Package 'HMMHSMM'

December 18, 2025

Type Package

Title Inference and Estimation of Hidden Markov Models and Hidden
Semi-Markov Models

Version 0.1.0

Maintainer Ting Wang <ting.wang@otago.ac.nz>

Description Provides flexible maximum likelihood estimation and inference for Hidden
Markov Models (HMMs) and Hidden Semi-Markov Models (HSMMs), as well as the
underlying systems in which they operate. The package supports a wide range
of observation and dwell-time distributions, offering a flexible modelling
framework suitable for diverse practical data. Efficient implementations of
the forward-backward and Viterbi algorithms are provided via 'Rcpp' for enhanced
computational performance. Additional functionality includes model simulation,
residual analysis, non-initialised estimation, local and global decoding,
calculation of diverse information criteria, computation of confidence intervals
using parametric bootstrap methods, numerical covariance matrix estimation, and
comprehensive visualisation functions for interpreting the data-generating
processes inferred from the models. Methods follow standard approaches described
by Guédon (2003) <doi:10.1198/1061860032030>,
Zucchini and MacDonald (2009, ISBN:9781584885733), and
O'Connell and Højsgaard (2011) <doi:10.18637/jss.v039.i04>.

License GPL-3

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.0), evd, extRemes, stats, MASS, mnormt,
grDevices, graphics, utils

LinkingTo Rcpp

Depends R (>= 3.5.0)

RoxygenNote 7.3.2

NeedsCompilation yes

Author Aimee Cody [aut],
Ting Wang [cre, ctb] (Research Supervisor)

Repository CRAN

Date/Publication 2025-12-18 14:20:10 UTC

Contents

---

conditionalreturnsHMMgev

Plot Conditional Return Levels from GEV-HMM

Description

Computes and plots conditional return levels from a fitted Hidden Markov Model (HMM) with generalized extreme value (GEV) state-dependent distributions. Bootstrap confidence intervals are included. Return levels are aggregated over user-defined time periods.

Usage

```
conditionalreturnsHMMgev(x, HMM, return_periods = c(50, 100, 200, 500),
                         varcov = NULL, B = 10000, level = 0.95,
                         time_structure = NULL,
                         plot_title = "Return Levels Over Time",
                         save_plot = FALSE, filename = NULL,
                         width = 12, height = 8, dpi = 300,
                         verbose = TRUE)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	A fitted HMM object (output from [findmleHMM](#)), with state-dependent GEV parameters `loc`, `scale`, `shape`, and estimated transition probabilities.
return_periods	Numeric vector. Return periods (in years) for which conditional return levels are computed. Default is `c(50, 100, 200, 500)`.
varcov	Optional variance–covariance matrix of parameter estimates. If `NULL`, computed internally via [HMMVarianceMatrix](#) with `obsdist = "gev"`.
B	Integer. Number of bootstrap replicates used for confidence intervals. Default is 10000.
level	Numeric between 0 and 1. Confidence level for intervals. Default is 0.95.
time_structure	Optional list describing the time scale. Includes `unit`, `observations_per_unit`, and optionally `start_point`, `end_point`. Supported units: "year", "day", "hour", "week", "month", or `"custom"` with `conversion_factor` and `unit_name`.
plot_title	Character string. Title for the plot. Default is "Return Levels Over Time".
save_plot	Logical. If `TRUE`, the plot is saved to file. Default is `FALSE`.
filename	Character string or NULL. Filename for saved plot (if `save_plot = TRUE`). Must be specified when `save_plot = TRUE`. Default is `NULL`.
width	Numeric. Width of saved plot in inches. Default is 12.
height	Numeric. Height of saved plot in inches. Default is 8.
dpi	Integer. Resolution of saved plot in dots per inch. Default is 300.
verbose	Logical. If `TRUE`, progress messages are printed to the console. Default is `TRUE`.

Details

For each state, the conditional return level for a given return period is computed from the fitted GEV parameters. Uncertainty is estimated by parametric bootstrap using draws from a multivariate normal approximation of parameter estimates. Observation-level return levels are aggregated over time periods defined by `time_structure` (e.g., years, weeks, months).

Value

Invisibly returns a list containing:

return_levels_states	List of matrices, one per return period, containing estimated return levels and confidence intervals for each state (3 rows: estimate, lower CI, upper CI).
return_levels_time	List of matrices, one per return period, mapping state-specific return levels to the time series.
return_periods	Numeric vector of return periods used.
time_info	List containing time axis information and labels.

A time-series plot is also produced, showing aggregated conditional return levels with bootstrap confidence intervals.

Author(s)

Aimee Cody

See Also

findmleHMM, for fitting HMMs. generateHMM, for simulating HMM data. globaldecodeHMM, for decoding the most likely state sequence. HMMVarianceMatrix, for variance–covariance estimation.

Examples

```
set.seed(123)
J <- 2
Pi <- matrix(c(0.9, 0.1,
               0.2, 0.8), nrow = J, byrow = TRUE)
delta <- c(0.5, 0.5)
obspar <- list(loc = c(0, 5), scale = c(1, 2), shape = c(0.1, -0.1))

sim <- generateHMM(n = 200, J = J, obsdist = "gev",
                    obspar = obspar, Pi = Pi, delta = delta)

HMM_fit <- findmleHMM(x = sim$x, J = J, obsdist = "gev",
                       obspar = obspar, Pi = Pi, delta = delta)

time_struct <- list(unit = "week", observations_per_unit = 10, start_point = 1)


result <- conditionalreturnsHMMgev(x = sim$x, HMM = HMM_fit,
                                    return_periods = c(50, 100, 200, 500),
                                    B = 1000,
                                    time_structure = time_struct,
                                    plot_title = "GEV-HMM Conditional Return Levels")

result <- conditionalreturnsHMMgev(x = sim$x, HMM = HMM_fit,
                                    return_periods = c(50, 100),
                                    B = 1000,
                                    time_structure = time_struct,
                                    save_plot = TRUE,
                                    filename = tempfile(fileext = ".png"))
```

---

conditionalreturnsHSMMgev
Plot Conditional Return Levels from GEV-HSMM

Description

Computes and plots conditional return levels from a fitted Hidden Semi-Markov Model (HSMM) with generalized extreme value (GEV) state-dependent distributions. Bootstrap confidence intervals are included. Return levels are aggregated over user-defined time periods.

Usage

```
conditionalreturnsHSMMgev(x, HSMM, dwelldist, M = NA,
                          return_periods = c(50, 100, 200, 500),
                          varcov = NULL, B = 10000, level = 0.95,
                          time_structure = NULL, shift = FALSE,
                          plot_title = "Return Levels Over Time",
                          save_plot = FALSE, filename = NULL,
                          width = 12, height = 8, dpi = 300,
                          verbose = TRUE)
```

Arguments

x	Numeric vector. The observed data sequence.
HSMM	A fitted HSMM object (output from [findmleHSMM](#)), with state-dependent GEV parameters loc, scale, shape, dwell parameters, and estimated transition probabilities.
dwelldist	Character string. The dwell-time distribution used in the HSMM, e.g. "pois", "nbinom", or "betabinom".
M	Optional integer. Truncation parameter for dwell-time distribution. Default is NA (no truncation).
return_periods	Numeric vector. Return periods (in years) for which conditional return levels are computed. Default is c(50, 100, 200, 500).
varcov	Optional variance–covariance matrix of parameter estimates. If NULL, computed internally via [HSMMVarianceMatrix](#) with obsdist = "gev".
B	Integer. Number of bootstrap replicates used for confidence intervals. Default is 10000.
level	Numeric between 0 and 1. Confidence level for intervals. Default is 0.95.
time_structure	Optional list describing the time scale. Includes unit, observations_per_unit, and optionally start_point, end_point. Supported units: "year", "day", "hour", "week", "month", or "custom" with conversion_factor and unit_name.
shift	Logical. If TRUE, uses shifted dwell-time distributions. Default is FALSE.
plot_title	Character string. Title for the plot. Default is "Return Levels Over Time".
save_plot	Logical. If TRUE, the plot is saved to file. Default is FALSE.
filename	Character string or NULL. Filename for saved plot (if save_plot = TRUE). Must be specified when save_plot = TRUE. Default is NULL.
width	Numeric. Width of saved plot in inches. Default is 12.
height	Numeric. Height of saved plot in inches. Default is 8.
dpi	Integer. Resolution of saved plot in dots per inch. Default is 300.
verbose	Logical. If TRUE, progress messages are printed to the console. Default is TRUE.

Details

For each state, the conditional return level for a given return period is computed from the fitted GEV parameters. Uncertainty is estimated by parametric bootstrap using draws from a multivariate normal approximation of parameter estimates. Observation-level return levels are aggregated over time periods defined by time_structure (e.g., years, weeks, months).

Value

Invisibly returns a list containing:

return_levels_states

List of matrices, one per return period, containing estimated return levels and confidence intervals for each state (3 rows: estimate, lower CI, upper CI).

return_levels_time

List of matrices, one per return period, mapping state-specific return levels to the time series.

return_periods Numeric vector of return periods used.

time_info   List containing time axis information and labels.

A time-series plot is also produced, showing aggregated conditional return levels with bootstrap confidence intervals.

Author(s)

Aimee Cody

See Also

[findmleHSMM](), for fitting HSMMs. [generateHSMM](), for simulating HSMM data. [globaldecodeHSMM](), for decoding the most likely state sequence. [HSMMVarianceMatrix](), for variance–covariance estimation.

Examples

```
set.seed(123)

J <- 2
Pi <- matrix(c(0, 1,
               1, 0), nrow = J, byrow = TRUE)
delta <- c(0.5, 0.5)

obspar <- list(
  loc   = c(0, 5),
  scale = c(1, 2),
  shape = c(0.1, -0.1)
)

dwellpar <- list(lambda = c(5, 10))

sim <- generateHSMM(
  n = 200, J = J, obsdist = "gev", dwelldist = 'pois',
  obspar = obspar, dwellpar = dwellpar, Pi = Pi, delta = delta
)

HSMM_fit <- findmleHSMM(
  x = sim$x, J = J, obsdist = "gev", dwelldist = 'pois',
  obspar = obspar, dwellpar = dwellpar, Pi = Pi, delta = delta
)
```

```
time_struct <- list(
  unit = "week", observations_per_unit = 10, start_point = 1
)


result <- conditionalreturnsHSMMgev(
  x = sim$x, HSMM = HSMM_fit,
  return_periods = c(50, 100, 200, 500),
  dwelldist = "pois", B = 1000,
  time_structure = time_struct,
  plot_title = "GEV-HSMM Conditional Return Levels"
)

result <- conditionalreturnsHSMMgev(
  x = sim$x, HSMM = HSMM_fit,
  return_periods = c(50, 100),
  dwelldist = "pois", B = 1000,
  time_structure = time_struct,
  save_plot = TRUE,
  filename = tempfile(fileext = ".png")
)
```

---

confintHMM   *Bootstrap Confidence Intervals for Hidden Markov Models*

Description

Computes bootstrap confidence intervals for the parameters of a fitted Hidden Markov Model (HMM). The function resamples data by simulating from the fitted model, refits the HMM repeatedly to the simulated data, and constructs confidence intervals for the state-dependent parameters and the transition probability matrix.

Usage

```
confintHMM(x, HMM, obsdist, level = 0.95, B = 100, EM = FALSE,
           verbose = TRUE, seed = NULL, ...)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	A list containing estimated HMM parameters, typically returned by findmleHMM. Must include estimate$Pi, estimate$delta, and state-dependent observation parameters.
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".

level	Numeric. Confidence level for the intervals. Default is `0.95`.
B	Integer. Number of bootstrap replicates to perform. Default is `100`.
EM	Logical. Whether to use the EM algorithm during refitting. Default is `FALSE`.
verbose	Logical. If `TRUE`, progress messages are printed to the console. Default is `TRUE`.
seed	Integer or `NULL`. Random seed for reproducibility. Default is `NULL`.
...	Any other parameters that may need to be passed to `findmleHMM`, for example `tol` or `maxiter`.

Details

The function uses a parametric bootstrap procedure. Data are simulated from the fitted HMM using `generateHMM`, and parameters are re-estimated with `findmleHMM`. Confidence intervals are then obtained from the empirical quantiles of the bootstrap replicates. When `verbose = TRUE`, the function reports progress. The function will stop if too many failed bootstrap attempts occur.

Value

A list with three elements:

obspar_CI	A data frame with columns: `Parameter`, `Estimate`, `Lower`, and `Upper`, giving bootstrap confidence intervals for the observation distribution parameters.
Pi_CI	A list containing: `Estimate`, the estimated transition matrix from the fitted model; `Lower`, the lower bounds of the bootstrap confidence intervals for each transition probability; `Upper`, the corresponding upper bounds.
bootstrap_samples	
	A list containing: `obspar`, matrix of bootstrap samples for observation parameters; `Pi`, array of bootstrap samples for transition matrix; `n_successful`, number of successful bootstrap iterations; `n_attempts`, total number of attempts made.

Author(s)

Aimee Cody

See Also

`findmleHMM`, for estimating HMM parameters from data. `generateHMM`, for simulating HMM data. `confintHSMM`, computing bootstrapped confidence intervals for a hidden semi-Markov model.

Examples

```
J <- 2
Pi <- matrix(c(0.9, 0.1,
               0.2, 0.8), nrow = 2, byrow = TRUE)
obspar <- list(lambda = c(2, 7))
sim_data <- generateHMM(n = 1000, J = J, obsdist = "pois", obspar = obspar, Pi = Pi)
HMM_fit <- findmleHMM(J = J, x = sim_data$x, obsdist = "pois",
                      obspar = obspar, Pi = Pi, EM = FALSE)
```

```
ci <- confintHMM(x = sim_data$x, HMM = HMM_fit, obsdist = "pois",
                 B = 50, level = 0.9)
ci$obspar_CI
ci$Pi_CI
ci_silent <- confintHMM(x = sim_data$x, HMM = HMM_fit, obsdist = "pois",
                        B = 50, level = 0.9, verbose = FALSE)
```

---

confintHSMM   *Bootstrap Confidence Intervals for Hidden Semi-Markov Models*

Description

Computes bootstrap confidence intervals for the parameters of a fitted Hidden Semi-Markov Model
(HSMM). The function resamples data by simulation from the fitted model, refits the HSMM repeat-
edly, and constructs confidence intervals for the state-dependent observation parameters, dwell-time
distribution parameters, and the transition probability matrix.

Usage

```
confintHSMM(x, HSMM, obsdist, dwelldist, level = 0.95, B = 100, M = NA,
            shift = FALSE, maxiter = 100, tol = 1e-5, verbose = TRUE, seed = NULL)
```

Arguments

x	Numeric vector. The observed data sequence.
HSMM	A list containing estimated HSMM parameters, typically returned by [findmleHSMM]. Must include the transition matrix Pi, observation parameters, and dwell-time parameters.
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. Dwell-time distribution. Supported distributions: "pois", "nbinom", "betabinom".
level	Numeric. Confidence level for the intervals. Default is 0.95.
B	Integer. Number of bootstrap replicates to perform. Default is 100.
M	Integer. Maximum dwell time to consider for semi-Markov states. If NA, defaults to min(length(x), 1000).
shift	Logical. Indicates whether the dwell-time distribution includes a user-specified shift parameter. If FALSE, a shift of 1 is applied automatically. Defaults to FALSE.
maxiter	Integer. Maximum number of EM iterations for each bootstrap refit. Default is 100.
tol	Numeric. Convergence tolerance for the EM algorithm during bootstrap refits. Default is 1e-5.
verbose	Logical. If TRUE, progress messages are printed to the console. Default is TRUE.
seed	Integer or NULL. Random seed for reproducibility. Default is NULL.

Details

The function uses a parametric bootstrap procedure. Data are simulated from the fitted HSMM using
[generateHSMM](#), and parameters are re-estimated with [findmleHSMM](#). Confidence intervals are then
obtained from the empirical quantiles of the bootstrap replicates. When verbose = TRUE, progress
is reported. Replicates that fail to converge are retried until B successful fits are obtained.

Value

A list with four elements:

obspar_CI	A data frame with columns: Parameter, Estimate, Lower, and Upper, giving bootstrap confidence intervals for the observation distribution parameters.
dwellpar_CI	A data frame with columns: Parameter, Estimate, Lower, and Upper, giving bootstrap confidence intervals for the dwell-time distribution parameters.
Pi_CI	A list containing: Estimate, the estimated transition matrix from the fitted model; Lower, the lower bounds of the bootstrap confidence intervals for each transition probability; Upper, the corresponding upper bounds.
bootstrap_samples	
	A list containing: obspar, matrix of bootstrap samples for observation parameters; dwellpar, matrix of bootstrap samples for dwell-time parameters; Pi, array of bootstrap samples for transition matrix; n_successful, number of successful bootstrap iterations; n_attempts, total number of attempts made.

Author(s)

Aimee Cody

See Also

[findmleHSMM](#), for estimating HSMM parameters from data. [generateHSMM](#), for simulating HSMM
data. [confintHMM](#), computing bootstrapped confidence intervals for a hidden Markov model.

Examples

```
J <- 2
Pi <- matrix(c(0, 1,
               1, 0), nrow = 2, byrow = TRUE)

obspar <- list(mean = c(4, 3), sd = c(1, 1))

dwellpar <- list(lambda = c(5, 7))

sim_data <- generateHSMM(n = 1000, J = J, obsdist = "norm",
                         dwelldist = "pois", obspar = obspar,
                         dwellpar = dwellpar, Pi = Pi)

HSMM_fit <- findmleHSMM(x = sim_data$x, J = J, M = 100,
                        obsdist = "norm", dwelldist = "pois",
                        obspar = obspar, dwellpar = dwellpar, Pi = Pi)
```

```
ci <- confintHSMM(x = sim_data$x, HSMM = HSMM_fit,
                  obsdist = "norm", dwelldist = "pois",
                  B = 50, M = 50, level = 0.9)

ci$obspar_CI
ci$dwellpar_CI
ci$Pi_CI

ci_silent <- confintHSMM(x = sim_data$x, HSMM = HSMM_fit,
                         obsdist = "norm", dwelldist = "pois",
                         B = 50, M = 50, level = 0.9, verbose = FALSE)
```

---

daily_maxima | *Daily maxima of geomagnetic rate-of-change data from Eskdalemuir Magnetic Observatory*

Description

A dataset containing daily block maxima of the horizontal geomagnetic field rate-of-change at the Eskdalemuir Magnetic Observatory, Scotland, United Kingdom. The data reflect the daily extremes of geomagnetic variability, serving as indicators of geomagnetic disturbance over the period 1999–2022.

Usage

```
data(daily_maxima)
```

Format

A numeric vector of daily maxima of

$$R_1(t) = \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2},$$

where X_t and Y_t denote the horizontal north and east components of the magnetic field, respectively.

Details

These daily maxima capture short-term geomagnetic disturbances while maintaining a sufficiently fine temporal resolution to observe rapid changes. Like the weekly maxima, they are naturally modelled using the Generalised Extreme Value (GEV) distribution, with location, scale, and shape parameters μ, σ, and ξ.

Source

Eskdalemuir Magnetic Observatory, British Geological Survey, United Kingdom.

Derived and aggregated for use in **HMMHSMM**.

Examples

```
data(daily_maxima)
plot(daily_maxima, type = "l", main = "Daily maxima of geomagnetic variability",
     ylab = "Daily maximum rate of change (nT/min)", xlab = "Day index")
```

---

exceedanceplotHMMgev   *Plot Exceedance Probabilities from GEV-HMM*

Description

Computes and plots exceedance probabilities of a given threshold from a fitted Hidden Markov Model (HMM) with generalized extreme value (GEV) state-dependent distributions. Bootstrap confidence intervals are included. Probabilities are aggregated over user-defined time periods.

Usage

```
exceedanceplotHMMgev(x, HMM, threshold, varcov = NULL, B = 10000, level = 0.95,
                     time_structure = NULL,
                     plot_title = "Exceedance Probabilities Over Time",
                     save_plot = FALSE, filename = NULL,
                     width = 12, height = 8, dpi = 300,
                     verbose = TRUE)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	A fitted HMM object (output from [findmleHMM](#)), with state-dependent GEV parameters loc, scale, shape, and estimated transition probabilities.
threshold	Numeric. The exceedance threshold. Probabilities of exceeding this value are computed.
varcov	Optional variance–covariance matrix of parameter estimates. If NULL, computed internally via [HMMVarianceMatrix](#) with obsdist = "gev".
B	Integer. Number of bootstrap replicates used for confidence intervals. Default is 10000.
level	Numeric between 0 and 1. Confidence level for intervals. Default is 0.95.
time_structure	Optional list describing the time scale. Includes unit, observations_per_unit, and optionally start_point, end_point. Supported units: "year", "day", "hour", "week", "month", or "custom" with conversion_factor and unit_name.
plot_title	Character string. Title for the plot. Default is "Exceedance Probabilities Over Time".
save_plot	Logical. If TRUE, the plot is saved to file. Default is FALSE.
filename	Character string or NULL. Filename for saved plot (if save_plot = TRUE). Must be specified when save_plot = TRUE. Default is NULL.
width	Numeric. Width of saved plot in inches. Default is 12.

height	Numeric. Height of saved plot in inches. Default is 8.
dpi	Integer. Resolution of saved plot in dots per inch. Default is 300.
verbose	Logical. If TRUE, progress messages are printed to the console. Default is TRUE.

Details

For each state, the exceedance probability of threshold is computed from the fitted GEV parameters. Uncertainty is estimated by parametric bootstrap using draws from a multivariate normal approximation of parameter estimates. Observation-level probabilities are aggregated over time periods defined by time_structure (e.g., years, months, weeks). When verbose = TRUE, progress messages are displayed during computation.

Value

Invisibly returns a list containing:

exceedance_probs

Matrix of exceedance probabilities over time (3 rows: estimate, lower CI, upper CI).

time_info	List containing time axis information and labels.
threshold	The threshold value used.

A time-series plot is also produced, showing aggregated exceedance probabilities with confidence intervals.

Author(s)

Aimee Cody

See Also

findmleHMM, for fitting HMMs. generateHMM, for simulating HMM data. globaldecodeHMM, for decoding the most likely state sequence. HMMVarianceMatrix, for variance–covariance estimation.

Examples

```
set.seed(99)
J <- 2
Pi <- matrix(c(0.9, 0.1,
               0.2, 0.8), nrow = J, byrow = TRUE)
delta <- c(0.5, 0.5)
obspar <- list(loc = c(0, 5), scale = c(1, 2), shape = c(0.1, -0.1))

sim <- generateHMM(n = 200, J = J, obsdist = "gev",
                   obspar = obspar, Pi = Pi, delta = delta)

HMM_fit <- findmleHMM(x = sim$x, J = J, obsdist = "gev",
                      obspar = obspar, Pi = Pi, delta = delta)

time_struct <- list(unit = "week", observations_per_unit = 10, start_point = 1)
```

```
result <- exceedanceplotHMMgev(x = sim$x, HMM = HMM_fit,
                               threshold = 20, B = 10000,
                               time_structure = time_struct,
                               plot_title = "GEV-HMM Exceedance Probabilities")

result <- exceedanceplotHMMgev(x = sim$x, HMM = HMM_fit,
                               threshold = 20, B = 10000,
                               time_structure = time_struct,
                               save_plot = TRUE,
                               filename = tempfile(fileext = ".png"),
                               verbose = FALSE)
```

---

exceedanceplotHSMMgev   *Plot Exceedance Probabilities from GEV-HSMM*

Description

Computes and plots exceedance probabilities of a given threshold from a fitted Hidden Semi-Markov Model (HSMM) with generalized extreme value (GEV) state-dependent distributions. Bootstrap confidence intervals are included. Probabilities are aggregated over user-defined time periods.

Usage

```
exceedanceplotHSMMgev(x, HSMM, threshold, dwelldist, M = NA,
                      varcov = NULL, B = 10000, level = 0.95,
                      time_structure = NULL, shift = FALSE,
                      plot_title = "Exceedance Probabilities Over Time",
                      save_plot = FALSE, filename = NULL,
                      width = 12, height = 8, dpi = 300,
                      verbose = TRUE)
```

Arguments

x	Numeric vector. The observed data sequence.
HSMM	A fitted HSMM object (output from [findmleHSMM](#)), with state-dependent GEV parameters loc, scale, shape, dwell parameters, and estimated transition probabilities.
threshold	Numeric. The exceedance threshold. Probabilities of exceeding this value are computed.
dwelldist	Character string. The dwell-time distribution used in the HSMM, e.g. "pois", "nbinom", or "betabinom".
M	Optional integer. Truncation parameter for dwell-time distribution. Default is NA (no truncation).

varcov	Optional variance–covariance matrix of parameter estimates. If NULL, computed internally via HSMMVarianceMatrix with obsdist = "gev".
B	Integer. Number of bootstrap replicates used for confidence intervals. Default is 10000.
level	Numeric between 0 and 1. Confidence level for intervals. Default is 0.95.
time_structure	Optional list describing the time scale. Includes unit, observations_per_unit, and optionally start_point, end_point. Supported units: "year", "day", "hour", "week", "month", or "custom" with conversion_factor and unit_name.
shift	Logical. If TRUE, uses shifted dwell-time distributions. Default is FALSE.
plot_title	Character string. Title for the plot. Default is "Exceedance Probabilities Over Time".
save_plot	Logical. If TRUE, the plot is saved to file. Default is FALSE.
filename	Character string or NULL. Filename for saved plot (if save_plot = TRUE). Must be specified when save_plot = TRUE. Default is NULL.
width	Numeric. Width of saved plot in inches. Default is 12.
height	Numeric. Height of saved plot in inches. Default is 8.
dpi	Integer. Resolution of saved plot in dots per inch. Default is 300.
verbose	Logical. If TRUE, progress messages are printed to the console. Default is TRUE.

Details

For each state, the exceedance probability of threshold is computed from the fitted GEV parameters. Uncertainty is estimated by parametric bootstrap using draws from a multivariate normal approximation of parameter estimates. Observation-level probabilities are aggregated over time periods defined by time_structure (e.g., years, months, weeks). When verbose = TRUE, progress messages are displayed during computation.

Value

Invisibly returns a list containing:

exceedance_probs

      Matrix of exceedance probabilities over time (3 rows: estimate, lower CI, upper CI).

time_info	List containing time axis information and labels.
decoded_states	Vector of decoded hidden states for each time point.
time_unit	Number of observations per aggregation period.
threshold	The threshold value used.

A time-series plot is also produced, showing aggregated exceedance probabilities with confidence intervals.

Author(s)

Aimee Cody

See Also

findmleHSMM, for fitting HSMMs. generateHSMM, for simulating HSMM data. globaldecodeHSMM, for decoding the most likely state sequence. HSMMVarianceMatrix, for variance–covariance estimation.

Examples

```
set.seed(123)

J <- 2
Pi <- matrix(c(0, 1,
               1, 0), nrow = J, byrow = TRUE)
delta <- c(0.5, 0.5)

obspar <- list(
  loc   = c(0, 5),
  scale = c(1, 2),
  shape = c(0.1, -0.1)
)

dwellpar <- list(lambda = c(5, 10))

sim <- generateHSMM(
  n = 200, J = J, obsdist = "gev", dwelldist = "pois",
  obspar = obspar, dwellpar = dwellpar, Pi = Pi, delta = delta
)

HSMM_fit <- findmleHSMM(
  x = sim$x, J = J, obsdist = "gev", dwelldist = "pois",
  obspar = obspar, dwellpar = dwellpar, Pi = Pi, delta = delta
)

time_struct <- list(
  unit = "week", observations_per_unit = 10, start_point = 1
)


result <- exceedanceplotHSMMgev(
  x = sim$x, HSMM = HSMM_fit,
  threshold = 20, dwelldist = "pois",
  B = 10000, time_structure = time_struct,
  plot_title = "GEV-HSMM Exceedance Probabilities"
)

result <- exceedanceplotHSMMgev(
  x = sim$x, HSMM = HSMM_fit,
  threshold = 20, dwelldist = "pois",
  B = 10000, time_structure = time_struct,
  save_plot = TRUE,
  filename = tempfile(fileext = ".png"),
  verbose = FALSE
)
```

---

findmleHMM   *Maximum Likelihood Estimation for Hidden Markov Models*

Description

Estimates the parameters of a Hidden Markov Model (HMM) using either the EM-based semi-Markov parameterisation or direct numerical maximization of the likelihood.

Usage

```
findmleHMM(J, x, obsdist, obspar, Pi, EM = FALSE, verbose = TRUE, seed = NULL, ...)
```

Arguments

J	Integer. The number of hidden states in the HMM. Must be greater than 1.
x	Numeric vector. The observed data sequence.
obsdist	Character string. The observation distribution. Supported distributions are: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
obspar	List. Initial parameters for the observation distribution. Required parameters vary by distribution:

- norm: mean, sd
- pois: lambda
- weibull: shape, scale
- zip: pi, lambda
- nbinom: size, mu
- zinb: pi, size, mu
- exp: rate
- gamma: shape, rate
- lnorm: meanlog, sdlog
- gev: loc, scale, shape
- ZInormal: mean, sd, pi
- ZIgamma: shape, rate, pi

Each parameter should be a vector of length J with values for each state.

Pi	Matrix. The J x J transition probability matrix between states. Rows must sum to 1.
EM	Logical. If TRUE, uses an EM-based semi-Markov approximation to estimate the HMM parameters. If FALSE, maximizes the likelihood directly using nlm. Defaults to FALSE.
verbose	Logical. If TRUE, progress messages are printed to the console when EM = TRUE. Default is TRUE.
seed	Integer or NULL. Random seed for reproducibility. Default is NULL.
...	Further arguments to be passed in the case of EM=TRUE, such as maxiter and tol.

Details

This function fits a Hidden Markov Model to a sequence of observations using either:

- An EM-based approximation via a semi-Markov model when `EM = TRUE`
- Direct numerical maximization of the likelihood using `nlm` when `EM = FALSE`

The EM-based approach initializes dwell-time probabilities from the diagonal of `Pi` and estimates the HMM via the corresponding semi-Markov model. After fitting, transition probabilities are adjusted to incorporate self-transition probabilities. Supported observation distributions include normal, Poisson, Weibull, zero-inflated Poisson (ZIP), negative binomial, zero-inflated negative binomial (ZINB), exponential, gamma, log-normal, generalized extreme value (GEV), zero-inflated normal, and zero-inflated gamma. Parameter requirements vary by distribution. When `verbose = TRUE` and `EM = TRUE`, progress messages from the EM algorithm are displayed.

Value

A list containing:

`estimate`	List of estimated HMM parameters, including state-dependent observation parameters and transition probabilities.
`loglik`	The maximized log-likelihood value.
`AIC`	The Akaike Information Criterion for the fitted model.
`BIC`	Bayesian Information Criteria for the fitted model.
`hessian`	Optional. The Hessian matrix at the maximum likelihood estimates (returned if `EM = FALSE`).

Author(s)

Aimee Cody

See Also

[generateHMM](#) for simulating HMM data. [findmleHSMM](#) for EM-based semi-Markov estimation.

Examples

```
J <- 3
Pi <- matrix(c(0.8, 0.15, 0.05,
               0.1, 0.7, 0.2,
               0.2, 0.3, 0.5), nrow = 3, byrow = TRUE)

obspar <- list(
  mean = c(-2, 0, 3),
  sd = c(0.5, 1, 1.5)
)

x <- generateHMM(n = 200, J = J, obsdist = "norm",
                       obspar = obspar, Pi = Pi)$x
```

```
fit <- findmleHMM(J = J, x = x, obsdist = "norm",
                    obspar = obspar, Pi = Pi, EM = FALSE)

fit$estimate
fit$loglik
fit$AIC

fit_em <- findmleHMM(J = J, x = x, obsdist = "norm",
                       obspar = obspar, Pi = Pi, EM = TRUE, verbose = FALSE)
```

---

findmleHMMnostarting   *Multiple Initialization Maximum Likelihood Estimation for Hidden*
                        Markov Models

Description

Fits a Hidden Markov Model (HMM) by repeatedly initializing observation and transition parameters and selecting the fit with the highest log-likelihood. This approach helps avoid convergence to poor local optima. For the generalized extreme value (GEV) distribution, starting values are generated from repeated maximum likelihood fits on random data subsets.

Usage

```
findmleHMMnostarting(J, x, obsdist, no.initials = 50, EM = FALSE,
                       verbose = TRUE, seed = NULL, ...)
```

Arguments

J	Integer. The number of hidden states in the HMM. Must be strictly greater than 1.
x	Numeric vector. The observed data sequence.
obsdist	Character string. The observation distribution. Supported distributions are: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
no.initials	Integer. The number of random initializations to attempt. Defaults to 50.
EM	Logical. If TRUE, uses an EM-based semi-Markov approximation for estimation. If FALSE, maximizes the likelihood directly using nlm. Defaults to FALSE.
verbose	Logical. If TRUE, progress messages are printed to the console. Default is TRUE.
seed	Integer or NULL. Random seed for reproducibility. Default is NULL.
...	Further arguments to be passed to findmleHMM in the case of EM=TRUE.

Details

This function automates multiple trials of `findmleHMM` with randomized starting values, returning the fit that achieves the highest log-likelihood.

- For most observation distributions, starting values are generated via `clusterHMM`.

- For the GEV distribution, starting values are drawn from repeated fits of `evd::fgev` on random data segments. Up to 20,000 attempts are made, and a warning is issued if fewer than 1000 valid estimates are obtained.

During each iteration:

1. Observation parameters are perturbed slightly to encourage exploration.

2. A transition matrix `Pi` is drawn from a random uniform distribution with added self-transition bias.

3. The HMM is estimated via `findmleHMM`.

4. If the resulting log-likelihood exceeds the current best, the model is updated.

At the end of all iterations, the best-fitting model is returned. When `verbose = TRUE`, iteration numbers and error messages are displayed during the fitting process.

Value

A list corresponding to the best fit across all initializations, containing:

estimate	List of estimated HMM parameters, including state-dependent observation parameters and transition probabilities.
loglik	The maximized log-likelihood value.
AIC	The Akaike Information Criterion for the fitted model.
BIC	Bayesian Information Criteria for the fitted model.
hessian	Optional. The Hessian matrix at the maximum likelihood estimates (returned if `EM = FALSE`).

Author(s)

Aimee Cody

See Also

`findmleHMM` for fitting an HMM with user-supplied starting values. `generateHMM` for simulating HMM data. `findmleHSMMnostarting` for the non-initialised estimation of hidden semi-Markov models.

Examples

```
set.seed(123)
J <- 3
Pi <- matrix(c(0.7, 0.2, 0.1,
               0.1, 0.8, 0.1,
               0.2, 0.3, 0.5), nrow = 3, byrow = TRUE)
obspar <- list(mean = c(-2, 0, 3),
               sd   = c(0.5, 1, 1.5))
x <- generateHMM(n = 200, J = J, Pi = Pi, obsdist = "norm", obspar = obspar)$x


fit <- findmleHMMnostarting(J = J, x = x, obsdist = "norm",
                            no.initials = 30)

fit$loglik
fit$estimate

fit_silent <- findmleHMMnostarting(J = J, x = x, obsdist = "norm",
                                   no.initials = 30, verbose = FALSE)
```

---

findmleHSMM   *Maximum Likelihood Estimation for Hidden Semi-Markov Models*

Description

Estimates the parameters of a Hidden Semi-Markov Model (HSMM) using the EM algorithm with specified observation and dwell-time distributions.

Usage

```
findmleHSMM(x, J, M = NA, obsdist, dwelldist, obspar, dwellpar,
            Pi, delta = NULL, maxiter = 100, tol = 1e-05, shift = FALSE,
            verbose = TRUE, seed = NULL)
```

Arguments

x	Numeric vector. The observed data sequence.
J	Integer. The number of hidden states in the model. Must be greater than 1.
M	Integer. Maximum dwell time to consider for semi-Markov states. Defaults to `min(length(x), 1000)` if NA.
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. Dwell-time distribution. Supported distributions: "pois", "nbinom", "betabinom", "geom".

obspar   List. Parameters for the observation distribution. Each parameter must be a
                vector of length J, with values for each state. Required parameters:

                    • norm: mean, sd
                    • pois: lambda
                    • weibull: shape, scale
                    • zip: lambda, pi
                    • nbinom: size, mu
                    • zinb: size, mu, pi
                    • exp: rate
                    • gamma: shape, rate
                    • lnorm: meanlog, sdlog
                    • gev: loc, scale, shape
                    • ZInormal: mean, sd, pi
                    • ZIgamma: shape, rate, pi

dwellpar   List. Parameters for the dwell-time distribution. Each parameter must be a
                vector of length J, with values for each state. Required parameters:

                    • pois: lambda
                    • nbinom: size, mu
                    • betabinom: size, prob
                    • geom: prob

Pi   Matrix. The J x J transition probability matrix between states. Rows must sum
                to 1.

delta   Numeric vector of length J. Initial state distribution. If NULL, the stationary
                distribution of Pi is used.

maxiter   Integer. Maximum number of EM iterations. Defaults to 100.

tol   Numeric. Convergence tolerance for the change in log-likelihood. Defaults to
                1e-05.

shift   Logical. Indicates whether the dwell-time distribution includes a user-specified
                shift parameter. If TRUE, the distribution uses the provided shift parameter. If
                FALSE, the dwell-time distribution is automatically shifted by 1. Defaults to
                FALSE.

verbose   Logical. If TRUE, progress messages including iteration numbers and log-likelihood
                values are printed to the console. Default is TRUE.

seed   Integer or NULL. Random seed for reproducibility. Default is NULL.

Details

This function fits a Hidden Semi-Markov Model to a sequence of observations using an EM algo-
rithm. At each iteration, the algorithm:

  • Computes observation probabilities for each state
  • Calculates dwell-time probabilities and survival functions
  • Performs a forward-backward-like procedure to estimate expected sufficient statistics

• Re-estimates observation and dwell-time parameters

Supported observation distributions include normal, Poisson, Weibull, zero-inflated Poisson (ZIP), negative binomial, zero-inflated negative binomial (ZINB), exponential, gamma, log-normal, generalized extreme value (GEV), zero-inflated normal, and zero-inflated gamma.

Supported dwell-time distributions include Poisson, negative binomial, beta-binomial, and geometric.

When verbose = TRUE, the function displays log-likelihood values at each iteration and reports whether convergence was achieved.

Value

A list containing:

loglikelihoods	Numeric vector of log-likelihood values across iterations.
AIC	Akaike Information Criteria for the fitted model.
BIC	Bayesian Information Criteria for the fitted model.
delta	Numeric vector. Estimated initial state probabilities.
Pi	Matrix. Estimated J x J transition probability matrix.
dwellparameters	
	List. Estimated dwell-time distribution parameters.
observationparameters	
	List. Estimated observation distribution parameters.

Author(s)

Aimee Cody

See Also

findmleHMM for Hidden Markov Model estimation. generateHSMM for simulating HSMM data.

Examples

```
J <- 3
Pi <- matrix(c(0.0, 0.6, 0.4,
               0.5, 0.0, 0.5,
               0.3, 0.7, 0.0), nrow = J, byrow = TRUE)

obspar <- list(
  mean = c(-2, 0, 3),
  sd   = c(1, 1.5, 2)
)

dwellpar <- list(
  lambda = c(3, 5, 4)
)

sim <- generateHSMM(n = 1000, J = J, obsdist = "norm",
```

```
                          dwelldist = "pois", obspar = obspar,
                          dwellpar = dwellpar, Pi = Pi)

  fit <- findmleHSMM(x = sim$x, J = J, M = 100,
                       obsdist = "norm", dwelldist = "pois",
                       obspar = obspar, dwellpar = dwellpar,
                       Pi = Pi)

  fit$observationparameters
  fit$dwellparameters
  fit$Pi
  fit$delta
  fit$loglikelihoods

  fit_silent <- findmleHSMM(x = sim$x, J = J, M = 100,
                             obsdist = "norm", dwelldist = "pois",
                             obspar = obspar, dwellpar = dwellpar,
                             Pi = Pi, verbose = FALSE)
```

---

findmleHSMMnostarting *Fit Hidden Semi-Markov Model (HSMM) Without User-Provided*
Starting Values

Description

Fits a hidden semi-Markov model (HSMM) to a univariate time series by maximum likelihood, using multiple randomly generated initializations instead of user-provided starting parameters. The best-fitting model is selected by log-likelihood.

Usage

```
findmleHSMMnostarting(J, x, obsdist, dwelldist, M = NA, no.initials = 50,
                      verbose = TRUE, seed = NULL, ...)
```

Arguments

J	Integer, number of hidden states (must be at least 2).
x	Numeric vector, observed time series.
obsdist	Character string specifying the observation distribution. One of "pois", "norm", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string specifying the dwell-time distribution. One of "pois", "nbinom", "betabinom".
M	Integer, truncation parameter for dwell-time distribution (default NA, computed automatically).
no.initials	Integer, number of random starting values to try (default 50).

verbose	Logical, if `TRUE` (default), progress messages are printed to the console. Set to `FALSE` to suppress informational output.
seed	Integer or `NULL`. Random seed for reproducibility. Default is `NULL`.
...	Further arguments to pass to [`findmleHSMM`](#), such as `maxiter` and `tol`.

Details

This function automates parameter initialization by sampling plausible random values for state-dependent observation parameters, dwell-time parameters, initial distribution δ, and transition matrix Π. It repeatedly calls [`findmleHSMM`](#) with these initializations, and retains the model with the highest log-likelihood. When `obsdist = "gev"`, initial values are obtained via repeated calls to `evd::fgev` on random data segments until a sufficient number of estimates are collected.

Value

An object containing:

loglikelihoods	Numeric vector of log-likelihood values across iterations.
AIC	Akaike Information Criteria for the fitted model.
BIC	Bayesian Information Criteria for the fitted model.
delta	Numeric vector. Estimated initial state probabilities.
Pi	Matrix. Estimated J x J transition probability matrix.
dwellparameters	
	List. Estimated dwell-time distribution parameters.
observationparameters	
	List. Estimated observation distribution parameters.

Author(s)

Aimee Cody

See Also

[`findmleHSMM`](#), for fitting an HSMM with user-supplied initial values. [`generateHSMM`](#), for simulating HSMM data. [`findmleHMMnostarting`](#), for the non-initialised estimation of hidden Markov models.

Examples

```
set.seed(321)
J <- 3
Pi <- matrix(c(0.0, 0.6, 0.4,
               0.5, 0.0, 0.5,
               0.3, 0.7, 0.0), nrow = J, byrow = TRUE)
obspar <- list(
  mean = c(-2, 0, 3),
  sd   = c(1, 1.5, 2)
)
dwellpar <- list(
  lambda = c(3, 5, 4)
```

```
)
sim <- generateHSMM(n = 200, J = J, obsdist = "norm",
                    dwelldist = "pois", obspar = obspar,
                    dwellpar = dwellpar, Pi = Pi)
fit <- findmleHSMMnostarting(J = J, x = sim$x,
                                    obsdist = "norm", dwelldist = "pois",
                                    M = 100, no.initials = 30)
fit$observationparameters
fit$dwellparameters
fit$Pi
fit$delta
fit$loglikelihoods
```

---

generateHMM   *Generate Data from a Hidden Markov Model*

Description

Simulates observations and hidden states from a Hidden Markov Model (HMM) with specified
observation distributions.

Usage

```
generateHMM(n, J, obsdist, obspar, Pi, delta = NULL, seed = NULL)
```

Arguments

n	Integer. The number of observations to generate.
J	Integer. The number of hidden states in the model.
obsdist	Character string. The observation distribution. Supported distributions are: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
obspar	List. Parameters for the observation distribution. Required parameters vary by distribution:

- norm: mean, sd
- pois: lambda
- weibull: shape, scale
- zip: pi, lambda
- nbinom: size, mu
- zinb: pi, size, mu
- exp: rate
- gamma: shape, rate
- lnorm: meanlog, sdlog
- gev: loc, scale, shape

- ZInormal: mean, sd, pi
- ZIgamma: shape, rate, pi

Each parameter should be a vector of length J with values for each state.

Pi   Matrix. The J x J transition probability matrix between states. Rows must sum
            to 1.

delta   Numeric vector of length J. The initial state distribution. If NULL, the stationary
            distribution is computed from Pi.

seed   Integer or NULL. Random seed for reproducibility. Default is NULL.

Details

This function simulates data from a Hidden Markov Model where:

- Hidden states follow a discrete-time Markov chain with transition matrix Pi
- At each time step, observations are generated from state-dependent distributions

The function supports multiple observation distributions including normal, Poisson, Weibull, zero-inflated Poisson (ZIP), negative binomial, zero-inflated negative binomial (ZINB), exponential, gamma, log-normal, generalized extreme value (GEV), zero-inflated normal, and zero-inflated gamma. The initial state is sampled according to the initial distribution delta. If delta is not provided, it is computed as the stationary distribution of the Markov chain defined by Pi.

Value

A list containing:

x   Numeric vector of the simulated observations.

state   Numeric vector of the simulated hidden state sequence.

Author(s)

[Aimee Cody]

See Also

generateHSMM for Hidden Semi-Markov Models.

Examples

```
# Example with 3 states, normal observations
J <- 3
# HMM transition matrix
Pi <- matrix(c(0.8, 0.15, 0.05,
               0.1, 0.7,  0.2,
               0.2, 0.3,  0.5), nrow = 3, byrow = TRUE)
# Observation parameters (normal distribution)
obspar <- list(
  mean = c(-2, 0, 3),
  sd = c(0.5, 1, 1.5)
)
```

```
# Generate 200 observations
sim_data <- generateHMM(n = 200, J = J, obsdist = "norm",
                        obspar = obspar, Pi = Pi)
# View the results
head(sim_data$x)      # observations
head(sim_data$state)  # hidden states
length(sim_data$x)    # number of observations
```

---

generateHSMM   *Generate Data from a Hidden Semi-Markov Model*

Description

Simulates observations and hidden states from a Hidden Semi-Markov Model (HSMM) with specified observation and dwell time distributions.

Usage

```
generateHSMM(n, J, obsdist, dwelldist, obspar, dwellpar, Pi,
             delta = NULL, simtype = "nobs", shift = FALSE, seed = NULL)
```

Arguments

n	Integer. The number of observations to generate (if simtype = "nobs") or the number of state sequences (if simtype = "nseq").
J	Integer. The number of hidden states in the model.
obsdist	Character string. The observation distribution. Supported distributions are: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. The dwell time distribution. Supported distributions are: "pois", "nbinom", "betabinom".
obspar	List. Parameters for the observation distribution. Required parameters vary by distribution:

- norm: mean, sd
- pois: lambda
- weibull: shape, scale
- zip: pi, lambda
- nbinom: size, mu
- zinb: pi, size, mu
- exp: rate
- gamma: shape, rate
- lnorm: meanlog, sdlog
- gev: loc, scale, shape
- ZInormal: mean, sd, pi

- ZIgamma: shape, rate, pi

Each parameter should be a vector of length J with values for each state.

dwellpar List. Parameters for the dwell time distribution. Required parameters vary by distribution:

- pois: lambda, shift
- nbinom: shift, size, mu
- betabinom: size, alpha, beta, shift

Each parameter should be a vector of length J with values for each state.

Pi Matrix. The J x J transition probability matrix between states. Rows must sum to 1.

delta Numeric vector of length J. The initial state distribution. If NULL, the stationary distribution is computed from Pi.

simtype Character string. Either "nobs" (generate n observations) or "nseq" (generate n state sequences). Default is "nobs".

shift Logical. If TRUE, uses the shift parameter from dwellpar. If FALSE and no shift is provided in dwellpar, sets shift to 1 for all states. Default is FALSE.

seed Integer or NULL. Random seed for reproducibility. Default is NULL.

Details

This function simulates data from a Hidden Semi-Markov Model where:

- Hidden states follow a Markov chain with transition matrix Pi
- Each state has an associated dwell time distribution that determines how long the process remains in that state
- Observations are generated from state-dependent distributions

The function supports multiple observation distributions including normal, Poisson, Weibull, zero-inflated Poisson (ZIP), negative binomial, zero-inflated negative binomial (ZINB), exponential, gamma, log-normal, generalized extreme value (GEV), zero-inflated normal and zero-inflated gamma.

Dwell time distributions include Poisson, negative binomial, and beta-binomial, all with optional shift parameters to ensure minimum dwell times.

When simtype = "nobs", the function generates exactly n observations. When simtype = "nseq", it generates n state sequences and the total number of observations depends on the realized dwell times.

Value

A list containing:

states Numeric vector of the simulated hidden state sequence.

x Numeric vector of the simulated observations.

N Integer. The number of observations generated.

Author(s)

[Aimee Cody]

See Also

[generateHMM](#) for Hidden Markov Models.

Examples

```
# Example with 3 states, normal observations, Poisson dwell times
J <- 3
# HSMM transition matrix
Pi <- matrix(c(0.0, 0.6, 0.4,
               0.5, 0.0, 0.5,
               0.3, 0.7, 0.0), nrow = 3, byrow = TRUE)

# Observation parameters (normal distribution)
obspar <- list(
  mean = c(-2, 0, 3),
  sd = c(1, 1.5, 2)
)

# Dwell time parameters (Poisson distribution)
dwellpar <- list(
  lambda = c(3, 5, 4)
)

# Generate 100 observations
sim_data <- generateHSMM(n = 100, J = J, obsdist = "norm", dwelldist = "pois",
                         obspar = obspar, dwellpar = dwellpar, Pi = Pi)

# View the results
head(sim_data$x)        # observations
head(sim_data$states)   # hidden states
sim_data$N              # total number of observations
```

---

globaldecodeHMM   *Global Decoding for Hidden Markov Models*

Description

Computes the most likely hidden state sequence for a univariate time series using a fitted Hidden Markov Model (HMM) via global decoding (Viterbi-style).

Usage

```
globaldecodeHMM(x, HMM, obsdist)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	A list containing estimated HMM parameters, typically returned by findmleHMM. Must include estimate$Pi and state-dependent observation parameters.
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".

Details

This function implements global decoding for an HMM using the estimated parameters from a fitted model. It calculates the forward probabilities and recursively determines the most probable sequence of hidden states. Global decoding provides the single most likely state sequence, unlike local decoding which gives the most probable state at each time point independently.

Value

states	Numeric vector. The most likely hidden state at each observation, computed via global decoding.

Author(s)

Aimee Cody

See Also

findmleHMM, for estimating HMM parameters from data. generateHMM, for simulating HMM data. localdecodeHMM, for local decoding of an HMM. globaldecodeHSMM, for global decoding of a hidden semi-Markov model.

Examples

```
# Example with 3 states, normal observations
J <- 3

# HMM transition matrix
Pi <- matrix(c(0.8, 0.15, 0.05,
               0.1, 0.7, 0.2,
               0.2, 0.3, 0.5), nrow = 3, byrow = TRUE)

# Observation parameters (normal distribution)
obspar <- list(
  mean = c(-2, 0, 3),
  sd = c(0.5, 1, 1.5)
)

# Simulate HMM data
sim_data <- generateHMM(n = 200, J = J, obsdist = "norm", obspar = obspar, Pi = Pi)

# Fit HMM to simulated data
```

```
HMM_fit <- findmleHMM(J = J, x = sim_data$x, obsdist = "norm",
                      obspar = obspar, Pi = Pi, EM = FALSE)

# Compute global decoding
decoded_states <- globaldecodeHMM(x = sim_data$x, HMM = HMM_fit, obsdist = "norm")

decoded_states
```

---

globaldecodeHSMM   *Global Decoding of Hidden Semi-Markov Models*

Description

Computes the most likely hidden state sequence for a univariate time series using a fitted Hidden Semi-Markov Model (HSMM) via the Viterbi algorithm.

Usage

```
globaldecodeHSMM(x, M = NA, HSMM, obsdist, dwelldist, shift = FALSE)
```

Arguments

x	Numeric vector. The observed data sequence.
M	Integer. Maximum dwell time to consider for semi-Markov states. Defaults to `min(length(x), 1000)` if NA.
HSMM	A fitted HSMM object (output from `findmleHSMM`) containing the estimated model parameters.
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. Dwell-time distribution. Supported distributions: "pois", "nbinom", "betabinom".
shift	Logical character. Whether or not the dwell distribution has been shifted by a shift parameter. Defaults to FALSE.

Details

This function computes the global most probable state sequence (Viterbi path) for a sequence of observations under a Hidden Semi-Markov Model using the parameters contained in a fitted HSMM object. The function calculates observation log-probabilities, dwell-time log-probabilities, and forward log-probabilities to determine the Viterbi path.

Value

A numeric vector of length `length(x)`, giving the most likely hidden state at each observation.

Author(s)

Aimee Cody

See Also

findmleHSMM, for estimating HSMM parameters from data. generateHSMM, for simulating HSMM data. localdecodeHSMM, for local decoding of HSMM states. globaldecodeHMM, for global decoding hidden Markov models.

Examples

```
J <- 3
Pi <- matrix(c(0.0, 0.6, 0.4,
               0.5, 0.0, 0.5,
               0.3, 0.7, 0.0), nrow = J, byrow = TRUE)

obspar <- list(
  mean = c(4, 2, 6),
  sd   = c(1, 1, 2)
)

dwellpar <- list(
  lambda = c(3, 5, 4)
)

# Simulate HSMM data
sim <- generateHSMM(n = 1000, J = J, obsdist = "norm",
                    dwelldist = "pois", obspar = obspar,
                    dwellpar = dwellpar, Pi = Pi)

# Fit HSMM using the true parameters
HSMM_true <- findmleHSMM(x = sim$x, J = J, M = 100,
                         obsdist = "norm", dwelldist = "pois",
                         obspar = obspar, dwellpar = dwellpar,
                         Pi = Pi)

# Decode states using globaldecodeHSMM
states <- globaldecodeHSMM(x = sim$x, M = 100, HSMM = HSMM_true,
                           obsdist = "norm", dwelldist = "pois")

head(states)
```

HMMVarianceMatrix   *Variance-Covariance Matrix for Hidden Markov Models*

Description

Computes an approximate variance-covariance matrix of the parameter estimates from a fitted Hidden Markov Model (HMM) using a numerical Hessian of the log-likelihood. This function reformulates the HMM as a special case of a Hidden Semi-Markov Model (HSMM) with geometric dwell-time distributions, and calls HSMMVarianceMatrix.

Usage

```
HMMVarianceMatrix(x, HMM, obsdist, h = 1e-5, method = "central",
                  verbose = TRUE)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	A fitted HMM object (typically the output from findmleHMM), containing estimated parameters Pi, delta, and observation parameters.
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "nbinom", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
h	Numeric. Step size for finite-difference approximation of the Hessian. Default is 1e-5.
method	Character string. Method for numerical Hessian computation. Options are "central" (default, central differences) or "forward" (forward differences).
verbose	Logical, if TRUE (default), progress messages are printed to the console. Set to FALSE to suppress informational output.

Details

This function is a wrapper around HSMMVarianceMatrix, treating an HMM as a special case of an HSMM with geometric dwell-time distributions. The Hessian is computed numerically, inverted, and adjusted for positive-definiteness to provide a variance-covariance matrix for the estimated parameters.

Value

A variance-covariance matrix of the HMM parameter estimates, computed as the inverse of the observed information matrix (negative Hessian of the log-likelihood), adjusted to be positive-definite.

Author(s)

Aimee Cody

See Also

generateHMM, for simulating HMM data. HSMMVarianceMatrix, for the general hidden semi-Markov case. confintHMM, for bootstrap confidence intervals of HMM parameters.
findmleHMM, for fitting Hidden Markov Models.

Examples

```
# Example with 2-state Normal HMM
J <- 2
Pi <- matrix(c(0.9, 0.1,
               0.2, 0.8), nrow = 2, byrow = TRUE)
delta <- c(0.5, 0.5)
obspar <- list(mean = c(0, 3), sd = c(1, 1))
# Simulate data
sim_data <- generateHMM(n = 100, J = J, obsdist = "norm",
                        obspar = obspar, Pi = Pi, delta = delta)
# Fit HMM
HMM_fit <- findmleHMM(x = sim_data$x, J = J,
                      obsdist = "norm", obspar = obspar,
                      Pi = Pi, delta = delta)
# Compute variance-covariance matrix
vcov_matrix <- HMMVarianceMatrix(x = sim_data$x, HMM = HMM_fit,
                                 obsdist = "norm")
vcov_matrix
```

HSMMVarianceMatrix   *Variance-Covariance Matrix for Hidden Semi-Markov Models*

Description

Computes an approximate variance-covariance matrix of the parameter estimates from a fitted Hidden Semi-Markov Model (HSMM) using a numerical Hessian of the log-likelihood. The Hessian is inverted and adjusted to ensure positive-definiteness, providing an estimate of parameter uncertainty.

Usage

```
HSMMVarianceMatrix(x, HSMM, M = NA, obsdist, dwelldist,
                   h = 1e-5, method = "central", verbose = TRUE)
```

Arguments

x	Numeric vector. The observed data sequence.
HSMM	A fitted HSMM object (typically the output from [findmleHSMM](#)), containing estimated model parameters Pi, delta, observation parameters, and dwell-time parameters.
M	Integer. Maximum dwell time to consider for semi-Markov states. If NA, defaults to min(length(x), 1000).
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "nbinom", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. Dwell-time distribution. Supported distributions: "pois", "nbinom", "geom".

h	Numeric. Step size for finite-difference approximation of the Hessian. Default is `1e-5`.
method	Character string. Method for numerical Hessian computation. Options are `"central"` (default, central differences) or `"forward"` (forward differences).
verbose	Logical, if `TRUE` (default), progress messages are printed to the console. Set to `FALSE` to suppress informational output.

Details

This function approximates the observed Fisher information for an HSMM by numerically differentiating the log-likelihood with respect to the model parameters. The resulting Hessian is inverted to obtain a variance-covariance matrix. Since numerical Hessians can yield non-positive-definite matrices due to rounding error or flat likelihood surfaces, the eigenvalues are adjusted to enforce positive-definiteness.

Value

A variance-covariance matrix of the HSMM parameter estimates, computed as the inverse of the observed information matrix (negative Hessian of the log-likelihood), adjusted to be positive-definite.

Author(s)

Aimee Cody

See Also

[generateHSMM](), for simulating HSMM data. [findmleHSMM](), for fitting Hidden Semi-Markov Models. [HMMVarianceMatrix](), for numerically computing the variance-covariance matrix of a fitted hidden Markov model.

Examples

```
# Example with 2-state Normal HSMM
J <- 2
Pi <- matrix(c(0, 1,
               1, 0), nrow = 2, byrow = TRUE)
obspar <- list(mean = c(0, 3), sd = c(1, 1))
dwellpar <- list(lambda = c(5, 7))
# Simulate data
sim_data <- generateHSMM(n = 100, J = J, obsdist = "norm",
                         dwelldist = "pois", obspar = obspar,
                         dwellpar = dwellpar, Pi = Pi)
# Fit HSMM
HSMM_fit <- findmleHSMM(x = sim_data$x, J = J, M = 50,
                        obsdist = "norm", dwelldist = "pois",
                        obspar = obspar, dwellpar = dwellpar, Pi = Pi)
# Compute variance-covariance matrix
vcov_matrix <- HSMMVarianceMatrix(x = sim_data$x, HSMM = HSMM_fit,
                                  obsdist = "norm", dwelldist = "pois",
                                  M = 50)
vcov_matrix
```

IC *Calculate Information Criteria for HMM/HSMM Models*

Description

Computes information criteria (AIC3, AICu, HQC, BICadj) for Hidden Markov Models (HMM) or Hidden Semi-Markov Models (HSMM).

Usage

```
IC(data, model, type)
```

Arguments

`data`	Numeric vector of observations.
`model`	Fitted HMM or HSMM model object.
`type`	Character string: `"AIC3"`, `"AICu"`, `"HQC"`, or `"BICadj"`.

Details

Automatically detects HMM vs HSMM models based on `dwellparameters` presence.

Information criteria formulas:

- **AIC3:** $-2 \log L + 3k$
- **AICu:** Unbiased AIC with finite sample correction
- **HQC:** $-2 \log L + 2k \log(\log n)$
- **BICadj:** $-2 \log L + k \log n - k \log(2\pi)$

where L is likelihood, k is number of parameters, n is sample size.

Value

Numeric value of the specified information criterion.

Author(s)

Aimee Cody

Examples

```
J <- 3
Pi <- matrix(c(0.0, 0.6, 0.4,
               0.5, 0.0, 0.5,
               0.3, 0.7, 0.0), nrow = J, byrow = TRUE)

obspar <- list(
  mean = c(-2, 0, 3),
```

```
  sd   = c(1, 1.5, 2)
)

dwellpar <- list(
  lambda = c(3, 5, 4)
)

sim <- generateHSMM(n = 1000, J = J, obsdist = "norm",
                      dwelldist = "pois", obspar = obspar,
                      dwellpar = dwellpar, Pi = Pi)

fit <- findmleHSMM(x = sim$x, J = J, M = 100,
                     obsdist = "norm", dwelldist = "pois",
                     obspar = obspar, dwellpar = dwellpar,
                     Pi = Pi)

AIC3 <- IC(sim, fit, "AIC3")
AICu <- IC(sim, fit, "AICu")
HQC <- IC(sim, fit, "HQC")
BICadj <- IC(sim, fit, "BICadj")
```

---

localdecodeHMM   *Local Decoding for Hidden Markov Models*

Description

Computes the most likely hidden state sequence for a univariate time series using a fitted Hidden
Markov Model (HMM) object via the forward-backward algorithm.

Usage

```
localdecodeHMM(x, HMM, obsdist)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	List. A fitted HMM object, typically returned by [findmleHMM](#). Must contain estimated parameters delta and Pi, as well as state-dependent observation parameters.
obsdist	Character string. The observation distribution used in the fitted HMM. Supported distributions: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".

Details

This function uses the forward-backward algorithm to compute local state probabilities from a fitted Hidden Markov Model (HMM). The most probable state at each observation is returned. Initial state probabilities are derived from the stationary distribution of the transition matrix `Pi` if not explicitly provided.

Supported observation distributions include normal, Poisson, Weibull, zero-inflated Poisson (ZIP), negative binomial, zero-inflated negative binomial (ZINB), exponential, gamma, log-normal, generalized extreme value (GEV), zero-inflated normal, and zero-inflated gamma.

Value

Numeric vector of length `length(x)`, containing the most likely hidden state at each observation, computed via local decoding using the forward-backward algorithm.

Author(s)

Aimee Cody

See Also

`findmleHMM`, for estimating HMM parameters from data. `generateHMM`, for simulating HMM data. `localdecodeHSMM`, for local decoding of Hidden Semi-Markov Models. `globaldecodeHMM`, for global decoding in HMMs.

Examples

```
# Example with 3 states, normal observations
J <- 3

# HMM transition matrix
Pi <- matrix(c(0.8, 0.15, 0.05,
               0.1, 0.7, 0.2,
               0.2, 0.3, 0.5), nrow = 3, byrow = TRUE)

# Observation parameters (normal distribution)
obspar <- list(
  mean = c(-2, 0, 3),
  sd = c(0.5, 1, 1.5)
)

# Simulate HMM data
sim_data <- generateHMM(n = 200, J = J, obsdist = "norm", obspar = obspar, Pi = Pi)

# Fit HMM to simulated data
HMM_fit <- findmleHMM(J = J, x = sim_data$x, obsdist = "norm",
                      obspar = obspar, Pi = Pi, EM = FALSE)

# Compute local decoding
decoded_states <- localdecodeHMM(x = sim_data$x, HMM = HMM_fit, obsdist = "norm")

decoded_states
```

localdecodeHSMM   *Local Decoding for Hidden Semi-Markov Models*

Description

Computes the most likely hidden state sequence for a univariate time series using a fitted Hidden Semi-Markov Model (HSMM).

Usage

```
localdecodeHSMM(x, M = NA, HSMM, obsdist, dwelldist)
```

Arguments

x	Numeric vector. The observed data sequence.
M	Integer. Maximum dwell time to consider for semi-Markov states. Defaults to min(length(x), 1000) if NA.
HSMM	A list containing estimated HSMM parameters, including observationparameters, dwellparameters, Pi, and delta. Typically returned by findmleHSMM.
obsdist	Character string. Observation distribution. Supported distributions: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. Dwell-time distribution. Supported distributions: "pois", "nbinom", "betabinom".

Details

This function uses the forward-backward algorithm for HSMMs to compute local state probabilities and returns the most probable state at each time point. It supports a wide range of observation distributions and dwell-time distributions. The user can supply a maximum dwell-time truncation parameter M and optionally use a shift parameter for dwell times.

Value

states	Numeric vector. The most likely hidden state at each observation, computed via local decoding.

Author(s)

Aimee Cody

See Also

findmleHSMM, for estimating HSMM parameters from data. generateHSMM, for simulating HSMM data. localdecodeHMM, for local decoding of a hidden Markov model. globaldecodeHSMM, for finding the most likely sequence of states using global decoding.

Examples

```
J <- 3
Pi <- matrix(c(0.0, 0.6, 0.4,
               0.5, 0.0, 0.5,
               0.3, 0.7, 0.0), nrow = J, byrow = TRUE)

obspar <- list(
  mean = c(-4, 0, 4),
  sd   = c(1, 0.5, 1.5)
)

dwellpar <- list(
  lambda = c(3, 5, 4)
)

# Simulate HSMM data
sim <- generateHSMM(n = 1000, J = J, obsdist = "norm",
                    dwelldist = "pois", obspar = obspar,
                    dwellpar = dwellpar, Pi = Pi)

# Fit HSMM using the true parameters
HSMM_true <- findmleHSMM(x = sim$x, J = J, M = 100,
                         obsdist = "norm", dwelldist = "pois",
                         obspar = obspar, dwellpar = dwellpar,
                         Pi = Pi)

# Decode states using localdecodeHSMM
decoded <- localdecodeHSMM(x = sim$x, HSMM = HSMM_true,
                           obsdist = "norm", dwelldist = "pois",
                           M = 100)
decoded
```

---

plotHMMParameters   *Plot Hidden Markov Model Parameters Over Time*

Description

Plots the estimated state-dependent parameters of a fitted Hidden Markov Model (HMM) over time. Confidence intervals can be included, along with optional overlay of observed data for visual comparison.

Usage

```
plotHMMParameters(x, HMM, obsdist, confint_result = NULL,
                  level = 0.95, B = 100,
                  time_structure = NULL,
                  plot_title = "HMM Parameters Over Time",
                  overlay_data = NULL, overlay_label = "Data",
```

```
                           colors = c("black", "red", "blue", "green"),
                           save_plot = FALSE, filename = NULL,
                           width = 12, height = 8, dpi = 300,
                           verbose = TRUE, seed=NULL)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	A fitted HMM object (typically the output from [findmleHMM](#)), containing estimated transition probabilities Pi, initial probabilities delta, and state-dependent observation parameters.
obsdist	Character string. Observation distribution. Supported distributions include: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
confint_result	Optional result from [confintHMM](#). If not provided, confidence intervals are computed internally.
level	Numeric between 0 and 1. Confidence level for parameter intervals. Default is 0.95.
B	Integer. Number of bootstrap replicates used when computing confidence intervals (if not supplied). Default is 100.
time_structure	Optional list specifying the time scale for plotting. May include unit, observations_per_unit, and optional start_point, end_point. Supports calendar units (e.g., "day", "year") or custom scaling.
plot_title	Character string. Title for the full parameter plot layout. Default is "HMM Parameters Over Time".
overlay_data	Optional numeric vector of length length(x). External series to overlay on parameter plots (e.g., covariates or raw data).
overlay_label	Character string. Label for overlayed data axis. Default is "Data".
colors	Character vector of plotting colors. Default is c("black", "red", "blue", "green").
save_plot	Logical. If TRUE, saves the plot to file. Default is FALSE.
filename	Character string or NULL. Filename for saved plot (if save_plot = TRUE). Must be specified when save_plot = TRUE. Default is NULL.
width	Numeric. Width of saved plot in inches. Default is 12.
height	Numeric. Height of saved plot in inches. Default is 8.
dpi	Integer. Resolution of saved plot in dots per inch. Default is 300.
verbose	Logical. If TRUE, progress messages are printed to the console. Default is TRUE.
seed	Integer or NULL. Random seed for reproducibility. Default is NULL.

Details

For each observation, the most likely hidden state is decoded using [globaldecodeHMM](#). The corresponding state-dependent parameters are extracted and plotted over time. If confint_result is not supplied, bootstrap confidence intervals are computed via [confintHMM](#). Custom time scaling

can be applied using the `time_structure` argument. Overlay data (such as covariates or raw series) can be plotted on a secondary axis for comparison. When `verbose = TRUE`, progress messages are displayed during confidence interval computation.

Value

Invisibly returns a list containing:

`param_series`	List of parameter time series for each parameter.
`ci_series`	List of confidence interval time series (lower and upper bounds) for each parameter.
`time_info`	List containing time axis information and labels.
`decoded_states`	Vector of decoded hidden states for each time point.

The function also produces parameter trajectory plots with confidence intervals and optional overlays.

Author(s)

Aimee Cody

See Also

[findmleHMM](), for fitting HMMs. [generateHMM](), for simulating HMM data. [globaldecodeHMM](), for decoding the most likely state sequence. [confintHMM](), for bootstrap confidence intervals of HMM parameters.

Examples

```
set.seed(123)
J <- 2
Pi <- matrix(c(0.9, 0.1,
               0.2, 0.8), nrow = 2, byrow = TRUE)
delta <- c(0.5, 0.5)
obspar <- list(lambda = c(2, 7))

sim_data <- generateHMM(n = 120, J = J, obsdist = "pois",
                        obspar = obspar, Pi = Pi, delta = delta)

HMM_fit <- findmleHMM(x = sim_data$x, J = J, obsdist = "pois",
                      obspar = obspar, Pi = Pi, delta = delta)

overlay_series <- sim_data$x

time_struct <- list(unit = "day", observations_per_unit = 12, start_point = 1)


result <- plotHMMParameters(x = sim_data$x, HMM = HMM_fit, obsdist = "pois",
                            overlay_data = overlay_series,
                            overlay_label = "Observed counts",
                            time_structure = time_struct,
```

```
                              plot_title = "Poisson HMM Parameters with Overlay")

result <- plotHMMParameters(x = sim_data$x, HMM = HMM_fit, obsdist = "pois",
                            overlay_data = overlay_series,
                            overlay_label = "Observed counts",
                            time_structure = time_struct,
                            save_plot = TRUE,
                            filename = tempfile(fileext = ".png"),
                            verbose = FALSE)
```

---

plotHSMMParameters   *Plot Hidden Semi-Markov Model Parameters Over Time*

Description

Plots state-dependent observation parameters and dwell-time parameters of a fitted Hidden Semi-Markov Model (HSMM) over time. Confidence intervals can be included, with optional overlay of observed data for comparison.

Usage

```
plotHSMMParameters(x, HSMM, obsdist, dwelldist, confint_result = NULL,
                   level = 0.95, B = 100, M = NA, include_dwell = FALSE,
                   shift = FALSE, time_structure = NULL,
                   plot_title = "HSMM Parameters Over Time",
                   overlay_data = NULL, overlay_label = "Data",
                   colors = c("black", "red", "blue", "darkgreen"),
                   save_plot = FALSE, filename = NULL,
                   width = 12, height = 8, dpi = 300,
                   verbose = TRUE, seed = NULL)
```

Arguments

x	Numeric vector. The observed data sequence.
HSMM	A fitted HSMM object (typically the output from [findmleHSMM](#)), containing estimated transition probabilities Pi, initial probabilities delta, state-dependent observation parameters observationparameters, and dwell-time parameters dwellparameters.
obsdist	Character string. Observation distribution. Supported: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. Dwell-time distribution. Supported: "pois", "nbinom", "geom", "betabinom".
confint_result	Optional result from [confintHSMM](#). If not provided, confidence intervals are computed internally.
level	Numeric between 0 and 1. Confidence level for parameter intervals. Default is 0.95.

B	Integer. Number of bootstrap replicates for confidence intervals if not supplied. Default is 100.
M	Integer. Maximum dwell time used in decoding. Defaults to `min(1000, length(x))` if `NA`.
include_dwell	Logical. If TRUE, the dwell parameters will also be plotted through time. Default is FALSE.
shift	Logical. If TRUE, fits shifted dwell-time distributions. Default is FALSE.
time_structure	Optional list specifying the time scale. Must include `unit`, `observations_per_unit`, and optionally `start_point`, `end_point`. Supports calendar units ("day", "year", "hour") or custom scaling.
plot_title	Character string. Title for the full plot layout. Default is "HSMM Parameters Over Time".
overlay_data	Optional numeric vector of length `length(x)`. External data to overlay on parameter plots.
overlay_label	Character string. Label for overlay axis. Default is "Data".
colors	Character vector of plot colors. Default is `c("black", "red", "blue", "darkgreen")`.
save_plot	Logical. If TRUE, the plot is saved to a file. Default is FALSE.
filename	Character string. File path for saving the plot. Required if `save_plot = TRUE`.
width	Numeric. Width of saved plot in inches. Default is 12.
height	Numeric. Height of saved plot in inches. Default is 8.
dpi	Numeric. Resolution (dots per inch) for saved plot. Default is 300.
verbose	Logical. If TRUE, progress messages are printed to the console. Default is TRUE.
seed	Integer or `NULL`. Random seed for reproducibility. Default is `NULL`.

Details

The most likely hidden state sequence is decoded using [globaldecodeHSMM](#). For each decoded state, observation and dwell-time parameters are extracted and plotted over time. If confint_result is not provided, bootstrap intervals are computed via [confintHSMM](#). Plots include observation parameter trajectories, dwell-time parameter trajectories (if include_dwell = TRUE), and optional overlay data on secondary axes. Time scaling is configurable using time_structure. When verbose = TRUE, progress messages are displayed during confidence interval computation. If save_plot = TRUE, the plot is saved as a PNG file to the specified filename with the given dimensions and resolution.

Value

Invisibly returns a list containing:

obs_param_series	
	List of observation parameter time series for each parameter.
obs_ci_series	List of confidence interval time series (lower and upper bounds) for each observation parameter.

dwell_param_series

> List of dwell parameter time series for each parameter (if include_dwell = TRUE).

dwell_ci_series

> List of confidence interval time series for each dwell parameter (if include_dwell = TRUE).

time_info   List containing time axis information and labels.

decoded_states   Vector of decoded hidden states for each time point.

The function also generates multi-panel plots of parameter trajectories with confidence intervals and optional overlays.

Author(s)

Aimee Cody

See Also

[findmleHSMM](#), for fitting HSMMs. [generateHSMM](#), for simulating HSMM data. [globaldecodeHSMM](#), for decoding the most likely state sequence. [confintHSMM](#), for bootstrap confidence intervals of HSMM parameters.

Examples

```
set.seed(42)
J <- 2
Pi <- matrix(c(0.0, 1.0,
               1.0, 0.0), nrow = J, byrow = TRUE)

obspar <- list(mean = c(0, 3), sd = c(1, 1.2))
dwellpar <- list(lambda = c(5, 8))

sim <- generateHSMM(n = 200, J = J,
                    obsdist = "norm", dwelldist = "pois",
                    obspar = obspar, dwellpar = dwellpar,
                    Pi = Pi)

HSMM_fit <- findmleHSMM(x = sim$x, J = J, M = 100,
                        obsdist = "norm", dwelldist = "pois",
                        obspar = obspar, dwellpar = dwellpar,
                        Pi = Pi)

overlay_series <- sim$x
time_struct <- list(unit = "day", observations_per_unit = 10, start_point = 1)


result <- plotHSMMParameters(x = sim$x, HSMM = HSMM_fit,
                             obsdist = "norm", dwelldist = "pois",
                             overlay_data = overlay_series,
                             overlay_label = "Observed values",
                             time_structure = time_struct,
```

```
                          plot_title = "HSMM Parameters with Overlay")

result_silent <- plotHSMMParameters(x = sim$x, HSMM = HSMM_fit,
                               obsdist = "norm", dwelldist = "pois",
                               include_dwell = TRUE,
                               overlay_data = overlay_series,
                               time_structure = time_struct,
                               verbose = FALSE)
```

---

residualsHMM   *Ordinary Residuals for Hidden Markov Models*

Description

Computes ordinary (probability integral transform) residuals for a fitted Hidden Markov Model (HMM). Generates quantile-quantile and autocorrelation plots for assessing model fit.

Usage

```
residualsHMM(x, HMM, obsdist, lag.max = 50, verbose = TRUE)
```

Arguments

x	Numeric vector. The observed data sequence.
HMM	A fitted HMM object (typically the output from [findmleHMM](#)), containing estimated transition probabilities Pi, initial probabilities delta, and state-dependent observation parameters.
obsdist	Character string. Observation distribution. Supported distributions include: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
lag.max	Integer. Maximum lag for autocorrelation function (ACF) computation. Default is 50.
verbose	Logical. If TRUE, prints proportion of residuals outside 95% interval. Default is TRUE.

Details

The function computes probability integral transform (PIT) residuals for HMM observations by combining forward and backward probabilities with the cumulative distribution functions of the state-dependent observations. The residuals are assessed via:

- QQ plots comparing observed residuals to the expected standard normal distribution.
- Autocorrelation function plots to detect remaining temporal dependence.

This provides a visual check of model adequacy, where ideally residuals should behave as independent standard normal variables. When verbose = TRUE, the proportion of residuals outside the 95% interval is printed.

Value

Invisibly returns a list containing:

residuals   Vector of ordinary residuals.

probabilities   Vector of probability integral transform values.

qq_bands   List containing median, lower, and upper quantiles for QQ plot confidence bands.

acf_values   ACF object containing autocorrelation values.

acf_bands   List containing lower and upper confidence bands for ACF plot.

proportion_outside
                   Numeric value indicating the proportion of residuals outside the 95% interval.

The function also produces diagnostic plots:

- Generates a QQ plot of ordinary residuals against expected standard normal quantiles, with 95% confidence intervals.
- Generates an ACF plot of the ordinary residuals with approximate 95% confidence bands.

Author(s)

Aimee Cody

See Also

[findmleHMM](), for fitting HMMs. [generateHMM](), for simulating HMM data. [residualsHSMM](), for residual analysis of hidden semi-Markov models.

Examples

```
J <- 2
Pi <- matrix(c(0.9, 0.1,
               0.2, 0.8), nrow = 2, byrow = TRUE)
delta <- c(0.5, 0.5)
obspar <- list(lambda = c(2, 7))

sim_data <- generateHMM(n = 200, J = J, obsdist = "pois", obspar = obspar, Pi = Pi, delta = delta)

HMM_fit <- findmleHMM(x = sim_data$x, J = J, obsdist = "pois",
                      obspar = obspar, Pi = Pi, delta = delta)

result <- residualsHMM(x = sim_data$x, HMM = HMM_fit, obsdist = "pois", lag.max = 30)

result_silent <- residualsHMM(x = sim_data$x, HMM = HMM_fit, obsdist = "pois",
                              lag.max = 30, verbose = FALSE)
```

---

residualsHSMM   *Ordinary Residuals for Hidden Semi-Markov Models*

Description

Computes ordinary (probability integral transform) residuals for a fitted Hidden Semi-Markov Model (HSMM). Generates quantile-quantile and autocorrelation plots for assessing model fit via simulation-based envelopes.

Usage

```
residualsHSMM(x, HSMM, obsdist, dwelldist, M = NA, lag.max = 50, nsim = 100,
              use.theoretical.acf = FALSE, verbose = TRUE, seed = NULL)
```

Arguments

x	Numeric vector. The observed data sequence.
HSMM	A fitted HSMM object (typically the output from findmleHSMM), containing estimated transition probabilities Pi, initial probabilities delta, state-dependent observation parameters observationparameters, and dwell-time parameters dwellparameters.
obsdist	Character string. Observation distribution. Supported distributions include: "norm", "pois", "weibull", "zip", "nbinom", "zinb", "exp", "gamma", "lnorm", "gev", "ZInormal", "ZIgamma".
dwelldist	Character string. Dwell-time distribution. Supported distributions: "pois", "nbinom", "geom", "betabinom".
M	Integer. Maximum dwell time to consider for semi-Markov states. Defaults to min(1000, length(x)) if NA.
lag.max	Integer. Maximum lag for autocorrelation function (ACF) computation. Default is 50.
nsim	Integer. Number of simulated datasets to generate for constructing the residual envelope. Default is 100.
use.theoretical.acf	
	Logical. If TRUE, uses theoretical 95% ACF bands for white noise instead of simulation-based envelope. Default is FALSE.
verbose	Logical. If TRUE, prints progress messages during simulation and diagnostic summaries. Default is TRUE.
seed	Integer or NULL. Random seed for reproducibility. Default is NULL.

Details

The function computes probability integral transform (PIT) residuals for HSMM observations by combining forward-backward probabilities with cumulative distribution functions of the state-dependent observations and dwell-time distributions. Residuals are assessed via:

- QQ plots comparing observed residuals to the expected standard normal distribution with a 95% simulation-based envelope.

- Autocorrelation function plots to detect remaining temporal dependence with simulation-based or theoretical 95% bands.

This provides a visual check of model adequacy, where ideally residuals behave as independent standard normal variables.

Value

The function generates QQ and ACF plots and prints messages about residuals outside the 95% envelope. An invisible list is also returned containing diagnostic information.

Author(s)

Aimee Cody

See Also

findmleHSMM, for estimating HSMM parameters from data. generateHSMM, for simulating HSMM data. residualsHMM, for residual analysis of hidden Markov models.

Examples

```
J <- 3
Pi <- matrix(c(0.0, 0.6, 0.4,
               0.5, 0.0, 0.5,
               0.3, 0.7, 0.0), nrow = J, byrow = TRUE)
obspar <- list(
  mean = c(-2, 0, 3),
  sd   = c(1, 1.5, 2)
)
dwellpar <- list(
  lambda = c(3, 5, 4)
)
# Simulate HSMM data
sim <- generateHSMM(n = 200, J = J, obsdist = "norm",
                    dwelldist = "pois", obspar = obspar,
                    dwellpar = dwellpar, Pi = Pi)
# Fit HSMM using the true parameters
HSMM_true <- findmleHSMM(x = sim$x, J = J, M = 100,
                         obsdist = "norm", dwelldist = "pois",
                         obspar = obspar, dwellpar = dwellpar,
                         Pi = Pi)
# Compute residuals and diagnostic plots
residualsHSMM(x = sim$x, HSMM = HSMM_true,
              obsdist = "norm", dwelldist = "pois",
              M = 100, nsim = 50, lag.max = 30)
```

weekly_maxima	*Weekly maxima of geomagnetic rate-of-change data from Eskdalemuir Magnetic Observatory*

Description

A dataset containing weekly block maxima of the horizontal geomagnetic field rate-of-change at the Eskdalemuir Magnetic Observatory, Scotland, United Kingdom. These data quantify the variability of the Earth's magnetic field and capture periods of extreme geomagnetic disturbance from 1999 to 2022.

Usage

```
data(weekly_maxima)
```

Format

A numeric vector of weekly maxima of

$$R_1(t) = \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2},$$

where X_t and Y_t are the horizontal north and east components of the Earth's magnetic field, respectively, measured at one-minute intervals.

Details

Aggregating to a weekly scale smooths out short-lived fluctuations while retaining the most significant geomagnetic disturbances, as well as reducing short-term serial dependencies. The extreme nature of the data motivates modelling via the Generalised Extreme Value (GEV) distribution, with density

$$f(x) = \frac{1}{\sigma} \left[1 + \xi \left(\frac{x - \mu}{\sigma} \right) \right]^{-\frac{1}{\xi} - 1} \exp \left\{ - \left[1 + \xi \left(\frac{x - \mu}{\sigma} \right) \right]^{-\frac{1}{\xi}} \right\},$$

where μ, σ, and ξ are the location, scale, and shape parameters, respectively.

Source

Eskdalemuir Magnetic Observatory, British Geological Survey, United Kingdom.

Derived and aggregated for use in **HMMHSMM**.

Examples

```
data(weekly_maxima)
hist(weekly_maxima, breaks = 30, main = "Weekly maxima of geomagnetic variability",
     xlab = "Weekly maximum rate of change (nT/min)")
```

Index