Package 'GeodRegr'

August 24, 2020

Type Package

Title Geodesic Regression

Version 0.1.0

Description

Provides a gradient descent algorithm to find a geodesic relationship between real-valued independent variables and a manifold-valued dependent variable (i.e. geodesic regression). Available manifolds are Euclidean space, the sphere, and Kendall's 2-dimensional shape space. Besides the standard least-squares loss, the least absolute deviations, Huber, and Tukey biweight loss functions can also be used to perform robust geodesic regression. Functions to help choose appropriate cutoff parameters to maintain high efficiency for the Huber and Tukey biweight estimators are included. The k-sphere is a k-dimensional manifold: we represent it as a sphere of radius 1 and center 0 embedded in (k+1)-dimensional space. Kendall's 2D shape space with K landmarks is of real dimension 2K-4; preshapes are represented as complex K-vectors with mean 0 and magnitude 1. Details are described in Shin, H.-Y. and Oh, H.-S. (2020) <arXiv:2007.04518>. Also see Fletcher, P. T. (2013) <doi:10.1007/s11263-012-0591-y> and Kim, H. J., Adluru, N., Collins, M. D., Chung, M. K., Bendin, B. B., Johnson, S. C., Davidson, R. J. and Singh, V. (2014) <doi:10.1109/CVPR.2014.352>.

Depends R (>= 4.0.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports zipfR (>= 0.6.66), stats (>= 3.6.2)

NeedsCompilation no

Author Ha-Young Shin [aut, cre],
Hee-Seok Oh [aut]

Maintainer Ha-Young Shin <hayoung.shin@gmail.com>

Repository CRAN

Date/Publication 2020-08-24 16:20:07 UTC

R topics documented:

`are` *Approximate ARE of an M-type estimator to the least-squares estimator*

Description

Approximate asymptotic relative efficiency (ARE) of an M-type estimator to the least-squares estimator given Gaussian errors, calculated using a tangent space approximation.

Usage

```
are(estimator, k, c = NULL)
```

Arguments

`estimator`	M-type estimator (`'l2'`, `'l1'`, `'huber'`, or `'tukey'`).
`k`	Dimension of the manifold.
`c`	A positive multiplier, or a vector of them, of σ, the square root of the variance, used in the cutoff parameter for the `'huber'` and `'tukey'` estimators; should be `NULL` for the `'l2'` or `'l1'` estimators.

Value

Approximate ARE

Author(s)

Ha-Young Shin

References

Shin, H.-Y. and Oh H.-S. (2020). Robust Geodesic Regression. <arXiv:2007.04518>

See Also

are_nr.

Examples

```
are('l1', 10)
```

---

are_nr *Newton-Raphson method for the* are *function*

Description

Finds the positive multiplier of σ, the square root of the variance, used in the cutoff parameter that will give the desired (approximate) level of efficiency for the provided M-type estimator. Does so by using are and its partial derivative with respect to c in the Newton-Raphson method.

Usage

```
are_nr(estimator, k, startingpoint, level = 0.95)
```

Arguments

estimator	M-type estimator ('huber' or 'tukey').
k	Dimension of the manifold.
startingpoint	Initial estimate for the Newton-Raphson method. May be determined after looking at a graph of the are function.
level	The desired ARE to the 'l2' estimator.

Details

As is often the case with the Newton-Raphson method, the starting point must be chosen carefully in order to ensure convergence. The use of the graph of the are function to find a starting point close to the root is recommended.

Value

Positive multiplier of σ, the square root of the variance, used in the cutoff parameter, to give the desired level of efficiency.

Author(s)

Ha-Young Shin

References

Shin, H.-Y. and Oh H.-S. (2020). Robust Geodesic Regression. <arXiv:2007.04518>

See Also

are.

Examples

```
dimension <- 4
x <- 1:10000 / 1000
# use a graph of the are function to pick a good starting point
plot(x, are('huber', dimension, x) - 0.95)
are_nr('huber', dimension, 2)
```

---

calvaria *Data on calvaria growth in rat skulls*

Description

Vilmann data for growth in rat calvariae, that is, upper skulls, for 21 rats. For each rat, the shape of the calavaria was measured at 8 different ages (7, 14, 21, 30, 40, 60, 90, and 150 days), for a total of 168 data points. The boundaries of the midsagittal sections of the rats' calvariae are each marked with 8 landmarks. The data points have been translated and scaled in order to make them preshapes.

Usage

```
data(calvaria)
```

Format

A named list containing x a vector containing the ages y a matrix where each column is a preshape. The 23rd, 101st, 104th, and 160th entries are corrupted)

Details

There are 4 corrupted data points: those corresponding to day 90 for the 3rd rat, day 40 and 150 for the 13th rat, and day 150 for the 20th rat (the 23rd, 101st, 104th, and 160th entries); one of the landmarks for each of these measurements has been entered as (9999, 9999) (before translation/scaling).

Source

Vilmann's rat data set from pp. 408-414 of Bookstein. Original data available at https://life.bio.sunysb.edu/morph/data/Book-VilmannRat.txt.

References

Bookstein, F. L. (1991). Morphometric Tools for Landmark Data: Geometry and Biology. Cambridge Univ, 408-414.

Hinkle, J., Muralidharan, P., Fletcher, P. T., Joshi, S. (2012). Polynomial Regression on Riemannian Manifolds. European Conference on Computer Vision, 1-14.

Examples

```
# we will test the robustness of each estimator by comparing their
# performance on the original (corrupted) data set to that of the L_2
# estimator on the uncorrupted data set (with the 4 problematic data points
# removed).

data(calvaria)

manifold <- 'kendall'

contam_x_data <- calvaria$x
contam_mean_x <- mean(contam_x_data)
contam_x_data <- contam_x_data - contam_mean_x # center x data
uncontam_x_data <- calvaria$x[ -c(23, 101, 104, 160)]
uncontam_mean_x <- mean(uncontam_x_data)
uncontam_x_data <- uncontam_x_data - uncontam_mean_x # center x data

contam_y_data <- calvaria$y
uncontam_y_data <- calvaria$y[, -c(23, 101, 104, 160)] # remove corrupted
    # columns

landmarks <- dim(contam_y_data)[1]
dimension <- 2 * landmarks - 4

# we ignore Huber's estimator as the L_1 estimator already has an
# (approximate) efficiency above 95% in 12 dimensions; see documentation for
# the are and are_nr functions

tol <- 1e-5
uncontam_l2 <- geo_reg(manifold, uncontam_x_data, uncontam_y_data,
    'l2', p_tol = tol, V_tol = tol)
contam_l2 <- geo_reg(manifold, contam_x_data, contam_y_data,
    'l2', p_tol = tol, V_tol = tol)
contam_l1 <- geo_reg(manifold, contam_x_data, contam_y_data,
    'l1', p_tol = tol, V_tol = tol)
contam_tukey <- geo_reg(manifold, contam_x_data, contam_y_data,
    'tukey', are_nr('tukey', dimension, 10, 0.99), p_tol = tol, V_tol = tol)

geodesics <- vector('list')
geodesics[[1]] <- uncontam_l2
geodesics[[2]] <- contam_l2
geodesics[[3]] <- contam_l1
geodesics[[4]] <- contam_tukey

loss(manifold, geodesics[[1]]$p, geodesics[[1]]$V, uncontam_x_data,
    uncontam_y_data, 'l2')
loss(manifold, geodesics[[2]]$p, geodesics[[2]]$V, contam_x_data,
    contam_y_data, 'l2')
loss(manifold, geodesics[[3]]$p, geodesics[[3]]$V, contam_x_data,
    contam_y_data, 'l1')
loss(manifold, geodesics[[4]]$p, geodesics[[4]]$V, contam_x_data,
    contam_y_data, 'tukey', are_nr('tukey', dimension, 10, 0.99))
```

```
# visualization of each geodesic

oldpar <- par(mfrow = c(1, 4))

days <- c(7, 14, 21, 30, 40, 60, 90, 150)
pal <- colorRampPalette(c("blue", "red"))(length(days))

# each predicted geodesic will be represented as a sequence of the predicted
# shapes at each of the above ages, the blue contour will show the predicted
# shape on day 7 and the red contour the predicted shape on day 150

contour <- vector('list')

for (i in 1:length(days)) {
  contour[[i]] <- exp_map(manifold, geodesics[[1]]$p, (days[i] -
      uncontam_mean_x) * geodesics[[1]]$V)
  contour[[i]] <- c(contour[[i]], contour[[i]][1])
}
plot(Re(contour[[length(days)]]), Im(contour[[length(days)]]), type = 'n',
    xaxt = 'n', yaxt = 'n', ann = FALSE, asp = 1)
 for (i in 1:length(days)) {
   lines(Re(contour[[i]]), Im(contour[[i]]), col = pal[i])
}
for (j in 2:4) {
  for (i in 1:length(days)) {
    contour[[i]] <- exp_map(manifold, geodesics[[j]]$p, (days[i] -
        contam_mean_x) * geodesics[[j]]$V)
    contour[[i]] <- c(contour[[i]], contour[[i]][1])
  }
  plot(Re(contour[[length(days)]]), Im(contour[[length(days)]]), type = 'n',
      xaxt = 'n', yaxt = 'n', ann = FALSE, asp = 1)
  for (i in 1:length(days)) {
    lines(Re(contour[[i]]), Im(contour[[i]]), col = pal[i])
  }
}
# even with a mere 4 corrupted landmarks out of a total of 8 * 168 = 1344, we
# can clearly see that contam_l2, the second image, looks slightly
# different from all the others, especially near the top of the image.

par(oldpar)
```

---

exp_map *Exponential map*

Description

Performs the exponential map $\text{Exp}_p(v)$ on the given manifold.

Usage

```
exp_map(manifold, p, v)
```

Arguments

manifold	Type of manifold ('euclidean', 'sphere', or 'kendall').
p	A vector (or column matrix) representing a point on the manifold.
v	A vector (or column matrix) tangent to p.

Value

A vector representing a point on the manifold.

Author(s)

Ha-Young Shin

References

Fletcher, P. T. (2013). Geodesic regression and the theory of least squares on Riemannian manifolds. International Journal of Computer Vision, 105, 171-185.

Cornea, E., Zhu, H., Kim, P. and Ibrahim, J. G. (2017). Regression models on Riemannian symmetric spaces. Journal of the Royal Statistical Society: Series B, 79, 463-482.

Shin, H.-Y. and Oh H.-S. (2020). Robust Geodesic Regression. <arXiv:2007.04518>

See Also

log_map.

Examples

```
exp_map('sphere', c(1, 0, 0, 0, 0), c(0, 0, pi / 4, 0, 0))
```

---

geo_dist *Geodesic distance between two points on a manifold*

Description

Finds the Riemannian distance $d(p_1, p_2) = ||\text{Log}_{p_1}(p_2)||$ between two points on the given manifold, provided p_2 is in the domain of Log_{p_1}.

Usage

```
geo_dist(manifold, p1, p2)
```

Arguments

manifold	Type of manifold ('euclidean', 'sphere', or 'kendall').
p1	A vector (or column matrix) representing a point on the manifold.
p2	A vector (or column matrix) representing a point on the manifold.

Details

On the sphere, $-p_1$ is not in the domain of Log_{p_1}.

Value

Riemannian distance between p1 and p2.

Author(s)

Ha-Young Shin

See Also

log_map.

Examples

```
p1 <- matrix(rnorm(10), ncol = 2)
p1 <- p1[, 1] + (1i) * p1[, 2]
p1 <- (p1 - mean(p1)) / norm(p1 - mean(p1), type = '2')
p2 <- matrix(rnorm(10), ncol = 2)
p2 <- p2[, 1] + (1i) * p2[, 2]
p2 <- (p2 - mean(p2)) / norm(p2 - mean(p2), type = '2')
geo_dist('kendall', p1, p2)
```

---

geo_reg *Gradient descent for (robust) geodesic regression*

Description

Finds $\mathrm{argmin}_{(p,V)\in M\times T_pM^n} \sum_{i=1}^{N} \rho(d(\mathrm{Exp}(p, Vx_i), y_i))$ through a gradient descent algorithm.

Usage

```
geo_reg(
  manifold,
  x,
  y,
  estimator,
  c = NULL,
  p_tol = 1e-05,
```

```
  V_tol = 1e-05,
  max_iter = 1e+05
)
```

Arguments

manifold	Type of manifold ('euclidean', 'sphere', or 'kendall').
x	A vector, matrix, or data frame of independent variables; for matrices and data frames, the rows and columns represent the subjects and independent variables, respectively.
y	A matrix or data frame whose columns represent points on the manifold.
estimator	M-type estimator ('l2', 'l1', 'huber', or 'tukey').
c	Multiplier of σ, the square root of the variance, used in the cutoff parameter for the 'huber' and 'tukey' estimators; should be NULL for the 'l2' or 'l1' estimators.
p_tol	Termination condition for the distance between consecutive updates of p.
V_tol	Termination condition for the distance between columns of consecutive updates of V, parallel transported to be in the same tangent space. Can be a vector of positive real numbers for each independent variable or a single positive number.
max_iter	Maximum number of gradient descent steps before ending the algorithm.

Details

Each column of x should be centered to have an average of 0 for the quickest and most accurate results. If all of the elements of a column of x are equal, the resulting vector will consist of NAs. In the case of the 'sphere', an error will be raised if all points are on a pair of antipodes.

Value

A named list containing

p	a vector representing the estimate of the initial point on the manifold
V	a matrix representing the estimate of the initial velocities for each independent variable; the columns represent the independent variables.
iteration	number of gradient descent steps taken.

Author(s)

Ha-Young Shin

References

Fletcher, P. T. (2013). Geodesic regression and the theory of least squares on Riemannian manifolds. International Journal of Computer Vision, 105, 171-185.

Kim, H. J., Adluru, N., Collins, M. D., Chung, M. K., Bendin, B. B., Johnson, S. C., Davidson, R. J. and Singh, V. (2014). Multivariate general linear models (MGLM) on Riemannian manifolds

with applications to statistical analysis of diffusion weighted images. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2705-2712.

Shin, H.-Y. and Oh H.-S. (2020). Robust Geodesic Regression. <arXiv:2007.04518>

See Also

intrinsic_location.

Examples

```
# an example of multiple regression with two independent variables, with 64
# data points

x <- matrix(runif(2 * 64), ncol = 2)
x <- t(t(x) - colMeans(x))
y <- matrix(0L, nrow = 4, ncol = 64)
for (i in 1:64) {
  y[, i] <- exp_map('sphere', c(1, 0, 0, 0), c(0, runif(1), runif(1),
      runif(1)))
}
geo_reg('sphere', x, y, 'tukey', c = are_nr('tukey', 2, 6))
```

---

intrinsic_location *Gradient descent for location based on M-type estimators*

Description

Finds $\operatorname{argmin}_{p \in M} \sum_{i=1}^{N} \rho(d(p, y_i))$ through a gradient descent algorithm.

Usage

```
intrinsic_location(
  manifold,
  y,
  estimator,
  c = NULL,
  p_tol = 1e-05,
  V_tol = 1e-05,
  max_iter = 1e+05
)
```

Arguments

manifold	Type of manifold ('euclidean', 'sphere', or 'kendall').
y	A matrix or data frame whose columns represent points on the manifold.
estimator	M-type estimator ('l2', 'l1', 'huber', or 'tukey').

c	Multiplier of σ, the square root of the variance, used in the cutoff parameter for the 'huber' and 'tukey' estimators; should be NULL for the 'l2' or 'l1' estimators.
p_tol	Termination condition for the distance between consecutive updates of p.
V_tol	Termination condition for the distance between columns of consecutive updates of V, parallel transported to be in the same tangent space. Can be a vector of positive real numbers for each independent variable or a single positive number.
max_iter	Maximum number of gradient descent steps before ending the algorithm.

Details

In the case of the 'sphere', an error will be raised if all points are on a pair of antipodes.

Value

A vector representing the location estimate

Author(s)

Ha-Young Shin

References

Fletcher, P. T. (2013). Geodesic regression and the theory of least squares on Riemannian manifolds. International Journal of Computer Vision, 105, 171-185.

Kim, H. J., Adluru, N., Collins, M. D., Chung, M. K., Bendin, B. B., Johnson, S. C., Davidson, R. J. and Singh, V. (2014). Multivariate general linear models (MGLM) on Riemannian manifolds with applications to statistical analysis of diffusion weighted images. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2705-2712.

Shin, H.-Y. and Oh H.-S. (2020). Robust Geodesic Regression. <arXiv:2007.04518>

See Also

geo_reg, rbase.mean, rbase.median.

Examples

```
y <- matrix(runif(100, 1000, 2000), nrow = 10)
intrinsic_location('euclidean', y, 'l2')
```

log_map *Logarithm map*

Description

Performs the logarithm map $\text{Log}_{p_1}(p_2)$ on the given manifold, provided p_2 is in the domain of Log_{p_1}.

Usage

```
log_map(manifold, p1, p2)
```

Arguments

manifold	Type of manifold (`'euclidean'`, `'sphere'`, or `'kendall'`).
p1	A vector (or column matrix) representing a point on the manifold.
p2	A vector (or column matrix) representing a point on the manifold.

Details

On the sphere, $-p_1$ is not in the domain of Log_{p_1}.

Value

A vector tangent to p1.

Author(s)

Ha-Young Shin

References

Fletcher, P. T. (2013). Geodesic regression and the theory of least squares on Riemannian manifolds. International Journal of Computer Vision, 105, 171-185.

Cornea, E., Zhu, H., Kim, P. and Ibrahim, J. G. (2017). Regression models on Riemannian symmetric spaces. Journal of the Royal Statistical Society: Series B, 79, 463-482.

Shin, H.-Y. and Oh H.-S. (2020). Robust Geodesic Regression. <arXiv:2007.04518>

See Also

exp_map, geo_dist.

Examples

```
log_map('sphere', c(0, 1, 0, 0), c(0, 0, 1, 0))
```

loss *Loss*

Description

Loss for a given p and V.

Usage

```
loss(manifold, p, V, x, y, estimator, cutoff = NULL)
```

Arguments

manifold	Type of manifold ('euclidean', 'sphere', or 'kendall').
p	A vector (or column matrix) on the manifold.
V	A matrix (or vector) where each column is a vector in the tangent space at p.
x	A matrix or data frame of independent variables; for matrices and data frames, the rows and columns represent the subjects and independent variables, respectively.
y	A matrix or data frame (or vector) whose columns represent points on the manifold.
estimator	M-type estimator ('l2', 'l1', 'huber', or 'tukey').
cutoff	Cutoff parameter for the 'huber' and 'tukey' estimators; should be NULL for the 'l2' or 'l1' estimators.

Value

Loss.

Author(s)

Ha-Young Shin

Examples

```
y <- matrix(0L, nrow = 3, ncol = 64)
for (i in 1:64) {
  y[, i] <- exp_map('sphere', c(1, 0, 0), c(0, runif(1), runif(1)))
}
intrinsic_mean <- intrinsic_location('sphere', y, 'l2')
loss('sphere', intrinsic_mean, numeric(3), numeric(64), y, 'l2')
```

`onmanifold` *Manifold check and projection*

Description

Checks whether each data point in y is on the given manifold, and if not, provides a modified version of y where each column has been projected onto the manifold.

Usage

```
onmanifold(manifold, y)
```

Arguments

`manifold`	Type of manifold (`'euclidean'`, `'sphere'`, or `'kendall'`).
`y`	A vector, matrix, or data frame whose columns should represent points on the manifold.

Value

A named list containing

`on`	a logical vector describing whether or not each column of `y` is on the manifold.
`data`	a matrix of data frame of the same dimensions as `y`; each column of `y` has been projected onto the manifold.

Author(s)

Ha-Young Shin

Examples

```
y1 <- matrix(rnorm(10), ncol = 2)
y1 <- y1[, 1] + (1i) * y1[, 2]
y2 <- matrix(rnorm(10), ncol = 2)
y2 <- y2[, 1] + (1i) * y2[, 2]
y3 <- matrix(rnorm(10), ncol = 2)
y3 <- y3[, 1] + (1i) * y3[, 2]
y3 <- (y3 - mean(y3)) / norm(y3 - mean(y3), type = '2') # project onto preshape space
y <- matrix(c(y1, y2, y3), ncol = 3)
onmanifold('kendall', y)
```

par_trans *Parallel transport*

Description

Performs $\Gamma_{p_1 \to p_2}(v)$, parallel transport along the unique minimizing geodesic connecting p_1 and p_2, if it exists, on the given manifold.

Usage

```
par_trans(manifold, p1, p2, v)
```

Arguments

manifold	Type of manifold ('euclidean', 'sphere', or 'kendall').
p1	A vector (or column matrix) representing a point on the manifold.
p2	A vector (or column matrix) representing a point on the manifold.
v	A vector (or column matrix) tangent to p1.

Details

On the sphere, there is no unique minimizing geodesic connecting p_1 and $-p_1$.

Value

A vector tangent to p2.

Author(s)

Ha-Young Shin

References

Fletcher, P. T. (2013). Geodesic regression and the theory of least squares on Riemannian manifolds. International Journal of Computer Vision, 105, 171-185.

Cornea, E., Zhu, H., Kim, P. and Ibrahim, J. G. (2017). Regression models on Riemannian symmetric spaces. Journal of the Royal Statistical Society: Series B, 79, 463-482.

Shin, H.-Y. and Oh H.-S. (2020). Robust Geodesic Regression. <arXiv:2007.04518>

Examples

```
p1 <- matrix(rnorm(10), ncol = 2)
p1 <- p1[, 1] + (1i) * p1[, 2]
p1 <- (p1 - mean(p1)) / norm(p1 - mean(p1), type = '2') # project onto pre-shape space
p2 <- matrix(rnorm(10), ncol = 2)
p2 <- p2[, 1] + (1i) * p2[, 2]
p2 <- (p2 - mean(p2)) / norm(p2 - mean(p2), type = '2') # project onto pre-shape space
```

```
p3 <- matrix(rnorm(10), ncol = 2)
p3 <- p3[, 1] + (1i) * p3[, 2]
p3 <- (p3 - mean(p3)) / norm(p3 - mean(p3), type = '2') # project onto pre-shape space
v <- log_map('kendall', p1, p3)
par_trans('kendall', p1, p2, v)
```

Index