Package: EvolutionaryGames (via r-universe)

August 24, 2024

Type Package

Title Important Concepts of Evolutionary Game Theory

Version 0.1.2

Maintainer Jochen Staudacher <jochen.staudacher@hs-kempten.de>

Description Evolutionary game theory applies game theory to evolving populations in biology, see e.g. one of the books by Weibull (1994, ISBN:978-0262731218) or by Sandholm (2010, ISBN:978-0262195874) for more details. A comprehensive set of tools to illustrate the core concepts of evolutionary game theory, such as evolutionary stability or various evolutionary dynamics, for teaching and academic research is provided.

License GPL-2

Encoding UTF-8

Imports deSolve (>= 1.14), geometry (>= 0.3-6), ggplot2 (>= 2.2.1), grDevices (>= 3.2.2), interp (>= 1.0-29), MASS (>= 7.3-43), reshape2 (>= 1.4.2)

Suggests knitr, rmarkdown, rgl

RoxygenNote 7.1.2

VignetteBuilder knitr

NeedsCompilation no

Author Daniel Gebele [aut, cph], Jochen Staudacher [aut, cre, cph]

Date/Publication 2022-08-29 00:10:02 UTC

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

RemoteUrl https://github.com/cran/EvolutionaryGames

RemoteRef HEAD

RemoteSha 088361b2739f7e6bb781f3d5cedf83cea08ac563

Contents

BNN *Brown-von Neumann-Nash dynamic*

Description

Brown-von Neumann-Nash replicator dynamic as a type of evolutionary dynamics.

Usage

```
BNN(time, state, parameters)
```

Arguments

`time`	Regular sequence that represents the time sequence under which simulation takes place.
`state`	Numeric vector that represents the initial state.
`parameters`	Numeric vector that represents parameters needed by the dynamic.

Value

Numeric list. Each component represents the rate of change depending on the dynamic.

Author(s)

Daniel Gebele `<dngebele@gmail.com>`

References

Brown, G. W. and von Neumann, J. (1950) "Solutions of games by differential equations", In: Kuhn, Harold William and Tucker, Albert William (Eds.) "Contributions to the Theory of Games I", Princeton University Press, pp. 73–79.

Examples

```
dynamic <- BNN
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3), 1, 3, byrow=TRUE)
phaseDiagram3S(A, dynamic, NULL, state, FALSE, FALSE)
```

BR *BR dynamic*

Description

Best response dynamic as a type of evolutionary dynamics.

Usage

```
BR(time, state, parameters)
```

Arguments

`time`	Regular sequence that represents the time sequence under which simulation takes place.
`state`	Numeric vector that represents the initial state.
`parameters`	Numeric vector that represents parameters needed by the dynamic.

Value

Numeric list. Each component represents the rate of change depending on the dynamic.

Author(s)

Daniel Gebele <dngebele@gmail.com>

References

Gilboa, I. and Matsui, A. (1991) "Social Stability and Equilibrium", Econometrica 59, pp. 859–867.

Examples

```
dynamic <- BR
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3), 1, 3, byrow=TRUE)
phaseDiagram3S(A, dynamic, NULL, state, FALSE, FALSE)
```

ESS — *ESS for two-player games with a maximum of three strategies*

Description

Computes Evolutionary Stable Strategies of a game with two players and a maximum of three strategies.

Usage

```
ESS(A, strategies = c(), floats = TRUE)
```

Arguments

`A`	Numeric matrix of size 2x2 or 3x3 representing the number of strategies of a symmetric matrix game.
`strategies`	String vector of length n that names all strategies whereas n represents the number of strategies.
`floats`	Logical value that handles number representation. If set to `TRUE`, floating-point arithmetic will be used, otherwise fractions. Default is `TRUE`.

Value

Numeric matrix. Each row represents an ESS.

Author(s)

Daniel Gebele <dngebele@gmail.com>

References

Smith, J. M. and Price, G. R. (1973) "The logic of animal conflict", Nature 246, pp. 15–18.

Examples

```
ESS(matrix(c(-1, 4, 0, 2), 2, byrow=TRUE), c("Hawk", "Dove"), FALSE)
ESS(matrix(c(1, 2, 0, 0, 1, 2, 2, 0, 1), 3, byrow=TRUE))
```

ESset *Evolutionarily stable set for two-player games with three strategies*

Description

Computes evolutionarily stable sets of a game with two players and three strategies.

Usage

```
ESset(A, strategies = c("1", "2", "3"), floats = TRUE)
```

Arguments

`A`	Numeric matrix of size 3x3 representing the number of strategies of a symmetric matrix game.
`strategies`	String vector of length 3 that names all strategies.
`floats`	Logical value that handles number representation. If set to `TRUE`, floating-point arithmetic will be used, otherwise fractions. Default is `TRUE`.

Value

Numeric matrix. Each row represents the start and end point of a line (ESset). In addition, a plot of the ESset in the game will be created.

Author(s)

Daniel Gebele <dngebele@gmail.com>

References

Thomas, B. (1985) "On evolutionarily stable sets", Journal of Mathematical Biology 22, pp. 105–115.

Examples

```
# Please note that the computation of evolutionarily stable sets
# is rather time-consuming.
# Depending on your machine you might need to wait more
# than 10 seconds in order to run the following example.
## Not run:
A <- matrix(c(-2, 5, 10/9, 0, 5/2, 10/9, -10/9, 35/9, 10/9), 3, byrow=TRUE)
strategies <- c("Hawk", "Dove", "Mixed ESS")
ESset(A, strategies)

## End(Not run)
```

ILogit *ILogit dynamic*

Description

Imitative Logit dynamic as a type of evolutionary dynamics.

Usage

```
ILogit(time, state, parameters)
```

Arguments

time	Regular sequence that represents the time sequence under which simulation takes place.
state	Numeric vector that represents the initial state.
parameters	Numeric vector that represents parameters needed by the dynamic.

Value

Numeric list. Each component represents the rate of change depending on the dynamic.

Author(s)

Jochen Staudacher <jochen.staudacher@hs-kempten.de>

References

Weibull, J. W. (1997) "Evolutionary Game Theory", MIT Press.

Examples

```
dynamic <- ILogit
A <- matrix(c(-1, 0, 0, 0, -1, 0, 0, 0, -1), 3, byrow=TRUE)
state <- matrix(c(0.1, 0.2, 0.7, 0.2, 0.7, 0.1, 0.9, 0.05, 0.05), 3, 3, byrow=TRUE)
eta <- 0.7
phaseDiagram3S(A, dynamic, eta, state, TRUE, FALSE)
```

`Logit` *Logit dynamic*

Description

Logit dynamic as a type of evolutionary dynamics.

Usage

```
Logit(time, state, parameters)
```

Arguments

`time`	Regular sequence that represents the time sequence under which simulation takes place.
`state`	Numeric vector that represents the initial state.
`parameters`	Numeric vector that represents parameters needed by the dynamic.

Value

Numeric list. Each component represents the rate of change depending on the dynamic.

Author(s)

Daniel Gebele <dngebele@gmail.com>

References

Fudenberg, D. and Levine, D. K. (1998) "The Theory of Learning in Games", MIT Press.

Examples

```
dynamic <- Logit
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3), 1, 3, byrow=TRUE)
eta <- 0.1
phaseDiagram3S(A, dynamic, eta, state, FALSE, FALSE)
```

`MSReplicator` *Maynard Smith replicator dynamic*

Description

Maynard Smith replicator dynamic as a type of evolutionary dynamics.

Usage

```
MSReplicator(time, state, parameters)
```

Arguments

`time`	Regular sequence that represents the time sequence under which simulation takes place.
`state`	Numeric vector that represents the initial state.
`parameters`	Numeric vector that represents parameters needed by the dynamic.

Value

Numeric list. Each component represents the rate of change depending on the dynamic.

Author(s)

Daniel Gebele <dngebele@gmail.com>

References

Smith, J. M. (1982) "Evolution and the Theory of Games", Cambridge University Press.

Examples

```
dynamic <- MSReplicator
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3), 1, 3, byrow=TRUE)
phaseDiagram3S(A, dynamic, NULL, state, FALSE, FALSE)
```

`phaseDiagram2S` *Phase Diagram for two-player games with two strategies*

Description

Plots phase diagram of a game with two players and two strategies.

Usage

```
phaseDiagram2S(
  A,
  dynamic,
  params = NULL,
  vectorField = TRUE,
  strategies = c("1", "2")
)
```

Arguments

`A`	Numeric matrix of size 2x2 representing the number of strategies of a symmetric matrix game.
`dynamic`	Function representing an evolutionary dynamic.
`params`	Numeric vector representing additional parameters for the evolutionary dynamic.
`vectorField`	Logical value that handles vector field presentation. If set to `TRUE`, vector field will be shown, otherwise not. Default is `TRUE`.
`strategies`	String vector of length 2 that names all strategies.

Value

None.

Author(s)

Daniel Gebele <dngebele@gmail.com>

Examples

```
A <- matrix(c(-1, 4, 0, 2), 2, 2, byrow=TRUE)
phaseDiagram2S(A, Replicator, strategies = c("Hawk", "Dove"))
```

phaseDiagram3S — *Phase Diagram for two-player games with three strategies*

Description

Plots phase diagram of a game with two players and three strategies.

Usage

```
phaseDiagram3S(
  A,
  dynamic,
  params = NULL,
  trajectories = NULL,
  contour = FALSE,
  vectorField = FALSE,
  strategies = c("1", "2", "3")
)
```

Arguments

`A`	Numeric matrix of size 3x3 representing the number of strategies of a symmetric matrix game.
`dynamic`	Function representing an evolutionary dynamic.
`params`	Numeric vector with additional parameters for the evolutionary dynamic.
`trajectories`	Numeric matrix of size mx3. Each row represents the initial values for the trajectory to be examined.
`contour`	Logical value that handles contour diagram presentation. If set to `TRUE`, contour diagram will be shown, otherwise not. Default is `FALSE`.
`vectorField`	Logical value that handles vector field presentation. If set to `TRUE`, vector field will be shown, otherwise not. Default is `FALSE`.
`strategies`	String vector of length 3 that names all strategies.

Value

None.

Author(s)

Daniel Gebele <dngebele@gmail.com>

Examples

```
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3), 1, 3, byrow=TRUE)

phaseDiagram3S(A, Replicator, NULL, state, FALSE, FALSE)
phaseDiagram3S(A, Replicator, NULL, state, TRUE, TRUE)

# Plot two trajectories rather than only one:
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3, 0.6, 0.2, 0.2), 2, 3, byrow=TRUE)
phaseDiagram3S(A, Replicator, NULL, state, FALSE, FALSE)
```

---

phaseDiagram4S *Phase Diagram for two-player games with four strategies*

Description

Plots phase diagram of a game with two players and four strategies.

Usage

```
phaseDiagram4S(
  A,
  dynamic,
  params = NULL,
  trajectory = NULL,
  strategies = c("1", "2", "3", "4"),
  noRGL = TRUE
)
```

Arguments

`A`	Numeric matrix of size 4x4 representing the number of strategies of a symmetric matrix game.
`dynamic`	Function representing an evolutionary dynamic.
`params`	Numeric vector with additional parameters for the evolutionary dynamic.
`trajectory`	Numeric vector of size 4 representing the initial value for the trajectory to be examined.
`strategies`	String vector of length 4 that names all strategies.
`noRGL`	Logical value that handles diagram rotation. If set to `FALSE`, diagram will be rotatable, otherwise not. Default is `TRUE`.

Value

None.

Author(s)

Daniel Gebele <dngebele@gmail.com>

Examples

```
A <- matrix(c(5, -9, 6, 8, 20, 1, 2, -18, -14, 0, 2, 20, 13, 0, 4, -13),
 4, 4, byrow=TRUE)
state <- c(0.3, 0.2, 0.1, 0.4)
phaseDiagram4S(A, Replicator, NULL, state)
```

`Replicator` *Replicator dynamic*

Description

Replicator dynamic as a type of evolutionary dynamics.

Usage

```
Replicator(time, state, parameters)
```

Arguments

`time`	Regular sequence that represents the time sequence under which simulation takes place.
`state`	Numeric vector that represents the initial state.
`parameters`	Numeric vector that represents parameters needed by the dynamic.

Value

Numeric list. Each component represents the rate of change depending on the dynamic.

Author(s)

Daniel Gebele <dngebele@gmail.com>

References

Taylor, P. D. and Jonker, L. B. (1978) "Evolutionary stable strategies and game dynamics", Mathematical Biosciences 40 (1-2), pp. 145–156.

Examples

```
dynamic <- Replicator
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3), 1, 3, byrow=TRUE)
phaseDiagram3S(A, dynamic, NULL, state, FALSE, FALSE)
```

Smith *Smith dynamic*

Description

Smith dynamic as a type of evolutionary dynamics.

Usage

```
Smith(time, state, parameters)
```

Arguments

`time`	Regular sequence that represents the time sequence under which simulation takes place.
`state`	Numeric vector that represents the initial state.
`parameters`	Numeric vector that represents parameters needed by the dynamic.

Value

Numeric list. Each component represents the rate of change depending on the dynamic.

Author(s)

Daniel Gebele <dngebele@gmail.com>

References

Smith, M. J. (1984) "The Stability of a Dynamic Model of Traffic Assignment – An Application of a Method of Lyapunov", Transportation Science 18, pp. 245–252.

Examples

```
dynamic <- Smith
A <- matrix(c(0, -2, 1, 1, 0, -2, -2, 1, 0), 3, byrow=TRUE)
state <- matrix(c(0.4, 0.3, 0.3), 1, 3, byrow=TRUE)
phaseDiagram3S(A, dynamic, NULL, state, FALSE, FALSE)
```

`triangle` *Triangle for 2-simplex operations*

Description

Generates a triangle representing the 2-simplex.

Usage

```
triangle(labels = c("1", "2", "3"))
```

Arguments

`labels`	String vector of length 3 that names the edges of the triangle.

Value

List of size 2 with members `coords` and `canvas`. `coords` holds edge coordinates of the 2-simplex, `canvas` a ggplot2 plot object of the 2-simplex.

Author(s)

Daniel Gebele <`dngebele@gmail.com`>

Examples

```
triangle()
```

Index