Package 'dyadicMarkov'

June 21, 2026

Type Package

Title Pattern Estimation and Identification for Dyadic Sequences Using Transition Matrices in R

Version 0.1.1

Description Provides methods for analyzing categorical dyadic sequences using transition matrices within the Longitudinal Actor-Partner Interdependence Model and Markov-chain framework. The package supports empirical transition counts, maximum likelihood estimation of transition probabilities, and identification of univariate and bivariate patterns of interaction in dyadic sequences.

License MIT + file LICENSE

URL https://github.com/BoellenruecherM/dyadicMarkov-public, https://boellenruecherm.github.io/dyadicMarkov-public/

BugReports https://github.com/BoellenruecherM/dyadicMarkov-public/issues

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.3

Config/testthat/edition 3

LazyData true

NeedsCompilation no

Author Mattia Böllenrücher [aut, cre, cph] (ORCID: <https://orcid.org/0009-0004-4149-4745>), Mégane Bollenrücher [aut] (ORCID: <https://orcid.org/0000-0002-9035-8799>), Jean-Philippe Antonietti [aut] (ORCID: <https://orcid.org/0000-0003-0117-4769>)

Maintainer Mattia Böllenrücher <mattia.boellenruecher@student.unisg.ch>

Repository CRAN

Date/Publication 2026-06-21 13:40:02 UTC

Contents

`bivariateCase` *Bivariate case identification for dyadic Markov chains*

Description

Identifies the bivariate case as `"trivial"`, `"univariate"`, `"partial"`, or `"complete"` using two likelihood-ratio tests against constrained bivariate structures.

Usage

```
bivariateCase(empirical, alpha = 0.05)
```

Arguments

`empirical`	An empirical bivariate count matrix with 16 rows and 2 columns, as returned by `countEmpBivariate`.
`alpha`	A single number in (0, 1) giving the significance level. Default is 0.05.

Details

The returned case corresponds to the global approach of the bivariate method. It determines whether the sequence analyzed is treated as trivial, univariate, partial bivariate, or complete bivariate before the local identification of the pattern of interaction.

Value

A list with class `c("dyadic_case", "list")` containing components `testUnivariate`, `testPartial`, `case`, and metadata fields `alpha` and `call`. It remains usable as an ordinary list.

Examples

```
chainFM_V1 <- c(1L, 2L, 1L, 2L, 2L, 1L)
chainSM_V1 <- c(2L, 1L, 2L, 1L, 1L, 2L)
chainFM_V2 <- c(1L, 1L, 2L, 2L, 1L, 2L)
chainSM_V2 <- c(2L, 2L, 1L, 1L, 2L, 1L)
emp <- countEmpBivariate(
  chainFM_V1, chainSM_V1, chainFM_V2, chainSM_V2,
  states = 2L
)
bivariateCase(emp, alpha = 0.05)
```

---

completePattern *Complete bivariate pattern identification by AIC*

Description

Compares the complete bivariate patterns C, D1–D4, and E1–E4 using AIC and returns the selected pattern.

Usage

```
completePattern(empirical)
```

Arguments

`empirical`	An empirical bivariate count matrix with 16 rows and 2 columns, as returned by `countEmpBivariate`.

Details

Conditional on the complete bivariate case, AIC is used to select among the C, D1–D4, and E1–E4 structures.

Value

A list with class `c("dyadic_pattern", "list")` containing components `aic` (a data frame with columns `pattern`, `matrix`, and `aic`), `pattern` (the selected pattern label), and `call`. It remains usable as an ordinary list.

Examples

```
chainFM_V1 <- c(1L, 2L, 1L, 2L, 2L, 1L)
chainSM_V1 <- c(2L, 1L, 2L, 1L, 1L, 2L)
chainFM_V2 <- c(1L, 1L, 2L, 2L, 1L, 2L)
chainSM_V2 <- c(2L, 2L, 1L, 1L, 2L, 1L)
emp <- countEmpBivariate(
  chainFM_V1, chainSM_V1, chainFM_V2, chainSM_V2,
  states = 2L
)
completePattern(emp)
```

countEmp *Empirical transition counts for univariate dyadic sequences*

Description

Computes empirical transition counts for the sequences of the first and second member. Rows correspond to dyadic states of the two members, and columns correspond to the state of the first member at the next time point.

Usage

```
countEmp(chainFM, chainSM, states)
```

Arguments

chainFM	Vector of observed states for the first member (FM).
chainSM	Vector of observed states for the second member (SM).
states	A single integer >= 2 giving the number of states.

Details

Rows correspond to current dyadic states (FM_t, SM_t). For general `states`, the row index is computed as `1 + states * (FM_t - 1) + (SM_t - 1)`. Columns correspond to the state of the first member at the next time point, FM_{t+1}.

Value

An integer matrix with class `c("dyadic_counts", "matrix", "array")`, with $states^2$ rows and `states` columns. It remains usable as an ordinary matrix.

Examples

```
chainFM <- c(1L, 2L, 1L, 2L, 2L, 1L)
chainSM <- c(2L, 1L, 2L, 1L, 1L, 2L)
countEmp(chainFM, chainSM, states = 2L)
```

countEmpBivariate *Empirical transition counts for bivariate dyadic sequences*

Description

Computes empirical transition counts for bivariate categorical dyadic sequences with two variables. This function currently supports `states = 2` only.

Usage

```
countEmpBivariate(chainFM_V1, chainSM_V1, chainFM_V2, chainSM_V2, states = 2L)
```

Arguments

`chainFM_V1`, `chainSM_V1`	Vectors of observed states for variable 1 for the first and second member.
`chainFM_V2`, `chainSM_V2`	Vectors of observed states for variable 2 for the first and second member.
`states`	A single integer. Currently only 2 is supported. Default is 2.

Details

The bivariate counter currently supports `states = 2` only. Rows represent the previous dyadic states of variable 1 and variable 2. The implementation uses the row mapping `states^2 * (states * (FM_V1,t - 1) + (SM_V1,t - 1)) + states * (FM_V2,t - 1) + (SM_V2,t - 1) + 1`. Columns correspond to the state of the first member on variable 1 at the next time point, $FM_{V1,t+1}$.

Value

An integer matrix with class `c("dyadic_counts", "matrix", "array")` with 16 rows and 2 columns when `states = 2`. It remains usable as an ordinary matrix.

Examples

```
chainFM_V1 <- c(1L, 2L, 1L, 2L, 2L, 1L)
chainSM_V1 <- c(2L, 1L, 2L, 1L, 1L, 2L)
chainFM_V2 <- c(1L, 1L, 2L, 2L, 1L, 2L)
chainSM_V2 <- c(2L, 2L, 1L, 1L, 2L, 1L)
emp <- countEmpBivariate(
  chainFM_V1, chainSM_V1, chainFM_V2, chainSM_V2,
  states = 2L
)
dim(emp)
```

`dyadic_bivariate_example` *Synthetic bivariate dyadic sequence example*

Description

A synthetic bivariate dyadic sequence with 90 observations, designed for package workflow examples.

Usage

```
dyadic_bivariate_example
```

Format

A data frame with 90 rows and 5 columns:

time Index of the measurement occasion.

FM_V1 Integer variable 1 state for the first member, taking values 1 or 2.

SM_V1 Integer variable 1 state for the second member, taking values 1 or 2.

FM_V2 Integer variable 2 state for the first member, taking values 1 or 2.

SM_V2 Integer variable 2 state for the second member, taking values 1 or 2.

Details

The bivariate workflow classifies this example as `complete` using `bivariateCase` with `alpha = 0.05`. The complete bivariate pattern selected by `completePattern` is `D2`.

`dyadic_univariate_example`

Synthetic univariate dyadic sequence example

Description

A synthetic dyadic sequence with 90 observations, designed for package workflow examples.

Usage

```
dyadic_univariate_example
```

Format

A data frame with 90 rows and 3 columns:

time Index of the measurement occasion.

FM Integer state for the first member, taking values 1 or 2.

SM Integer state for the second member, taking values 1 or 2.

Details

The package workflow classifies this example as `PM (A3)` using `univariatePattern` with `states = 2`.

mleEstimation *Maximum likelihood estimation of transition probabilities*

Description

Estimates transition probabilities by maximum likelihood from an empirical count matrix returned by `countEmp` or `countEmpBivariate`.

Usage

```
mleEstimation(empirical)
```

Arguments

`empirical`	An empirical count matrix.

Details

Each row of `empirical` is normalized independently. Rows with zero total count are assigned a uniform probability vector, so each row of the returned matrix sums to one.

Value

A numeric matrix with class `c("dyadic_mle", "matrix", "array")` containing estimated transition probabilities with the same dimensions as `empirical`. It remains usable as an ordinary matrix.

Examples

```
chainFM <- c(1L, 2L, 1L, 2L, 2L, 1L)
chainSM <- c(2L, 1L, 2L, 1L, 1L, 2L)
emp <- countEmp(chainFM, chainSM, states = 2L)
mleEstimation(emp)
```

partialPattern *Partial bivariate pattern identification by AIC*

Description

Compares the partial bivariate patterns B1, B2, and B3 using AIC and returns the selected pattern.

Usage

```
partialPattern(empirical)
```

Arguments

`empirical`	An empirical bivariate count matrix with 16 rows and 2 columns, as returned by `countEmpBivariate`.

Details

Conditional on the partial bivariate case, AIC is used to select among the B1, B2, and B3 structures.

Value

A list with class `c("dyadic_pattern", "list")` containing components `aic` (a data frame with candidate patterns and AIC values), `pattern` (the selected pattern label), and `call`. It remains usable as an ordinary list.

Examples

```
chainFM_V1 <- c(1L, 2L, 1L, 2L, 2L, 1L)
chainSM_V1 <- c(2L, 1L, 2L, 1L, 1L, 2L)
chainFM_V2 <- c(1L, 1L, 2L, 2L, 1L, 2L)
chainSM_V2 <- c(2L, 2L, 1L, 1L, 2L, 1L)
emp <- countEmpBivariate(
  chainFM_V1, chainSM_V1, chainFM_V2, chainSM_V2,
  states = 2L
)
partialPattern(emp)
```

`univariatePattern` *Univariate pattern identification for dyadic Markov chains*

Description

Computes empirical transition counts, estimates transition probabilities by maximum likelihood, and performs likelihood-ratio tests against the actor-only and partner-only constrained models to identify the univariate pattern of interaction.

Usage

```
univariatePattern(chainFM, chainSM, states, alpha = 0.05)
```

Arguments

`chainFM`	Vector of observed states for the first member (FM).
`chainSM`	Vector of observed states for the second member (SM).
`states`	A single integer >= 2 giving the number of states.
`alpha`	A single number in (0, 1) giving the significance level. Default is 0.05.

Details

Pattern labels summarize which structure is retained by the tests: `IM` (`A0`) denotes an independence pattern, `APM` (`A1`) an actor-partner pattern, `AM` (`A2`) an actor-only pattern, and `PM` (`A3`) a partner-only pattern.

Value

A list with class `c("dyadic_pattern", "list")` containing two `htest` objects (`TEST.AM`, `TEST.PM`), a string `pattern`, and metadata fields `alpha`, `states`, and `call`. It remains usable as an ordinary list.

Examples

```
chainFM <- c(1L, 2L, 1L, 2L, 2L, 1L)
chainSM <- c(2L, 1L, 2L, 1L, 1L, 2L)
univariatePattern(chainFM, chainSM, states = 2L, alpha = 0.05)
```

Index