Package ‘dma’

February 19, 2015

Type  Package
Title  Dynamic model averaging
Version  1.2-2
Date  2014-11-5
Author  Tyler H. McCormick, Adrian Raftery, David Madigan
Maintainer  Hana Sevcikova <hanas@uw.edu>
Description  Dynamic model averaging for binary and continuous outcomes.
Imports  MASS, mnormt
License  GPL-2
LazyLoad  yes
NeedsCompilation  no
Repository  CRAN
Date/Publication  2014-11-06 10:37:09

R topics documented:

  dma-package .................................................. 1
  dma ............................................................ 3
  logistic.dma .................................................. 5

Index  8

  dma-package  Dynamic model averaging

Description

This package implements dynamic Bayesian model averaging as described for continuous outcomes in Raftery et al. (2010, Technometrics) and for binary outcomes in McCormick et al. (2011, Biometrics).
DNA Package

Details
**Package: dma**

**Type: Package**

**Version: 1.2-2**

**Date: 2014-11-5**

**License: GPL2**

**LazyLoad: yes**

---

**Author(s)**

Tyler H. McCormick, Adrian Raftery, David Madigan

Maintainer: Hana Sevcikova <hanas@uw.edu>

---

**References**


---

**Description**

Implement dynamic model averaging for continuous outcomes as described in Raftery, A.E., Karny, M., and Ettrler, P. (2010). Online Prediction Under Model Uncertainty Via Dynamic Model Averaging: Application to a Cold Rolling Mill. Technometrics 52:52-66. Along with the values described below, plot() creates a plot of the posterior model probabilities over time and model-averaged fitted values and print() returns model matrix and posterior model probabilities. There are TT time points, K models, and d total covariates.

**Usage**

dma(x, y, models.which, lambda=0.99, gamma=0.99, eps=.001/nrow(models.which), delay=1, initialperiod=200)

**Arguments**

- **x**: TTxd matrix of system inputs
- **y**: TT-vector of system outputs
- **models.which**: Kxd matrix, with 1 row per model and 1 col per variable indicating whether that variable is in the model (the state theta is of dim (model.dim+1); the extra 1 for the intercept)
lambda  parameter forgetting factor
gamma  flattening parameter for model updating
eps    regularization parameter for regularizing posterior model model probabilities away from zero
delay  When \( y_t \) is controlled, only \( y_{t-delay-1} \) and before are available. This is determined by the machine. Note that delay as defined here corresponds to \( (k-1) \) in the Ettler et al (2007, MixSim) paper. Thus we use the default delay=24, corresponding to \( k=25 \).
initialperiod  length of initial period. Performance is summarized with and without the first initialperiod samples.

Value

\texttt{yhat.bymodel} \quad \text{TTxK matrix whose tk element gives yhat for yt for model k}
\texttt{yhat.ma} \quad \text{TT vector whose t element gives the model-averaged yhat for yt}
\texttt{pmp} \quad \text{TTxK matrix whose tk element is the post prob of model k at t}
\texttt{thetahat.ma} \quad \text{TTx(nvar+1) matrix whose tk element is the model-averaged estimate of theta}_{j-1} \text{ at t}
\texttt{Vtheta.ma} \quad \text{TTx(nvar+1) matrix whose tk element is the model-averaged variance of thetahat}_{j-1} \text{ at t}
\texttt{mse.bymodel} \quad \text{MSE for each model}
\texttt{mse.ma} \quad \text{MSE of model-averaged prediction}
\texttt{mseinitialperiod.bymodel} \quad \text{MSE for each model excluding the first initialperiod samples}
\texttt{mseinitialperiod.ma} \quad \text{MSE of model averaging excluding the first initialperiod samples}
\texttt{model.forget} \quad \text{forgetting factor for the model switching matrix}

Author(s)

Adrian Raftery, Tyler H. McCormick

References


Examples

#simulate some data to test
#first, static coefficients
coeff<-c(1.8,3.4,-2.3,-2.8,3)
coefmat<-cbind(rep(coef[1],200),rep(coef[2],200),
rep(coef[3],200),rep(coef[4],200),
rep(coef[5],200),rep(coef[6],200))
#then, dynamic ones
coefmat<-cbind(coefmat, seq(1, 2.45, length.out=nrow(coefmat)),
seq(-.75, -.2.75, length.out=nrow(coefmat)),
c(rep(-1.5, nrow(coefmat)/2), rep(-.5, nrow(coefmat)/2)))
npar<-ncol(coefmat)-1
dat<-matrix(rnorm(200*(npar), 0, 1), 200, npar))
ydat<-rowsums((cbind(rep(1, nrow(dat)), dat))[,1:100]*coefmat[,1:100])
ydat<-c(ydat, rowsums((cbind(rep(1, nrow(dat)), dat)*coefmat[,c(1,100),c(6:9)]))
mmat<-matrix(c(rep(1,1,0,0,rep(1, (npar-7)), 0,0),
c(rep(0,(npar-4)), rep(1,4)), rep(1, npar)), 3, npar, byrow=TRUE)
dma.test<-dma(dat, ydat, mmat, lambda=.99, gamma=.99, initialperiod=20)
plot(dma.test)

logistic.dma

Dynamic model averaging for binary outcomes

Description

Implement dynamic model averaging for continuous outcomes as described in McCormick, T.M., Raftery, A.E., Madigan, D. and Burd, R.S. (2011) "Dynamic Logistic Regression and Dynamic Model Averaging for Binary Classification." Biometrics, 66:1162-1173. Along with the values described below, plot() creates a plot of the posterior model probabilities over time and model-averaged fitted values (with smooth curve overlay) and print() returns model matrix and posterior model probabilities. There are K candidate models, T time points, and d total covariates (including the intercept).

Usage

logistic.dma(x, y, models.which, lambda=0.99, alpha=0.99, autotune=TRUE,
initmodelprobs=NULL, initialsamp=NULL)

Arguments

x T by (d-1) matrix of observed covariates. Note that a column of 1's is added automatically for the intercept.
y T vector of binary responses
models.which K by (d-1) matrix defining models. A 1 indicates a covariate is included in a particular model, a 0 if it is excluded. Model averaging is done over all model defined in models.which.
lambda scalar forgetting factor with each model
alpha scalar forgetting factor for model averaging
autotune T/F indicates whether or not the automatic tuning procedure described in McCormick et al. should be applied. Default is true.
initmodelprobs K vector of starting probabilities for model averaging. If null (default), then use 1/K for each model.
initialsamp scalar indicating how many observations to use for generating initial values. If null (default), then use the first 10 percent of observations.
Value

- **x**: T by (d-1) matrix of covariates
- **y**: T by 1 vector of binary responses
- **models.which**: K by (d-1) matrix of candidate models
- **lambda**: scalar, tuning factor within models
- **alpha**: scalar, tuning factor for model averaging
- **autotune**: T/F, indicator of whether or not to use autotuning algorithm
- **alpha.used**: T vector of alpha values used
- **theta**: T vector of alpha values used
- **vartheta**: K by T by d array of dynamic logistic regression variances for each model
- **pmp**: K by T array of posterior model probabilities
- **yhatdma**: T vector of model-averaged predictions
- **yhatmodel**: K by T vector of fitted values for each model

Author(s)

Tyler H. McCormick, David Madigan, Adrian Raftery

References


Examples

```R
# simulate some data to test
# first, static coefficients
coef<-c(.08,.4,-.1)
coefmat<-cbind(rep(coef[1],200),rep(coef[2],200),rep(coef[3],200))
# then, dynamic ones
coefmat<-cbind(coefmat,seq(1,45,length.out=nrow(coefmat)),
    seq(-.75,-.15,length.out=nrow(coefmat)),
    c(rep(-1,nrow(coefmat)/3),rep(-.5,nrow(coefmat)/3))
npar<-ncol(coefmat)-1

# simulate data
dat<-matrix(rnorm(200*(npar),0,1),200,(npar))
ydat<-exp(rowSums(cbind(rep(1,nrow(dat)),dat))[-c(1:100),]
    coefmat[-c(1:100),
    ]/ (1+exp(rowSums(cbind(rep(1,nrow(dat)),dat)[-c(1:100),
        coefmat[-c(1:100),
        ]])))
y<-c(ydat,exp(rowSums(cbind(rep(1,nrow(dat)),dat)[-c(1:100),
            coefmat[-c(1:100),
            ]]))/ (1+exp(rowSums(cbind(rep(1,nrow(dat)),dat)[-c(1:100),
                coefmat[-c(1:100),
                ]])))
u <- runif (length(y))
y <- as.numeric (u < y)

# Consider three candidate models
```
mmat<-matrix(c(1,1,1,1,0,0,0,1,1,0,1,0,1,3,5),byrow=TRUE)

# Fit model and plot
# autotuning is turned off for this demonstration example
ldma.test<-logistic.dma(dat,y,mmat,lambda=.99, alpha=.99, autotune=FALSE)
plot(ldma.test)
Index

*Topic **dma**
  dma-package, 1
  coef.dma (dma), 3
  dlogr.init (logistic.dma), 5
  dlogr.predict (logistic.dma), 5
  dlogr.step (logistic.dma), 5
  dma, 3
  dma-package, 1
  laplace.fn (logistic.dma), 5
  logistic.dma, 5
  makf4 (dma), 3
  model.update3 (dma), 3
  plot.dma (dma), 3
  plot.logistic.dma (logistic.dma), 5
  print.dma (dma), 3
  print.logistic.dma (logistic.dma), 5
  rm.Kalman (dma), 3
  tunemat.fn (logistic.dma), 5