

Package: rgeomstats (via r-universe)

October 26, 2024

Title Interface to 'Geomstats'

Version 0.0.1.9000

Description Provides an interface to the Python package 'Geomstats'
authored by Miolane et al. (2020) <[arXiv:2004.04667](https://arxiv.org/abs/2004.04667)>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/LMJL-Alea/rgeomstats>,
<https://lmjl-alea.github.io/rgeomstats/>

BugReports <https://github.com/LMJL-Alea/rgeomstats/issues>

Config/reticulate list(packages = list(list(package = ``geomstats"))
)

Imports cli, purrr, R6, Rdpack, reticulate, rlang

RdMacros Rdpack

Repository <https://lmjl-alea.r-universe.dev>

RemoteUrl <https://github.com/lmjl-alea/rgeomstats>

RemoteRef HEAD

RemoteSha 8e60931e78a4482b5efc753db0c371d31fb46d37

Contents

NFoldManifold	2
SPDMatrix	3
SPDMetricAffine	4
SPDMetricBuresWasserstein	5
SPDMetricEuclidean	6
SPDMetricLogEuclidean	7
SpecialOrthogonal	9

NFoldManifold	<i>Class for N-Fold Product Manifolds</i>
---------------	---

Description

Class for an n -fold product manifold M^n . It defines a manifold as the product manifold of n copies of a given base manifold M .

Super classes

`rgeomstats::PythonClass` -> `rgeomstats::Manifold` -> `NFoldManifold`

Methods

Public methods:

- `NFoldManifold$new()`
- `NFoldManifold$clone()`

Method `new()`: The `NFoldManifold` class constructor.

Usage:

```
NFoldManifold$new(
  base_manifold,
  n_copies,
  metric = NULL,
  default_coords_type = "intrinsic",
  py_cls = NULL
)
```

Arguments:

`base_manifold` An `R6::R6Class` specifying the base manifold to copy.

`n_copies` An integer value specifying the number of replication of the base manifold.

`metric` An `R6::R6Class` specifying the base metric to use. Defaults to `NULL` which uses the Riemannian metric.

`default_coords_type` A string specifying the coordinate type. Choices are "intrinsic" or "extrinsic". Defaults to "intrinsic".

`py_cls` A Python object of class `NFoldManifold`. Defaults to `NULL` in which case it is instantiated on the fly using the other input arguments.

Returns: A `NFoldManifold R6::R6Class` object.

Examples:

```
if (reticulate::py_module_available("geomstats")) {
  nfm <- NFoldManifold$new(
    base_manifold = SPDMatrix(n = 3),
    n_copies = 3
  )
  nfm
}
```

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
NFoldManifold$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Author(s)

Nicolas Guigui

Examples

```
## -----
## Method `NFoldManifold$new`
## -----

if (reticulate::py_module_available("geomstats")) {
  nfm <- NFoldManifold$new(
    base_manifold = SPDMatrix(n = 3),
    n_copies = 3
  )
  nfm
}
```

SPDMatrix

Class for the Manifold of Symmetric Positive Definite Matrices

Description

This function generates an instance of the class for the manifold of symmetric positive definite matrices $SPD(n)$.

Usage

```
SPDMatrix(n, ...)
```

Arguments

`n` An integer value specifying the number of rows and columns of the matrices.

`...` Extra arguments to be passed to parent class constructors. See [OpenSet](#) and [Manifold](#) classes.

Value

An object of class [SPDMatrixes](#).

Author(s)

Yann Thanwerdas

See Also

Other symmetric positive definite matrix classes: [SPDMatrixes](#)

Examples

```
if (reticulate::py_module_available("geomstats")) {
  spd3 <- SPDMatrix(n = 3)
  spd3
}
```

SPDMetricAffine	<i>Class for the Affine Metric on the Manifold of Symmetric Positive Definite Matrices</i>
-----------------	--

Description

An [R6::R6Class](#) object implementing the [SPDMetricAffine](#) class. This is the class for the affine-invariant metric on the SPD manifold (Thanwerdas and Pennec 2019).

Super classes

```
rgeomstats::PythonClass -> rgeomstats::Connection -> rgeomstats::RiemannianMetric
-> SPDMetricAffine
```

Public fields

`n` An integer value specifying the shape of the matrices: $n \times n$.

`power_affine` An integer value specifying the power transformation of the classical SPD metric.

Methods**Public methods:**

- [SPDMetricAffine\\$new\(\)](#)
- [SPDMetricAffine\\$clone\(\)](#)

Method `new()`: The [SPDMetricAffine](#) class constructor.

Usage:

```
SPDMetricAffine$new(n, power_affine = 1, py_cls = NULL)
```

Arguments:

`n` An integer value specifying the shape of the matrices: $n \times n$.

`power_affine` An integer value specifying the power transformation of the classical SPD metric. Defaults to 1L.

`py_cls` A Python object of class `SPDMetricAffine`. Defaults to NULL in which case it is instantiated on the fly using the other input arguments.

Returns: An object of class `SPDMetricAffine`.

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
SPDMetricAffine$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Author(s)

Yann Thanwerdas

References

Thanwerdas Y, Pennec X (2019). “Is affine-invariance well defined on SPD matrices? A principled continuum of metrics.” In *International Conference on Geometric Science of Information*, 502–510. Springer.

SPDMetricBuresWasserstein

Class for the Bures-Wasserstein Metric on the Manifold of Symmetric Positive Definite Matrices

Description

An `R6::R6Class` object implementing the `SPDMetricBuresWasserstein` class. This is the class for the Bures-Wasserstein metric on the SPD manifold (Bhatia et al. 2019; Malagò et al. 2018).

Super classes

```
rgeomstats::PythonClass -> rgeomstats::Connection -> rgeomstats::RiemannianMetric
-> SPDMetricBuresWasserstein
```

Public fields

`n` An integer value specifying the shape of the matrices: $n \times n$.

Methods

Public methods:

- `SPDMetricBuresWasserstein$new()`
- `SPDMetricBuresWasserstein$clone()`

Method `new()`: The `SPDMetricBuresWasserstein` class constructor.

Usage:

```
SPDMetricBuresWasserstein$new(n, py_cls = NULL)
```

Arguments:

`n` An integer value specifying the shape of the matrices: $n \times n$.

`py_cls` A Python object of class `SPDMetricBuresWasserstein`. Defaults to `NULL` in which case it is instantiated on the fly using the other input arguments.

Returns: An object of class `SPDMetricBuresWasserstein`.

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
SPDMetricBuresWasserstein$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Author(s)

Yann Thanwerdas

References

Bhatia R, Jain T, Lim Y (2019). “On the Bures–Wasserstein distance between positive definite matrices.” *Expositiones Mathematicae*, **37**(2), 165–191.

Malagò L, Montrucchio L, Pistone G (2018). “Wasserstein Riemannian geometry of Gaussian densities.” *Information Geometry*, **1**(2), 137–179.

SPDMetricEuclidean	<i>Class for the Euclidean Metric on the Manifold of Symmetric Positive Definite Matrices</i>
--------------------	---

Description

An `R6::R6Class` object implementing the `SPDMetricEuclidean` class. This is the class for the Euclidean metric on the SPD manifold.

Super classes

```
rgeomstats::PythonClass -> rgeomstats::Connection -> rgeomstats::RiemannianMetric
-> SPDMetricEuclidean
```

Public fields

`n` An integer value specifying the shape of the matrices: $n \times n$.

`power_euclidean` An integer value specifying the power transformation of the classical SPD metric.

Methods**Public methods:**

- [SPDMetricEuclidean\\$new\(\)](#)
- [SPDMetricEuclidean\\$clone\(\)](#)

Method `new()`: The [SPDMetricEuclidean](#) class constructor.

Usage:

```
SPDMetricEuclidean$new(n, power_euclidean = 1, py_cls = NULL)
```

Arguments:

`n` An integer value specifying the shape of the matrices: $n \times n$.

`power_euclidean` An integer value specifying the power transformation of the classical SPD metric. Defaults to 1L.

`py_cls` A Python object of class [SPDMetricEuclidean](#). Defaults to NULL in which case it is instantiated on the fly using the other input arguments.

Returns: An object of class [SPDMetricEuclidean](#).

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
SPDMetricEuclidean$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Author(s)

Yann Thanwerdas

SPDMetricLogEuclidean *Class for the log-Euclidean Metric on the Manifold of Symmetric Positive Definite Matrices*

Description

An [R6::R6Class](#) object implementing the [SPDMetricLogEuclidean](#) class. This is the class for the log-Euclidean metric on the SPD manifold.

Super classes

```
rgeomstats::PythonClass -> rgeomstats::Connection -> rgeomstats::RiemannianMetric
-> SPDMetricLogEuclidean
```

Public fields

`n` An integer value specifying the shape of the matrices: $n \times n$.

Methods

Public methods:

- `SPDMetricLogEuclidean$new()`
- `SPDMetricLogEuclidean$clone()`

Method `new()`: The `SPDMetricLogEuclidean` class constructor.

Usage:

```
SPDMetricLogEuclidean$new(n, py_cls = NULL)
```

Arguments:

`n` An integer value specifying the shape of the matrices: $n \times n$.

`py_cls` A Python object of class `SPDMetricLogEuclidean`. Defaults to `NULL` in which case it is instantiated on the fly using the other input arguments.

Returns: An object of class `SPDMetricLogEuclidean`.

Examples:

```
if (reticulate::py_module_available("geomstats")) {
  mt <- SPDMetricLogEuclidean$new(n = 3)
  mt
}
```

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
SPDMetricLogEuclidean$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Author(s)

Yann Thanwerdas

Examples

```
## -----
## Method `SPDMetricLogEuclidean$new`
## -----

if (reticulate::py_module_available("geomstats")) {
  mt <- SPDMetricLogEuclidean$new(n = 3)
  mt
}
```

SpecialOrthogonal *Class for the Special Orthogonal Group*

Description

This function generates an instance of the class for the special orthogonal group $SO(n)$.

Usage

```
SpecialOrthogonal(n, point_type = "matrix", epsilon = 0, ..., py_cls = NULL)
```

Arguments

n	An integer value representing the shape of the $n \times n$ matrices.
point_type	A character string specifying how elements of the group should be represented. Choices are either "vector" or "matrix". Defaults to "matrix".
epsilon	A numeric value specifying the precision to use for calculations involving potential division by 0 in rotations. Defaults to 0.0 .
...	Extra arguments to be passed to parent class constructors. See LieGroup , MatrixLieAlgebra , LevelSet and Manifold classes.
py_cls	A Python object of class SpecialOrthogonal. Defaults to NULL in which case it is instantiated on the fly using the other input arguments.

Value

An object of class [SpecialOrthogonal](#) which is an instance of one of three different `R6::R6Class` depending on the values of the input arguments. Specifically:

- if $n == 2$ and `point_type == "vector"`, then the user wants to instantiate the space of 2D rotations in vector representations and thus the output is an instance of the [SpecialOrthogonal2Vectors](#) class;
- if $n == 3$ and `point_type == "vector"`, then the user wants to instantiate the space of 3D rotations in vector representations and thus the output is an instance of the [SpecialOrthogonal3Vectors](#) class;
- in all other cases, either the user is dealing with rotations in matrix representation or with rotations in dimension greater than 3 and thus the output is an instance of the [SpecialOrthogonalMatrices](#) class.

Author(s)

Nicolas Guigui and Nina Miolane

See Also

Other special orthogonal classes: [SpecialOrthogonal2Vectors](#), [SpecialOrthogonal3Vectors](#), [SpecialOrthogonalMatrices](#)

Examples

```
if (reticulate::py_module_available("geomstats")) {  
  so3 <- SpecialOrthogonal(n = 3)  
  so3  
}
```

Index

- * **product manifold**
 - NFoldManifold, 2
- * **special orthogonal classes**
 - SpecialOrthogonal, 9
- * **symmetric positive definite matrix classes**
 - SPDMatrix, 3

LevelSet, 9
LieGroup, 9

Manifold, 3, 9
MatrixLieAlgebra, 9

NFoldManifold, 2, 2

OpenSet, 3

R6::R6Class, 2, 4–7, 9
rgeomstats::Connection, 4–7
rgeomstats::Manifold, 2
rgeomstats::RiemannianMetric, 4–7

SPDMatrices, 3, 4
SPDMatrix, 3
SPDMetricAffine, 4, 4, 5
SPDMetricBuresWasserstein, 5, 5, 6
SPDMetricEuclidean, 6, 6, 7
SPDMetricLogEuclidean, 7, 7, 8
SpecialOrthogonal, 9, 9
SpecialOrthogonal2Vectors, 9
SpecialOrthogonal3Vectors, 9
SpecialOrthogonalMatrices, 9