

Package ‘rgeomstats’

July 23, 2025

Title Interface to 'Geomstats'

Version 0.0.1

Description Provides an interface to the Python package 'Geomstats' authored by Miolane et al. (2020) <[doi:10.48550/arXiv.2004.04667](https://doi.org/10.48550/arXiv.2004.04667)>.

License MIT + file LICENSE

Encoding UTF-8

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

NeedsCompilation no

Author Aymeric Stamm [aut, cre] (ORCID:

<<https://orcid.org/0000-0002-8725-3654>>),

Nicolas Guigui [ctb] (Author of the Geomstats Python package, ORCID:

<<https://orcid.org/0000-0002-7901-0732>>),

Alice Le Brigant [ctb] (Author of the Geomstats Python package, ORCID:

<<https://orcid.org/0000-0002-8055-4753>>),

Johan Mathe [ctb] (Author of the Geomstats Python package),

Nina Miolane [ctb] (Author of the Geomstats Python package, ORCID:

<<https://orcid.org/0000-0002-1200-9024>>),

Xavier Pennec [ctb] (Author of the Geomstats Python package, ORCID:

<<https://orcid.org/0000-0002-6617-7664>>),

Luis Pereira [ctb] (Author of the Geomstats Python package),

Yann Thanwerdas [ctb] (Author of the Geomstats Python package, ORCID:

<<https://orcid.org/0000-0002-9351-6318>>)

Maintainer Aymeric Stamm <aymeric.stamm@math.cnrs.fr>
Repository CRAN
Date/Publication 2022-11-04 10:10:02 UTC

Contents

NFoldManifold	2
SPDMatrix	4
SPDMetricAffine	5
SPDMetricBuresWasserstein	6
SPDMetricEuclidean	7
SPDMetricLogEuclidean	8
SpecialOrthogonal	9
Index	11

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 $\text{SPD}(n)$.

Usage

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

Arguments

<code>n</code>	An integer value specifying the number of rows and columns of the matrices.
<code>...</code>	Extra arguments to be passed to parent class constructors. See OpenSet and Manifold classes.

Value

An object of class [SPDMatrices](#).

Author(s)

Yann Thanwerdas

See Also

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

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

<code>n</code>	An integer value representing the shape of the $n \times n$ matrices.
<code>point_type</code>	A character string specifying how elements of the group should be represented. Choices are either "vector" or "matrix". Defaults to "matrix".
<code>epsilon</code>	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 <code>SpecialOrthogonal</code> . Defaults to <code>NULL</code> 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, [4](#)

LevelSet, [10](#)
LieGroup, [10](#)

Manifold, [4](#), [10](#)
MatrixLieAlgebra, [10](#)

NFoldManifold, [2](#), [2](#), [3](#)

OpenSet, [4](#)

R6::R6Class, [2](#), [3](#), [5–8](#), [10](#)
rgeomstats::Connection, [5–8](#)
rgeomstats::Manifold, [2](#)
rgeomstats::RiemannianMetric, [5–8](#)

SPDMatrices, [4](#)
SPDMatrix, [4](#)
SPDMetricAffine, [5](#), [5](#)
SPDMetricBuresWasserstein, [6](#), [6](#)
SPDMetricEuclidean, [7](#), [7](#), [8](#)
SPDMetricLogEuclidean, [8](#), [8](#), [9](#)
SpecialOrthogonal, [9](#), [10](#)
SpecialOrthogonal2Vectors, [10](#)
SpecialOrthogonal3Vectors, [10](#)
SpecialOrthogonalMatrices, [10](#)