`init_mllh`	initial value of the marginalized likelihood function
`opt_mllh`	maximized marginalized likelihood function
`opt_res`	object returned by `optim`

Examples

```
# generate data
paras_true <- sim_hmm_paras(5, 2)
sim_data <- sim_hmm(20, paras_true, 4, 10)
# randomly generate initial values of parameters
paras_init <- sim_hmm_paras(5, 2, return_prob=FALSE)
# fit hmm
hmm_res <- hmm(sim_data$seqs, 2, paras_init)
```

---

lhmm *MMLE of LHMM*

Description

Maximum marginalized likelihood estimation of LHMM. Marginalization over latent trait is computed numerically using Guassian-Hermite quadratures from `statmod`. Optimization is performed through `optim`.

Usage

```
lhmm(action_seqs, K, paras, n_pts = 500, ...)
```

Arguments

`action_seqs`	a list of `n` action sequences
`K`	number of hidden states
`paras`	a list of elements named `para_a`, `para_b`, `para_alpha`, `para_beta`, and `para_P1`, providing initial values of model parameters
`n_pts`	number of quadrature points
`...`	additional arguments passed to `optim`

Value

A list containing the following elements

`seqs`	action sequences coded in integers
`K`	number of hidden states
`N`	number of distinct actions
`paras_init`	a list containing initial values of parameters
`paras_est`	a list containing parameter estimates

`theta_est` a vector of length `n`. estimated latent traits

`init_mllh` initial value of the marginalized likelihood function

`opt_mllh` maximized marginalized likelihood function

`opt_res` object returned by `optim`

Examples

```
# generate data
paras_true <- sim_lhmm_paras(5, 2)
sim_data <- sim_lhmm(20, paras_true, 3, 8)
# randomly initialize parameters
paras_init <- sim_lhmm_paras(5, 2)
# fit model
lhmm_res <- lhmm(sim_data$seqs, 2, paras_init)
```

`proclhmm` *proclhmm: Latent Hidden Markov Models for Response Process Data*

Description

This package provides functions for simulating from and fitting the latent hidden Markov models for response process data (Tang, 2024). It also includes functions for simulating from and fitting ordinary hidden Markov models.

Data Simulation Functions

- `sim_hmm_paras` generates parameters of HMM
- `sim_hmm` generates actions sequences from HMM.
- `sim_lhmm_paras` generates parameters of LHMM
- `sim_lhmm` generates actions sequences from LHMM.

Model Fitting Functions

- `hmm` fits HMM models. Parameters are estimated through marginalized maximum likelihood estimation.
- `lhmm` fits LHMM models. Parameters are estimated through marginalized maximum likelihood estimation.
- `compute_theta` compute MAP estimates of latent traits in LHMM.
- `find_state_seq` compute the most likely hidden state sequence.

Acknowledgment

The development of this package is supported by National Science Foundation grant DMS-2310664.

References

Tang, X. (2024) Latent Hidden Markov Models for Response Process Data. Psychometrika 89, 205-240. doi:10.1007/s11336023099381

`sim_hmm` *Simulating action sequences using HMM*

Description

`sim_hmm` generate `n` action sequences from HMM based on given parameters. The lengths of the generated sequences are simulated from a Poission distribution with mean `mean_len` and at least `min_len`.

Usage

```
sim_hmm(n, paras, min_len, mean_len, return_state = TRUE)
```

Arguments

`n`	number of action sequences to be generated
`paras`	a list containing specified HMM parameters: state-transition probability matrix (`P`), state-action probability matrix (`Q`), and initial state probability (`P1`).
`min_len`	minimum length of generated sequences
`mean_len`	mean length of generated sequences
`return_state`	logical. Whether generated hidden state sequences should be returned or not.

Value

`sim_hmm` returns a list of `n` generated action sequences if `return_state = FALSE`. If `return_state = TRUE`, it returns a list of two lists, `seqs` and `state_seqs`. `seqs` gives the generated action sequences. `state_seqs` gives the corresponding hidden state sequences.

Examples

```
paras <- sim_hmm_paras(5,2)
sim_data <- sim_hmm(20, paras, 3, 10)
```

sim_hmm_paras *generate HMM parameters*

Description

sim_hmm_paras generates logit scale parameters of HMM with K hidden states and N distinct actions from Uniform(-0.5, 0.5).

Usage

```
sim_hmm_paras(N, K, return_prob = TRUE)
```

Arguments

N	number of distinct actions
K	number of hidden states
return_prob	logical. indicates to return parameters in probability scale (TRUE, default) or logit scale.

Value

a list of three elements. If return_prob = TRUE, the element names are P1, P, and Q. If return_prob = FALSE, the element names are para_P1, para_P, and oara_Q.

Examples

```
# generate probability parameters
set.seed(12345)
paras1 <- sim_hmm_paras(5, 2)
names(paras1)

# generate parameters in the logit scale
set.seed(12345)
paras2 <- sim_hmm_paras(5, 2, return_prob = FALSE)
names(paras2)

paras1$P1
paras2$para_P1

# logit scale parameters can be transformed to probability parameters
all.equal(compute_paras_hmm(paras2$para_P, paras2$para_Q, paras2$para_P1), paras1)
```

sim_lhmm *Simulating action sequences using LHMM*

Description

sim_lhmm generate n action sequences from LHMM based on given parameters. The lengths of the generated sequences are simulated from a Poission distribution with mean mean_len and at least min_len. The latent trait is generated from standard normal.

Usage

```
sim_lhmm(n, paras, min_len, mean_len, return_state = TRUE)
```

Arguments

n	number of action sequences to be generated
paras	a list containing specified LHMM parameters: para_a, para_b, para_alpha, para_beta, and para_P1.
min_len	minimum length of generated sequences
mean_len	mean length of generated sequences
return_state	logical. Whether generated hidden state sequences should be returned or not.

Value

If return_state = TRUE, sim_hmm returns a list of three elements

seqs	a list of n generated action sequences
theta	latent traits as a vector of length n
state_seqs	a list of n hidden state sequences

If return_state = FALSE, the returned list only contains seqs and theta.

Examples

```
paras <- sim_lhmm_paras(5,2)
sim_data <- sim_lhmm(20, paras, 4, 10)
```

sim_lhmm_paras *generate LHMM parameters*

Description

sim_hmm_paras generates the parameters of LHMM with K hidden states and N distinct actions from Uniform(-0.5, 0.5).

Usage

```
sim_lhmm_paras(N, K)
```

Arguments

N	number of distinct actions
K	number of hidden states

Value

a list of five elements, para_a, para_b, para_alpha, para_beta, and para_P1.

Examples

```
paras <- sim_lhmm_paras(5, 2)
paras
```

Index