Package 'NITPicker'

January 10, 2019

Type Package

Title Finds the Best Subset of Points to Sample

Version 1.0.1

Author Daphne Ezer

Maintainer Daphne Ezer <dezer@turing.ac.uk>

Description Given a few examples of experiments over a time (or spatial) course, 'NITPicker' selects a subset of points to sample in follow-up experiments, which would (i) best distinguish between the experimental conditions and the control condition (ii) best distinguish between two models of how the experimental condition might differ from the control (iii) a combination of the two. Ezer and Keir (2018) <doi:10.1101/301796>.

License GPL (>= 2)

Encoding UTF-8

Depends fdasrvf, fda, stats, fda.usc

URL https://doi.org/10.1101/301796, https://daphneezer.wordpress.com

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-10 12:50:07 UTC

R topics documented:

findPathF1 *Find best subset of points for follow-up experiments, using F1 metric*

Description

findPathF1 finds the best subset of points to sample from a time course (or spatial axis, along a single axis), based on a set of example curves. Specifically, it finds subsets of points that estimate the shape of the curve effectively.

Usage

```
findPathF1(tp, training, numSubSamples, spline = 1,
  resampleTraining = T, iter = 20, knots = 100, numPerts = 1000,
  fast = T, mult = F, weights = c())
```

Arguments

`tp`	A numerical vector of time points (or spatial coordinates along a single axis)
`training`	this is a numerical matrix of training data, where the rows represent different samples, columns represent different time points (or points on a single spatial axis), and the values correspond to measurements. (If `mult==TRUE`, then this is instead a list of training matrices)
`numSubSamples`	integer that represents the number of time points that will be subsampled
`spline`	A positive integer representing the spline used to interpolate between knots when generating perturbations. Note that this does NOT designate the spline used when calculating the L2-error.
`resampleTraining`	A boolean designating whether the exact training data should be used (False) or whether a probability distribution of curves should be generated and training curves resampled (True).
`iter`	A positive integer, representing the maximum number of iterations employed during time warping (see time_warping in fdasrvf library)
`knots`	A positive integer– for time warping to work optimally, the points must be evenly sampled. This determines how many points do we evenly sample before conducting time warping
`numPerts`	a positive integer, representing the number of sampled curves to output.
`fast`	is a boolean, which determines whether the algorithm runs in fast mode where the sum of the perturbations is calculated prior to integration.
`mult`	is a boolean. If mult is true, then training will be a list of training matrices. This will be the case if there are multiple genes to consider at the same time. Training sets will be normalised by the size of the L2-error.
`weights`	is a vector of numbers that is the same length as the number of training curves. This describes the relative importance of these curves.

Value

An integer vector of the indices of the time points selected to be subsampled. The actual time points can be found by `tp[output]`. The length of this vector should be `numSubSamples`.

Examples

```
#load data:
#matrix with 12 rows, representing months (time)
#and 35 columns, representing cities (experiments)
mat=CanadianWeather$monthlyTemp
#find a set of points that help predict the shape of the curve:
a=findPathF1(c(1:12), mat, 5, numPerts=3) #make numPerts>=20 for real data
print(a) #indices of months to select for follow-up experiments
print(rownames(CanadianWeather$monthlyTemp)[a]) #month names selected
```

`findPathF2` *Find best subset of points for follow-up experiments, using F2 metric*

Description

findPathF2 finds the best subset of points to sample from a time course (or spatial axis, along a single axis), based on a set of example curves. Specifically, it compares between a control curve and a set of experimental curves.

Usage

```
findPathF2(tp, y, training, numSubSamples, spline = 1,
  resampleTraining = T, iter = 20, knots = 100, numPerts = 1000,
  fast = T, mult = F, weights = c())
```

Arguments

`tp` A numerical vector of time points (or spatial coordinates along a single axis)

`y` A numerical vector of measurements (of the control). If `mult==TRUE`, then this will be a matrix, where each column would be the y that corresponds with each training matrix.

`training` This is a numerical matrix of training data, where the rows represent different samples, columns represent different time points (or points on a single spatial axis), and the values correspond to measurements. (If `mult==TRUE`, then this is instead a list of training matrices).

`numSubSamples` integer that represents the number of time points that will be subsampled `spline` A positive integer representing the spline used to interpolate between knots when generating perturbations. Note that this does NOT designate the spline used when calculating the L2-error.

`resampleTraining` A boolean designating whether the exact training data should be used (False) or whether a probability distribution of curves should be generated and training curves resampled (True).

`iter` A positive integer, representing the maximum number of iterations employed during time warping (see time_warping in fdasrvf library)

`knots` A positive integer– for time warping to work optimally, the points must be evenly sampled. This determines how many points do we evenly sample before conducting time warping

`numPerts` a positive integer, representing the number of sampled curves to output.

`fast` is a boolean, which determines whether the algorithm runs in fast mode where the sum of the perturbations is calculated prior to integration.

`mult` is a boolean, which will determine whether multiple genes are considered at once.

`weights` is a vector of numbers that is the same length as the number of training curves. This describes the relative importance of these curves.

Value

An integer vector of the indices of the time points selected to be subsampled. The actual time points can be found by `tp[output]`. The length of this vector should be `numSubSamples`.

Examples

```
#load data:
# a matrix with 12 rows, representing months (time)
# and 35 columns, representing cities (experiments)
mat=CanadianWeather$monthlyTemp
y=CanadianWeather$monthlyTemp[,"Resolute"]
#find a set of points that help predict the shape of the curve
a=findPathF2(c(1:12), y, mat, 5, numPerts=3) #make numPerts>=20 for real data
print(a) #indices of months to select for follow-up experiments
print(rownames(CanadianWeather$monthlyTemp)[a]) #month names selected
```

`findPathF3` *Find best subset of points for follow-up experiments, using F3 metric*

Description

findPathF3 finds the best subset of points to sample from a time course (or spatial axis, along a single axis), based on a set of example curves. Specifically, it finds subsets of points that estimate the shape of the curve, normalised by the variance.

Usage

```
findPathF3(tp, training1, training2, numSubSamples, spline = 1,
  resampleTraining = F, iter = 20, knots = 100, numPerts = 1000,
  fast = T)
```

Arguments

`tp` A numerical vector of time points (or spatial coordinates along a single axis)

`training1` this is a numerical matrix of training data of experimental condition 1, where the rows represent different samples, columns represent different time points (or points on a single spatial axis), and the values correspond to measurements.

`training2` this is a numerical matrix of training data of experimental condition 2, where the rows represent different samples, columns represent different time points (or points on a single spatial axis), and the values correspond to measurements.

`numSubSamples` integer that represents the number of time points that will be subsampled

`spline` A positive integer representing the spline used to interpolate between knots when generating perturbations. Note that this does NOT designate the spline used when calculating the L2-error.

`resampleTraining`
A boolean designating whether the exact training data should be used (False) or whether a probability distribution of curves should be generated and training curves resampled (True).

`iter` A positive integer, representing the maximum number of iterations employed during time warping (see time_warping in fdasrvf library)

`knots` A positive integer– for time warping to work optimally, the points must be evenly sampled. This determines how many points do we evenly sample before conducting time warping

`numPerts` a positive integer, representing the number of sampled curves to output.

`fast` is a boolean, which determines whether the algorithm runs in fast mode where the sum of the perturbations is calculated prior to integration.

Value

An integer vector of the indices of the time points selected to be subsampled. The actual time points can be found by `tp[output]`. The length of this vector should be `numSubSamples`.

Examples

```
#Set up data:
namAtlantic=CanadianWeather$region[as.character(colnames(CanadianWeather$monthlyTemp))]
atlanticCities=which(namAtlantic=="Atlantic")
matAtlantic=CanadianWeather$monthlyTemp[, names(atlanticCities)]

namContinental=CanadianWeather$region[as.character(colnames(CanadianWeather$monthlyTemp))]
continentalCities=which(namContinental=="Continental")
```

```
matContinental=CanadianWeather$monthlyTemp[, names(continentalCities)]

#find a set of points that helps capture the difference
#between Atlantic and Continental cities, normalised by the variance
#make numPerts >=20 for real data
a=findPathF3(c(1:12),  matAtlantic,  matContinental, 5, numPerts=3)
print(a) #indices of months to select for follow-up experiments
print(rownames(CanadianWeather$monthlyTemp)[a]) #month names selected
```

generatePerturbations *Generate Perturbations*

Description

Find curves similar to a set of example curves. This function takes as input a set of example curves, and uses them to infer a probability distribution of curves. numPert curves are sampled from this probability distribution.

Usage

```
generatePerturbations(training, tp, iterations = 20, spline = 3,
  knots = 100, numPert = 20)
```

Arguments

training	This is a numerical matrix of training data, where the rows represent different samples, columns represent different time points (or points on a single spatial axis), and the values correspond to measurements
tp	A numerical vector of time points (or spatial coordinates along a single axis)
iterations	a positive integer, representing the maximum number of iterations employed during time warping (see time_warping in fdasrvf library)
spline	a positive integer, representing the degree of the B-spline interpolation when calculating values at the new, evenly spaced knot positions
knots	a positive integer– for time warping to work optimally, the points must be evenly sampled. This determines how many points do we evenly sample before conducting time warping
numPert	a positive integer, representing the number of sampled curves to output.

Value

An fdawarp object (see fdasrvf library)

Examples

```
mat=CanadianWeather$monthlyTemp
generated=generatePerturbations(mat, c(1:length(mat[,1])))
```

`L2` *L2-error*

Description

Given two functions y1(t) and y2(t), this function finds the L2-distance between the following two curves: a) y1(t)-y2(t) sampled at all time points (`tp`) b) y1(t)-y2(t) sampled at the time points indexed by `index` (`tp[index]`). Note that by setting `y2` to `rep(0,length(tp))`, this function can be used to estimate the L2-error in the shape of `y1`.

Usage

```
L2(tp, y1, y2, start, stop, index, numSubdivisions = 2000)
```

Arguments

`tp`	A numerical vector of time points (or spatial coordinates along a single axis)
`y1`	A numerical vector of measurements (of the control)
`y2`	A numerical vector of measurements (of the experimental condition)
`start`	A numerical value representing the start time (or spatial coordinate) of the integration
`stop`	A numerical value representing the end time (or spatial coordinate) of the integration
`index`	A vector of positive integers representing the indices of `tp` that we subsample
`numSubdivisions`	This can be adjusted to ensure the integration doesn't take too long, especially if we aren't overly concerned with rounding errors.

Value

A numeric value– the L2 error.

Index