Package 'aplms'

November 7, 2025

Type Package

Title Additive Partial Linear Models with Symmetric Autoregressive Errors

Version 0.1.0

Maintainer Shu Wei Chou-Chen <shuwei.chou@ucr.ac.cr>

Description Set of tools for fitting the additive partial linear models with symmetric autoregressive errors of order p, or APLMS-AR(p). This setup enables the modeling of a time series response variable using linear and nonlinear structures of a set of explanatory variables, with nonparametric components approximated by natural cubic splines or P-splines. It also accounts for autoregressive error terms with distributions that have lighter or heavier tails than the normal distribution. The package includes various error distributions, such as normal, generalized normal, Student's t, generalized Student's t, power-exponential, and Cauchy distributions. Chou-Chen, S.W., Oliveira, R.A., Raicher, I., Gilberto A. Paula (2024) <doi:10.1007/s00362-024-01590-w>.

License GPL-2

URL https://github.com/shuwei325/aplms

BugReports https://github.com/shuwei325/aplms/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5.0), mgcv

Imports MASS, Matrix, rlist, stats, methods, psych, rmutil

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Shu Wei Chou-Chen [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-5495-2486>), Rodrigo A. Oliveira [aut] (ORCID: <https://orcid.org/0000-0003-3328-6669>), Gilberto A. Paula [aut] (ORCID: <https://orcid.org/0000-0002-9906-9942>), Sebastian Sanchez Sandi [ctb]

Repository CRAN

Date/Publication 2025-11-07 14:00:25 UTC

Contents

`AIC.aplms` *Akaike information criterion*

Description

Print the AIC of the fitted APLMS model.

Usage

```
## S3 method for class 'aplms'
AIC(object, ...)
```

Arguments

`object`	APLMS object.
`...`	other arguments

Value

a numeric value of the corresponding AIC.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
AIC(model)
```

---

aplms *Fitting Additive partial linear models with symmetric errors*

Description

aplms fits additive partial linear models with autoregressive symmetric errors. This method is suitable for data sets where the response variable is continuous and symmetric, with either heavy or light tails, and measured over time. The model includes a parametric component for a set of covariates, while another set of covariates can be specified as semi-parametric functions, typically time-related. In this setup, natural cubic splines or cubic P-splines are used to approximate the nonparametric components.

Usage

```
aplms(
  formula,
  npc,
  basis,
  Knot,
  data,
  family = Normal(),
  p = 1,
 control = list(tol = 0.001, algorithm1 = c("P-GAM"), algorithm2 = c("BFGS"), Maxiter1 =
    20, Maxiter2 = 25),
  init,
  lam,
  verbose = FALSE
)

## S3 method for class 'aplms'
print(x, ...)
```

Arguments

formula	A symbolic description of the parametric component of the model to be fitted. The details of model specification are given under 'Details'.
npc	A vector of names of non parametric component.
basis	A vector of names of the basis to be used for each non parametric covariate.
Knot	A vector of the number of knots in each non-linear component of the model.
data	A data frame containing the variables in the model
family	Symmetric error distribution. The implemented distribution are: `Normal()`, `LogisI()`, `LogisII()`, `Student(df)`, `Powerexp(k)`, `Gstudent(parm=c(s,r))`.
p	autoregressive order of the error
control	optimization rutine.
init	A list of initial values for the symmetric error scale, phi, and autoregressive coefficients, rhos.
lam	smoothing parameter vector.
verbose	Logical; if TRUE, prints estimation progress messages. Default is FALSE.
x	APLMS object.
...	other arguments

Value

Returns an object of class "aplms", a list with following components.

formula the `formula` object used.

family the `family` object used.

npc the `npc` object used.

Knot the `Knot` object used.

lam the `lam` object used.

rdf Degrees of freedom: `n - q - p - 1`.

VAR_F Estimate the asymptotic covariance matrix for the gamma parameters.

basis The `basis` to be used for each non parametric covariate.

WALD_f The summary table of the Wald statistics.

summary_table_phirho The summary table of the rho and phi parameters.

N_i Basis functions.

f Estimated gamma parameters.

Dv Dv values for the symmetric error.

Dm Dm values for the symmetric error.

Dc Dc values for the symmetric error.

Dd Dd values for the symmetric error.

delta delta_i for the symmetric error.

LL_obs Observed information matrix of the fitted model.

loglike The estimated loglikelihood function of the fitted model.

total_df The total effective degree of freedom of the model.

parametric_df The degree of freedom of the parametric components.

npc_df The effective degree of freedom of the non parametric components.

AIC Akaike information criterion of the estimated model.

BIC Bayesian information criterion of the estimated model.

AICC Corrected Akaike information criterion of the estimated model.

GCV The generalized cross-validation (GCV).

yhat The fitted response values of the model.

muhat The fitted mean values of the model.

residuals_y The response residuals

residuals_mu Raw (Ordinary) residuals: $y_t - (\mathbf{x}_i^\top \beta + f_1(t_{i1}) + \ldots + f_k(t_{ik}))$

data the `data` object used.

this.call the function call used.

References

Chou-Chen, S.W., Oliveira, R.A., Raicher, I., Gilberto A. Paula (2024) *Additive partial linear models with autoregressive symmetric errors and its application to the hospitalizations for respiratory diseases* Stat Papers 65, 5145–5166. doi:10.1007/s0036202401590w

Oliveira, R.A., Paula, G.A. (2021) *Additive partial linear models with autoregressive symmetric errors and its application to the hospitalizations for respiratory diseases* Comput Stat 36, 2435–2466. doi:10.1007/s00180021011062

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
plot(model)
summary(model)
print(model)
```

aplms.diag.plot *Diagnostic Plots for additive partial linear models with symmetric errors*

Description

Diagnostic Plots for additive partial linear models with symmetric errors

Usage

```
aplms.diag.plot(model, which, labels = NULL, iden = FALSE, ...)
```

Arguments

model	an object with the result of fitting additive partial linear models with symmetric errors.
which	an optional numeric value with the number of only plot that must be returned.
labels	a optional string vector specifying a labels plots.
iden	a logical value used to identify observations. If TRUE the observations are identified by user in the graphic window.
...	graphics parameters to be passed to the plotting routines.

Value

Return an interactive menu with eleven options to make plots. This menu contains the follows graphics: 1: Response residuals against fited values. 2: Response residuals against time index. 3: Histogram of Response residuals. 4: Autocorrelation function of response residuals. 5: Partial autocorrelation function of response residuals. 6: Conditional quantile residuals against fited values. 7: Conditional quantile residuals against time index. 8: Histogram of conditional quantile residuals. 9: Autocorrelation function of conditional quantile residual. 10: Partial autocorrelation function of conditional quantile residuals. 11: QQ-plot of conditional quantile residuals.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
aplms.diag.plot(model, which = 1)
```

BIC.aplms *Bayesian information criterion*

Description

Print the BIC of the fitted APLMS model.

Usage

```
## S3 method for class 'aplms'
BIC(object, ...)
```

Arguments

object	APLMS object.
...	other arguments

Value

a numeric value of the corresponding BIC.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
BIC(model)
```

coef.aplms *Extract the coefficients of the fitted APLMS model*

Description

Extract the coefficients of the fitted APLMS model.

Usage

```
## S3 method for class 'aplms'
coef(object, ...)
```

Arguments

object	APLMS object.
...	other arguments

Value

A list of vectors of the corresponding estimated parameters.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
coef(model)
```

---

family.elliptical *Family Objects for Elliptical Models*

Description

The family object provide an specify details of the model APLMS. The distribution functions are necessary to specify the random component of the regression models with elliptical errors. The code is derived from the archived package **gwer** (Araujo, Y.A., Cysneiros, F.J.A., and Cysneiros, A.H.M.A., 2022), originally available on CRAN.

Usage

```
## S3 method for class 'elliptical'
family(object, ...)

Normal()

Cauchy()

LogisI()

LogisII()

Student(df = stop("no df argument"))
```

```
Powerexp(k = stop("no k argument"))

Glogis(parma = stop("no alpha=alpha(m) or m argument"))

Gstudent(parm = stop("no s or r argument"))

Cnormal(parmt = stop("no epsi or sigma argument"))

GNormal(nu = stop("no nu argument"))
```

Arguments

object	an object with the result of the fitted elliptical regression model.
...	arguments to be used to form the default control argument if it is not supplied directly.
df	degrees of freedom.
k	shape parameter.
parma	parameter vector (alpha, m).
parm	parameter vector (s, r) for this distribuition.
parmt	parameters vector (epsi, sigma).
nu	degrees of freedom.

Value

An object of class "family" specifying a list with the follows elements:

family	character: the family name.
g0, g1, g2, g3, g4, g5	derived fuctions associated with the distribution family defined.
df	degree of freedom for t-Student distribution.
s, r	shape parameters for generalized t-Student distribution.
alpha	shape parameter for contaminated normal and generalized logistic distributions.
mp	shape parameter for generalized logistic distribution.
epsi, sigmap	dispersion parameters for contaminated normal distribution.
k	shape parameter for power exponential distribution.

a family elliptical object using a Normal distribution.

a family elliptical object using a Couchy distribution.

a family elliptical object using a LogisI distribution.

a family elliptical object using a LogisII distribution.

a family elliptical object using a Student distribution with a specific degrees of freedom.

a family elliptical object using an exponential distribution.

a famility elliptical object using a Glogis distribution.

a famility elliptical object using a Gstudent distribution.

a famility elliptical object using a Cnormal distribution

a famility elliptical object using a Gnormal distribution

References

Fang, K. T., Kotz, S. and NG, K. W. (1990, ISBN:9781315897943). Symmetric Multivariate and Related Distributions. London: Chapman and Hall.

Examples

```
Normal()
Powerexp(k=0.1)
```

`hospitalization` *Respiratory diseases hospitalization Dataset*

Description

This dataset consists of respiratory diseases hospitalization in Sorocaba, São Paulo, Brazil. The details of the statistical modeling using the APLMS-AR(p) approach can be found in Chou-Chen, et al. (2024). doi:10.1007/s0036202401590w. The hospitalization data of respiratory diseases in Sorocaba city, São Paulo, Brazil are obtained from the Hospital Information System of Brazil's Unified National Health System (SIH-SUS), and the climatic and pollution data are provided by the QUALAR system.

Usage

```
data(hospitalization)
```

Format

The "data" slot is a data frame with 932 weekly data on the following 29 variables.

date, year, epi.week, tdate Date, year, epidemiologic weeks, and time index.

y Respiratory hospitalization count.

MP10_max, MP10_min, MP10_avg Maximum, minimum and average of $MP10$.

NO_max,NO_min,NO_avg Maximum, minimum and average of NO.

NO2_max, NO2_min, NO2_avg Maximum, minimum and average of NO_2.

NOx_max, NOx_min, NOx_avg Maximum, minimum and average of NO_x.

O3_max, O3_min, O3_avg Maximum, minimum and average of O_3.

TEMP_max, TEMP_min, TEMP_avg Maximum, minimum and average of temperature.

RH_max, RH_min, RH_avg Maximum, minimum and average of relative humidity.

ampl_max, ampl_min, ampl_avg Maximum, minimum and average of daily temperature amplitude.

References

Chou-Chen, S.W., Oliveira, R.A., Raicher, I. et al. (2024) Additive partial linear models with autoregressive symmetric errors and its application to the hospitalizations for respiratory diseases. Stat Papers 65, 5145–5166. doi:10.1007/s0036202401590w

Examples

```
data(hospitalization)
head(hospitalization)
```

`influence.aplms` *local influence analysis of the object 'aplms()'*

Description

Takes a fitted 'aplms' object and outputs some diagnostic information about the fitting procedure and results. Returns the conformal normal curvature of the fitted 'aplms' model object. The 'case-weight', 'dispersion', 'response', 'explanatory', and 'corAR' perturbations are available.

Usage

```
## S3 method for class 'aplms'
influence(
  model,
 perturbation = c("case-weight", "dispersion", "response", "explanatory", "corAR"),
  part = TRUE,
  ...
)
```

Arguments

`model`	an object with the result of fitting additive partial linear models with symmetric errors.
`perturbation`	A string vector specifying a perturbation scheme: 'case-weight', 'dispersion', 'response', 'explanatory', and 'corAR'.
`part`	A logical value to indicate whether the influential analysis is performed for γ, ϕ and ρ.
`...`	other arguments.

Value

A list object containing the conformal normal curvature of the specified perturbations.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
influence(model, perturbation = c("case-weight"))
```

`influenceplot.aplms` *Local influence plots of the object 'aplms()'*

Description

Takes a fitted 'aplms' object and outputs diagnostics of the sensitivity analysis by assessing the effects of perturbations in the model and/or data, on the parameter estimates. The 'case-weight', 'dispersion', 'response', 'explanatory', and 'corAR' perturbations are available.

Usage

```
influenceplot.aplms(
  model,
 perturbation = c("case-weight", "dispersion", "response", "explanatory", "corAR"),
  part = TRUE,
  C = 4,
  labels = NULL
)
```

Arguments

`model`	an object with the result of fitting additive partial linear models with symmetric errors.
`perturbation`	A string vector specifying a perturbation scheme: 'case-weight', 'dispersion', 'response', 'explanatory', and 'corAR'.
`part`	A logical value to indicate whether the influential analysis is performed for γ, ϕ and ρ.
`C`	The cutoff criterion such that $C_i > \bar{C}_i + C * sd(C_i)$ to detect influential observations.
`labels`	label to especify each data point.

Value

The conformal normal curvature of the specified perturbations is plotted.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
influenceplot.aplms(model, perturbation = c("case-weight"))
```

---

plot.aplms *Default APLMS plotting*

Description

Compute and plot the estimated mean and confidence intervals of the non-parametric component of a 'APLMS' object fited by 'aplms()'.

Usage

```
## S3 method for class 'aplms'
plot(x, len = 100, plot = TRUE, level = 0.95, ...)
```

Arguments

x	an object with the result of fitting additive partial linear models with symmetric errors.
len	The desired length of the sequence of covariates to compute the non parametric component functions.
plot	a logical value to return plots. Default value is TRUE.
level	Confidence level.
...	other arguments.

Value

Return a list of all non parametric component functions with their confidence intervals. If plot=TRUE, the estimated nonparametric component functions are plotted.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                   algorithm1 = c("P-GAM"),
                                   algorithm2 = c("BFGS"),
                                   Maxiter1 = 20,
                                   Maxiter2 = 25),
                    lam=c(10))
plot(model)
```

---

residuals.aplms *Extract Residuals for APLMS fits*

Description

Extract Residuals for APLMS fits

Usage

```
## S3 method for class 'aplms'
residuals(object, ...)
```

Arguments

`object`	an object with the result of fitting additive partial linear models with symmetric errors.
`...`	other arguments.

Value

Returns a dataframe with the following columns

res the residual,

res_pearson the Pearson residual, and

res_quant the normal quantile of the standarized resiudals.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
```

```
                                algorithm1 = c("P-GAM"),
                                algorithm2 = c("BFGS"),
                                Maxiter1 = 20,
                                Maxiter2 = 25),
                    lam=c(10))
residuals(model)
```

---

summary.aplms *Print method for "aplms" class*

Description

Print method for "aplms" class

Usage

```
## S3 method for class 'aplms'
summary(object, ...)
```

Arguments

object	an object with the result of fitting additive partial linear models with symmetric errors.
...	Other arguments passed to or from other methods.

Value

Prints the main results of the fitted APLMS model.

Examples

```
data(temperature)
temperature.df = data.frame(temperature,time=1:length(temperature))
model<-aplms::aplms(temperature ~ 1,
                    npc=c("time"), basis=c("cr"),Knot=c(60),
                    data=temperature.df,family=Powerexp(k=0.3),p=1,
                    control = list(tol = 0.001,
                                algorithm1 = c("P-GAM"),
                                algorithm2 = c("BFGS"),
                                Maxiter1 = 20,
                                Maxiter2 = 25),
                    lam=c(10))
summary(model)
```

`temperature` *Global Annual Mean Surface Air Temperature Change*

Description

Land-ocean temperature index from 1880–2021 (with base period 1951-1980).

Usage

```
data(temperature)
```

Format

Time series data.

Source

NASA/GISS/GISTEMP

References

Land-Ocean Temperature Index.

Examples

```
data(temperature)
```

Index