Package ‘monfuncreg’

March 28, 2018

Type Package

Title Monotone Nonparametric Regression for Functional/Longitudinal Data

Version 1.0

Date 2018-03-15

Author Ziqi Chen, Hongtu Zhu

Maintainer Ziqi Chen <chenzq453@gmail.com>

Description
Monotone Nonparametric Regression for Mean Function in Functional/Longitudinal Models.

License GPL (>= 2)

LinkingTo Rcpp, RcppArmadillo

R topics documented:

`monfuncreg` *Monotone Nonparametric Regression for Functional/Longitudinal Data*

Description

monfuncreg provides a increasing monotone estimator of the mean regression function in functional/longitudinal models.

Usage

```
monfuncreg(x, y, NN, N, hr, hd, weight="OBS", t)
```

Arguments

`x`	vector containing the x-values (design points) of a sample, scaled x to [0,1] $x = (s_{11}, ..., s_{1m_1}, ..., s_{nm_n})$.
`y`	vector containing the y-values (response) of a sample, $y = (y_{11}, ..., y_{1m_1}, ..., y_{nm_n})$.
`NN`	the number of observations for each subject, $NN = (m_1, ..., m_n)$.
`N`	the number of evaluation points of the unconstrained nonparametric regression estimator.
`hr`	bandwith of kernel K_r of the regression estimation step.
`hd`	bandwith of kernel K_d of the density estimation step.
`weight`	"OBS" or "SUBJ".
`t`	vector of points where the monotone estimation is computed, which is on [0,1].

Details

Functional/Longitudinal data analysis has wide application in the biomedical, psychometric and environmental sciences (Fitzmaurice et al., 2004; Yao et al., 2005; Wu and Zhang, 2006; Wang et al., 2016; Zhu et al., 2018). In this type of analysis, subjects are repeatedly measured over time, and measurements from the same subject are usually highly correlated. Let m_i be the number of repeated measurements for subject i and n be the total number of subjects. The observations from each subject are assumed to be noisy discrete realizations of an underlying process $X(\cdot)$and given by

$$y_{ij} = X_i(s_{ij}) + \sigma(s_{ij})\varepsilon_{ij} \;\text{ for }\; j = 1, \ldots, m_i;\; i = 1, \ldots, n,$$

where $y_{ij}$is the response variable of interest for subject i measured at time s_{ij}, the $X_i(\cdot)$'s are independent realizations of the underlying process $\{X(\cdot)\}$, and the ε_{ij}'s are random errors with zero mean and variance of 1. By using a mixed effects approach, we decompose $X_i(s_{ij})$ into an unknown population mean $m(\cdot) = E\{X_i(\cdot)\}$ and a subject-specific trajectory $\eta_i(\cdot)$ with zero mean and covariance function $\gamma(s,t) = \text{cov}\{\eta_i(s), \eta_i(t)\}$. Then, we can rewrite the model as

$$y_{ij} = m(s_{ij}) + \eta_i(s_{ij}) + \sigma(s_{ij})\varepsilon_{ij} \;\text{ for }\; j = 1, \ldots, m_i;\; i = 1, \ldots, n.$$

To obtain an estimator of $m^{-1}(t)$, we need an unconstrained estimator of $m(t)$, denoted as $\widehat{m}(s)$, as follows:

$$\widehat{m}(s) = \frac{\sum_{i=1}^{n} \omega_i \sum_{j=1}^{m_i} K_{r,h_r}(s_{ij} - s) Y_{ij}}{\sum_{i=1}^{n} \omega_i \sum_{j=1}^{m_i} K_{r,h_r}(s_{ij} - s)},$$

where the ω_i's are weights satisfying $\sum_{i=1}^{n} m_i \omega_i = 1$. We consider two commonly used weighting schemes, OBS for equal weight per observation and SUBJ for equal weight per subject (Yao et al., 2005; Li and Hsing, 2010; Kim and Zhao, 2012; Zhang and Wang, 2016). Specifically, we set $\omega_i = 1/(\sum_{i=1}^{n} m_i)$ for OBS, whereas we set $\omega_i = 1/(nm_i)$ for SUBJ. Moreover, $\widehat{m}(s)$ is a local constant estimator of $m(\cdot)$. By plugging $\widehat{m}(s)$, we obtain

$$\widehat{m}_I^{-1}(t) = N^{-1} \int_{-\infty}^{t} \sum_{i=1}^{N} K_{d,h_d}(\widehat{m}(i/N) - u) du.$$

Our constrained estimator of $m(s)$, denoted as $\widehat{m}_I(s)$, is then calculated by using a numerical inversion (see also, Dette, et al., 2006).

Value

`monfuncreg` returns a list of values

`mon1$variable`
: the points, for which the monotone function values will be estimated

`mon1$estimate`
: the monotone estimate at mon1$variable

References

Monotone Nonparametric Regression for Functional/Longitudinal Data, by Chen, Gao, Fu and Zhu, 2018

Examples

```
# This example is analyzing the real grey matter volume data
# obtained from the ADNI study using our model, method and package.
library(monfuncreg)
shuju=read.csv("Origdata1.csv",head=F,quote="")
YY=list()
TT=list()
jishu=0
for(i in 1:length(shuju[,1])){
  if((shuju[i,1]==1)&(is.na(shuju[i,13])==FALSE)){
    jishu1=1
    jishu=jishu+1
    YY[[jishu]]=numeric(0)
    TT[[jishu]]=numeric(0)
    YY[[jishu]][jishu1]=shuju[i,13]
    TT[[jishu]][jishu1]=shuju[i,5]
  }
  if(i>1){
    if((shuju[i,1]>shuju[i-1,1])&(is.na(shuju[i,13])==FALSE)){
    jishu1=jishu1+1
    YY[[jishu]][jishu1]=shuju[i,13]
    TT[[jishu]][jishu1]=shuju[i,5]
    }
 }
}

NN=numeric(0)
for(i in 1:562){
  NN[i]=length(YY[[i]])
}

TT2=numeric(sum(NN))
YY2=numeric(sum(NN))

shu1=0
for(i in 1:562){
  TT2[(shu1+1):(shu1+NN[i])]=TT[[i]]
  YY2[(shu1+1):(shu1+NN[i])]=YY[[i]]
  shu1=shu1+NN[i]
}

TT1=(TT2-min(TT2))/(max(TT2)-min(TT2))
```

```
YY1=-(YY2-mean(YY2))/sd(YY2)
n=length(NN)

quant1=quantile(TT1,probs=seq(0,1,0.25))
hr=1.06*(n*mean(NN))^(-1/5)*min(sd(TT1),(quant1[4]-quant1[2])/1.34)
quant1=quantile(YY1,probs=seq(0,1,0.25))
hd=1.06*(n*mean(NN))^(-0.3)*min(sd(YY1),(quant1[4]-quant1[2])/1.34)
t1=seq(0.01,0.99,by=0.001)
N=1000

weight="OBS"
jieguo1=monfuncreg(TT1, YY1, NN,  hr, hd, N, weight,t1)
weight="SUBJ"
jieguo2=monfuncreg(TT1, YY1, NN,  hr, hd, N, weight,t1)

matplot(jieguo1$variable,cbind(-jieguo1$estimate,-jieguo2$estimate),type="l",
lty=1,lwd=2,xlab="Scaled Age", ylab="Standardized Volume of Grey Matter")
```

Index