Package ‘princurve’

February 20, 2015

Version 1.1-12
Title Fits a Principal Curve in Arbitrary Dimension
Author S original by Trevor Hastie <trevor@research.att.com> R port by Andreas Weingessel <Andreas.Weingessel@ci.tuwien.ac.at>
Maintainer Andreas Weingessel <Andreas.Weingessel@ci.tuwien.ac.at>
Description fits a principal curve to a data matrix in arbitrary dimensions
License GPL-2
Imports stats, graphics
NeedsCompilation yes
Repository CRAN
Date/Publication 2013-04-25 10:01:27

R topics documented:

  get.lam ................................................................. 1
  principal.curve ..................................................... 2

Index

  get.lam

Description

Finds the projection index for a matrix of points x, when projected onto a curve s. The curve need not be of the same length as the number of points. If the points on the curve are not in order, this order needs to be given as well, in tag.

Usage

  get.lam(x, s, tag, stretch = 2)
principal.curve

Arguments

- `x`: a matrix of data points
- `s`: a parametrized curve, represented by a polygon.
- `tag`: the order of the point in `s`. Default is the given order.
- `stretch`: A stretch factor for the endpoints of the curve; a maximum of 2. It allows the curve to grow, if required, and helps avoid bunching at the end.

Value

A structure is returned which represents a fitted curve. It has components

- `s`: The fitted points on the curve corresponding to each point `x`.
- `tag`: the order of the fitted points
- `lambda`: The projection index for each point
- `dist`: The total squared distance from the curve

See Also

principal.curve

principal.curve  Fit a Principal Curve

Description

Fits a principal curve which describes a smooth curve that passes through the middle of the data `x` in an orthogonal sense. This curve is a nonparametric generalization of a linear principal component. If a closed curve is fit (using smoother = "periodic.lowess") then the starting curve defaults to a circle, and each fit is followed by a bias correction suggested by J. Banfield.

Usage

```
principal.curve(x, start=NULL, thresh=0.001, plot.true=FALSE, maxit=10,
               stretch=2, smoother="smooth.spline", trace=FALSE, ...)
```

Arguments

- `x`: a matrix of points in arbitrary dimension
- `start`: either a previously fit principal curve, or else a matrix of points that in row order define a starting curve. If missing or NULL, then the first principal component is used. If the smoother is "periodic.lowess", then a circle is used as the start.
- `thresh`: convergence threshold on shortest distances to the curve.
- `plot.true`: If TRUE the iterations are plotted.
- `maxit`: maximum number of iterations.
**principal.curve**

stretch  a factor by which the curve can be extrapolated when points are projected. Default is 2 (times the last segment length). The default is 0 for smoother equal to "periodic.lowess".

smoother  choice of smoother. The default is "smooth.spline", and other choices are "lowess" and "periodic.lowess". The latter allows one to fit closed curves. Beware, you may want to use iter = 0 with lowess().

trace  If TRUE, the iteration information is printed

...  additional arguments to the smoothers

**Value**

An object of class "principal.curve" is returned. For this object the following generic methods a currently available: plot, points,lines.

It has components:

- s  a matrix corresponding to x, giving their projections onto the curve.
- tag  an index, such that s[tag, ] is smooth.
- lambda  for each point, its arc-length from the beginning of the curve. The curve is parametrized approximately by arc-length, and hence is unit-speed.
- dist  the sum-of-squared distances from the points to their projections.
- converged  A logical indicating whether the algorithm converged or not.
- nbrofIterations  Number of iterations completed before returning.
- call  the call that created this object; allows it to be updated().

**References**

"Principal Curves" by Hastie, T. and Stuetzle, W. 1989, JASA. See also Banfield and Raftery (JASA, 1992).

**See Also**

get.lam

**Examples**

```r
x <- runif(100, -1, 1)
x <- cbind(x, x ^ 2 + rnorm(100, sd = 0.1))
fit1 <- principal.curve(x, plot = TRUE)
fit2 <- principal.curve(x, plot = TRUE, smoother = "lowess")
lines(fit1)
points(fit1)
plot(fit1)
whiskers <- function(from, to)
    segments(from[, 1], from[, 2], to[, 1], to[, 2])
whiskers(x, fit1$s)
```
Index

*Topic nonparametric
  get.lam, 1
  principal.curve, 2

*Topic regression
  get.lam, 1
  principal.curve, 2

*Topic smooth
  get.lam, 1
  principal.curve, 2

get.lam, 1, 3

lines.principal.curve
  (principal.curve), 2

plot.principal.curve (principal.curve),
  2

points.principal.curve
  (principal.curve), 2

principal.curve, 2, 2