Package ‘alpaca’

January 12, 2020

Type Package

Title Fit GLM's with High-Dimensional k-Way Fixed Effects

Version 0.3.2

Description Provides a routine to concentrate out factors with many levels during the
optimization of the log-likelihood function of the corresponding generalized linear model (glm).
The package is based on the algorithm proposed by Stammann (2018) <arXiv:1707.01815> and is
restricted to glm's that are based on maximum likelihood estimation and non-linear. It also offers
an efficient algorithm to recover estimates of the fixed effects in a post-estimation routine and
includes robust and multi-way clustered standard errors. Further the package provides analytical
bias corrections for binary choice models (logit and probit) derived by Fernandez-Val
and Weidner (2016) <doi:10.1016/j.jeconom.2015.12.014> and Hinz, Stammann, and Wan-
ner (2019).

License GPL-3

Depends R (>= 3.1.0)

Imports data.table, Formula, MASS, Rcpp, stats, utils

LinkingTo Rcpp, RcppArmadillo

URL https://github.com/amrei-stammann/alpaca

BugReports https://github.com/amrei-stammann/alpaca/issues

RoxygenNote 7.0.2

Suggests bife, car, knitr, lfe

VignetteBuilder knitr

NeedsCompilation yes

Author Amrei Stammann [aut, cre],
Daniel Czarnowske [aut] (<https://orcid.org/0000-0002-0030-929X>)

Maintainer Amrei Stammann <amrei.stammann@hhu.de>

Repository CRAN

Date/Publication 2020-01-12 16:30:03 UTC
Description

Concentrates out factors with many levels during the optimization of the log-likelihood function of the corresponding generalized linear model (glm). The package is restricted to glm’s that are based on maximum likelihood estimation. This excludes all quasi-variants of glm. The package also offers an efficient algorithm to recover estimates of the fixed effects in a post-estimation routine and includes robust and multi-way clustered standard errors. Further the package provides analytical bias corrections for binary choice models (logit and probit) derived by Fernandez-Val and Weidner (2016) and Hinz, Stammann, and Wanner (2019).

Note: Linear models are also beyond the scope of this package since there is already a comprehensive procedure available **felm**.
**biasCorr**  
*Asymptotic bias correction after fitting binary choice models with a two-/three-way error component*

**Description**

**biasCorr** is a post-estimation routine that can be used to substantially reduce the incidental parameter bias problem (Neyman and Scott (1948)) present in non-linear fixed effects models (see Fernandez-Val and Weidner (2018) for an overview). The command applies the analytical bias correction derived by Fernandez-Val and Weidner (2016) and Hinz, Stammann, and Wanner (2019) to obtain bias-corrected estimates of the structural parameters and is currently restricted to logit and probit models.

**Usage**

```r
biasCorr(object = NULL, L = 0L, panel.structure = c("classic", "network"))
```

**Arguments**

- `object`  
an object of class "feglm"; currently restricted to **binomial** with "logit" or "probit" link function.
- `L`  
unsigned integer indicating a bandwidth for the estimation of spectral densities proposed by Hahn and Kuersteiner (2011). Default is zero, which should be used if all regressors are assumed to be strictly exogenous. In the presence of weakly exogenous or predetermined regressors, Fernandez-Val and Weidner (2016, 2018) suggest to choose a bandwidth not higher than four. Note that the order of factors to be concentrated out is important for bandwidths larger than zero (see vignette for details).
- `panel.structure`  
a string equal to "classic" or "network" which determines the structure of the panel used. "classic" denotes panel structures where for example the same cross-sectional units are observed several times (this includes pseudo panels). "network" denotes panel structures where for example bilateral trade flows are observed for several time periods. Default is "classic".

**Value**

The function **biasCorr** returns a named list of classes "biasCorr" and "feglm".

**References**


See Also

feglm

Examples

# Generate an artificial data set for logit models
library(alpaca)
data <- simGLM(1000L, 20L, 1805L, model = "logit")

# Fit 'feglm()
mod <- feglm(y ~ x1 + x2 + x3 | i + t, data)

# Apply analytical bias correction
mod.bc <- biasCorr(mod)
summary(mod.bc)

coef.APEs

Extract estimates of average partial effects

Description

coef.APEs is a generic function which extracts estimates of the average partial effects from objects returned by getAPEs.

Usage

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

Arguments

object an object of class "APEs".
...
other arguments.
Value

The function `coef.APEs` returns a named vector of estimates of the average partial effects.

See Also

`getAPEs`

---

### coef.feglm

**Extract estimates of structural parameters**

**Description**

`coef.feglm` is a generic function which extracts estimates of the structural parameters from objects returned by `feglm`.

**Usage**

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

**Arguments**

- `object` an object of class "feglm".
- `...` other arguments.

**Value**

The function `coef.feglm` returns a named vector of estimates of the structural parameters.

See Also

`feglm`

---

### coef.summary.APEs

**Extract coefficient matrix of average partial effects**

**Description**

`coef.summary.APEs` is a generic function which extracts a coefficient matrix of average partial effects from objects returned by `getAPEs`.

**Usage**

```r
## S3 method for class 'summary.APEs'
coef(object, ...)
```
Arguments

object an object of class "summary.APEs".
...
oner arguments.

Value

The function `coef.summary.APEs` returns a named matrix of estimates related to the average partial effects.

See Also

g ape s

c oef.summary.feglm

---

c oef.summary.feg lm  Extract coefficient matrix of structural parameters

Description

`coef.summary.feglm` is a generic function which extracts a coefficient matrix of structural parameters from objects returned by `feglm`.

Usage

```r
## S3 method for class 'summary.feglm'
coef(object, ...)
```

Arguments

object an object of class "summary.feglm".
...
oner arguments.

Value

The function `coef.summary.feglm` returns a named matrix of estimates related to the structural parameters.

See Also

`feglm`
**Description**

`feglm` can be used to fit generalized linear models with many high-dimensional fixed effects. The estimation procedure is based on unconditional maximum likelihood and can be interpreted as a “pseudo demeaning” approach that combines the work of Gaure (2013) and Stammann et. al. (2016). For technical details see Stammann (2018). The routine is well suited for large data sets that would be otherwise infeasible to use due to memory limitations.

**Remark:** The term fixed effect is used in econometrician’s sense of having intercepts for each level in each category.

**Usage**

```r
feglm(
  formula = NULL,
  data = NULL,
  family = binomial(),
  beta.start = NULL,
  eta.start = NULL,
  control = NULL
)
```

**Arguments**

- `formula`: an object of class "formula": a symbolic description of the model to be fitted. `formula` must be of type \( y \sim x | k \), where the second part of the formula refers to factors to be concentrated out. It is also possible to pass additional variables to `feglm` (e.g. to cluster standard errors). This can be done by specifying the third part of the formula: \( y \sim x | k | add \).
- `data`: an object of class "data.frame" containing the variables in the model.
- `family`: a description of the error distribution and link function to be used in the model. Similar to `glm.fit` this has to be the result of a call to a family function. Default is `binomial()`. See `family` for details of family functions.
- `beta.start`: an optional vector of starting values for the structural parameters in the linear predictor. Default is \( \beta = 0 \).
- `eta.start`: an optional vector of starting values for the linear predictor.
- `control`: a named list of parameters for controlling the fitting process. See `feglmControl` for details.

**Details**

If `feglm` does not converge this is usually a sign of linear dependence between one or more regressors and a fixed effects category. In this case, you should carefully inspect your model specification.
The function `feglm` returns a named list of class "feglm".

References

Gaure, S. (2013). "OLS with Multiple High Dimensional Category Variables". Computational Statistics and Data Analysis, 66.


Examples

```r
# Generate an artificial data set for logit models
library(alpaca)
data <- simGLM(1000L, 20L, 1805L, model = "logit")

# Fit 'feglm()
mod <- feglm(y ~ x1 + x2 + x3 | i + t, data)
summary(mod)
```

Description

`feglm.nb` can be used to fit negative binomial generalized linear models with many high-dimensional fixed effects (see `feglm`).

Usage

```r
feglm.nb(
  formula = NULL,
  data = NULL,
  beta.start = NULL,
  eta.start = NULL,
  init.theta = NULL,
  link = c("log", "identity", "sqrt"),
  control = NULL
)
```
Arguments
  formula, data, beta.start, eta.start, control
  see feglm.
init.theta an optional initial value for the theta parameter (see glm.nb).
link the link function. Must be one of "log", "sqrt", or "identity".

Details
  If feglm.nb does not converge this is usually a sign of linear dependence between one or more regressors and a fixed effects category. In this case, you should carefully inspect your model specification.

Value
  The function feglm.nb returns a named list of class "feglm".

References
  Gaure, S. (2013). "OLS with Multiple High Dimensional Category Variables". Computational Statistics and Data Analysis. 66.

See Also
  glm.nb, feglm

feglmControl Set feglm Control Parameters

Description
  Set and change parameters used for fitting feglm.
  Note: feglm.control is deprecated and will be removed soon.

Usage
  feglmControl(
    dev.tol = 1e-08,
    center.tol = 1e-05,
    rho.tol = 1e-04,
    conv.tol = 1e-06,
    iter.max = 100L,
    limit = 10L,
trace = FALSE,
drop.pc = TRUE,
pseudo.tol = NULL,
step.tol = NULL
)

feglm.control(…)

Arguments

dev.tol    tolerance level for the first stopping condition of the maximization routine. The stopping condition is based on the relative change of the deviance in iteration \( r \) and can be expressed as follows: \( (dev_{r-1} - dev_r)/(0.1 + dev_r) < tol \). Default is \( 1.0e^{-08} \).

center.tol tolerance level for the stopping condition of the centering algorithm. The stopping condition is based on the relative change of euclidean norm in iteration \( i \) and can be expressed as follows: \( ||v_i - v_{i-1}||_2 < tol ||v_{i-1}|| \). Default is \( 1.0e^{-05} \).

rho.tol    tolerance level for the stephalving in the maximization routine. Step halving only takes place if the deviance in iteration \( r \) is larger than the one of the previous iteration. If this is the case, \( ||\beta_r - \beta_{r-1}||_2 \) is halfed until the deviance is less or numerically equal compared to the deviance of the previous iteration. Step halving fails if the following condition holds: \( \rho < tol \), where \( \rho \) is the step correction factor. If step halving fails the maximization routine is canceled. Default is \( 1.0e^{-04} \).

conv.tol   tolerance level that accounts for rounding errors inside the stephalving routine when comparing the deviance with the one of the previous iteration. Default is \( 1.0e^{-06} \).

iter.max   unsigned integer indicating the maximum number of iterations in the maximization routine. Default is \( 100L \).

limit      unsigned integer indicating the maximum number of iterations of \( \text{theta.ml} \). Default is \( 10L \).

trace      logical indicating if output should be produced in each iteration. Default is FALSE.

drop.pc    logical indicating to drop observations that are perfectly classified (perfectly seperated) and hence do not contribute to the log-likelihood. This option is useful to reduce the computational costs of the maximization problem, since it reduces the number of observations and does not affect the estimates. Default is TRUE.

pseudo.tol deprecated; use center.tol instead.

step.tol   deprecated; termination conditions is now similar to \( \text{glm} \).

…         arguments passed to the deprecated function \( \text{feglm.control} \).

Value

The function \( \text{feglmControl} \) returns a named list of control parameters.
fitted.feglm

See Also
feglm

tfitted.feglm Extract feglm fitted values

Description
fitted.feglm is a generic function which extracts fitted values from an object returned by feglm.

Usage

## S3 method for class 'feglm'
fitted(object, ...)

Arguments

object an object of class "feglm".
...
other arguments.

Value

The function fitted.feglm returns a vector of fitted values.

See Also
feglm

ggetAPeEs Compute average partial effects after fitting binary choice models with a two-/three-way error component

Description
ggetAPeEs is a post-estimation routine that can be used to estimate average partial effects with respect to all covariates in the model and the corresponding covariance matrix. The estimation of the covariance is based on a linear approximation (delta method). Note that the command automatically determines which of the regressors are continuous or binary.

Remark: The routine currently does not allow to compute average partial effects based on functional forms like interactions and polynomials.
Usage

getAPEs(
  object = NULL,
  n.pop = NULL,
  panel.structure = c("classic", "network"),
  sampling.fe = c("independence", "unrestricted"),
  weak.exo = FALSE
)

Arguments

object an object of class "biasCorr" or "feglm"; currently restricted to binomial
with "logit" or "probit" link function.
n.pop unsigned integer indicating a finite population correction for the estimation of
the covariance matrix of the average partial effects proposed by Cruz-Gonzalez,
Fernandez-Val, and Weidner (2017). The correction factor is computed as fol-
lows: \( \frac{n^* - n}{n^* - 1} \), where \( n^* \) and \( n \) are the size of the entire population
and the full sample size. Default is NULL, which refers to a factor of one and is
equal to an infinitely large population.
panel.structure a string equal to "classic" or "network" which determines the structure of the
panel used. "classic" denotes panel structures where for example the same
cross-sectional units are observed several times (this includes pseudo panels).
"network" denotes panel structures where for example bilateral trade flows are
observed for several time periods. Default is "classic".
sampling.fe a string equal to "independence" or "unrestricted" which imposes sampling
assumptions about the unobserved effects. "independence" imposes that all
unobserved effects are mutually independent sequences. "unrestricted" does
not impose any sampling assumptions. Note that this option only affects the
estimation of the covariance. Default is "independence".
weak.exo logical indicating if some of the regressors are assumed to be weakly exoge-
nous (e.g. predetermined). If object is of class "biasCorr", the option will
be automatically set to TRUE if the choosen bandwidth parameter is larger than
zero. Note that this option only affects the estimation of the covariance matrix.
Default is FALSE, which assumes that all regressors are strictly exogenous.

Value

The function getAPEs returns a named list of class "APEs".

References

logit models with two-way fixed effects". The Stata Journal, 17(3), 517-545.

Czarnowske, D. and Stammann, A. (2019). "Binary Choice Models with High-Dimensional Indi-
getFEs

Efficiently recover estimates of the fixed effects after fitting feglm

Description

Recover estimates of the fixed effects by alternating between the normal equations of the fixed effects as shown by Stammann (2018).

Remark: The system might not have a unique solution since we do not take collinearity into account. If the solution is not unique, an estimable function has to be applied to our solution to get meaningful estimates of the fixed effects. See Gaure (n. d.) for an extensive treatment of this issue.
Usage

getFEs(object = NULL, alpha.tol = 1e-08)

Arguments

object an object of class "feglm".
alpha.tol tolerance level for the stopping condition. The algorithm is stopped in iteration
   if ||\alpha_i - \alpha_{i-1}||_2 < tol ||\alpha_{i-1}||_2. Default is 1.0e-08.

Value

The function getFEs returns a named list containing named vectors of estimated fixed effects.

References

Stammann, A. (2018). "Fast and Feasible Estimation of Generalized Linear Models with High-
   Dimensional k-way Fixed Effects". ArXiv e-prints.

See Also

feglm

predict.feglm

Description

predict.feglm is a generic function which obtains predictions from an object returned by feglm.

Usage

## S3 method for class 'feglm'
predict(object, type = c("link", "response"), ...)

Arguments

object an object of class "feglm".
type the type of prediction required. "link" is on the scale of the linear predictor
   whereas "response" is on the scale of the response variable. Default is "link".
   ...
other arguments.

Value

The function predict.feglm returns a vector of predictions.
print.APEs

See Also
feglm

print.APEs is a generic function which displays some minimal information from objects returned by getAPEs.

Usage
## S3 method for class 'APEs'
print(x, digits = max(3L,getOption("digits") - 3L), ...)  

Arguments  
x an object of class "APEs".  
digits unsigned integer indicating the number of decimal places. Default is max(3L,getOption("digits") - 3L).  
... other arguments.  

See Also  
getAPEs

print.feglm

Description
print.feglm is a generic function which displays some minimal information from objects returned by feglm.

Usage
## S3 method for class 'feglm'
print(x, digits = max(3L,getOption("digits") - 3L), ...)  

Arguments  
x an object of class "feglm".  
digits unsigned integer indicating the number of decimal places. Default is max(3L,getOption("digits") - 3L).  
... other arguments.
See Also

feglm

---

Description

`print.summary.APEs` is a generic function which displays summary statistics from objects returned by `summary.APEs`.

Usage

```r
## S3 method for class 'summary.APEs'
print(x, digits = max(3L,getOption("digits") - 3L), ...)
```

Arguments

- `x`: an object of class "summary.APEs".
- `digits`: unsigned integer indicating the number of decimal places. Default is `max(3L,getOption("digits") - 3L)`. 
- `...`: other arguments.

See Also

getaPEs

---

Description

`print.summary.feglm` is a generic function which displays summary statistics from objects returned by `summary.feglm`.

Usage

```r
## S3 method for class 'summary.feglm'
print(x, digits = max(3L,getOption("digits") - 3L), ...)
```

Arguments

- `x`: an object of class "summary.feglm".
- `digits`: unsigned integer indicating the number of decimal places. Default is `max(3L,getOption("digits") - 3L)`. 
- `...`: other arguments.
**Description**

Constructs an artificial data set with \( n \) cross-sectional units observed for \( t \) time periods for logit, poisson, or gamma models. The “true” linear predictor (\( \eta \)) is generated as follows:

\[
\eta_{it} = \mathbf{x}_{it}' \beta + \alpha_i + \gamma_t,
\]

where \( \mathbf{X} \) consists of three independent standard normally distributed regressors. Both parameter referring to the unobserved heterogeneity (\( \alpha_i \) and \( \gamma_t \)) are generated as iid. standard normal and the structural parameters are set to \( \beta = [1, -1, 1]' \).

**Note:** The poisson and gamma model are based on the logarithmic link function.

**Usage**

\[
\text{simGLM}(n = \text{NULL}, t = \text{NULL}, \text{seed} = \text{NULL}, \text{model} = \text{c("logit", "poisson", "gamma")})
\]

**Arguments**

- **n** a strictly positive integer equal to the number of cross-sectional units.
- **t** a strictly positive integer equal to the number of time periods.
- **seed** a seed to ensure reproducibility.
- **model** a string equal to "logit", "poisson", or "gamma". Default is "logit".

**Value**

The function `simGLM` returns a data.frame with 6 variables.

**See Also**

`feglm`
### summary.APEs

**Summary:**

Summary statistics for objects of class "APEs".

#### Usage

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

#### Arguments

- `object`: an object of class "APEs".
- `...`: other arguments.

#### Value

Returns an object of class "summary.APEs" which is a list of summary statistics of `object`.

#### See Also

- `getAPEs`

### summary.feglm

**Summary:**

Summary statistics for objects of class "feglm".

#### Usage

```r
## S3 method for class 'feglm'
summary(
  object,
  type = c("hessian", "outer.product", "sandwich", "clustered"),
  cluster = NULL,
  cluster.vars = NULL,
  ...  
)
```

#### Description

Summarizing models of class APEs
vcov.feglm

Arguments

- **object**: an object of class "feglm".
- **type**: the type of covariance estimate required. "hessian" refers to the inverse of the negative expected Hessian after convergence and is the default option. "outer.product" is the outer-product-of-the-gradient estimator, "sandwich" is the sandwich estimator (sometimes also referred as robust estimator), and "clustered" computes a clustered covariance matrix given some cluster variables.
- **cluster**: a symbolic description indicating the clustering of observations.
- **cluster.vars**: deprecated; use **cluster** instead.
- **...**: other arguments.

Details

Multi-way clustering is done using the algorithm of Cameron, Gelbach, and Miller (2011). An example is provided in the vignette "Replicating an Empirical Example of International Trade".

Value

Returns an object of class "summary.feglm" which is a list of summary statistics of **object**.

References


See Also

feglm

description

vcov.feglm computes an estimate of the covariance matrix of the estimator of the structural parameters from objects returned by feglm. The estimate is obtained using the Hessian, the scores, or a combination of boths after convergence.

Usage

```r
## S3 method for class 'feglm'
vcov(
  object,
  type = c("hessian", "outer.product", "sandwich", "clustered"),
  cluster = NULL,
  cluster.vars = NULL,
  ...
)
```
Arguments

- **object**: an object of class "feglm".
- **type**: the type of covariance estimate required. "hessian" refers to the inverse of the negative expected Hessian after convergence and is the default option. "outer.product" is the outer-product-of-the-gradient estimator, "sandwich" is the sandwich estimator (sometimes also referred as robust estimator), and "clustered" computes a clustered covariance matrix given some cluster variables.
- **cluster**: a symbolic description indicating the clustering of observations.
- **cluster.vars**: deprecated; use cluster instead.
- **...**: other arguments.

Details

Multi-way clustering is done using the algorithm of Cameron, Gelbach, and Miller (2011). An example is provided in the vignette "Replicating an Empirical Example of International Trade".

Value

The function `vcov.feglm` returns a named matrix of covariance estimates.

References


See Also

`feglm`
Index

alpaca-package, 2
biasCorr, 3, 3, 13
binomial, 3, 12

ccoef.APEs, 4, 4, 5
ccoef.feglm, 5, 5
ccoef.summary.APEs, 5, 5, 6
ccoef.summary.feglm, 6, 6

cfamily, 7
cfeglm, 4–7, 7, 8, 9, 11, 13–17, 19, 20
cfeglm.control, 9, 10
cfeglm.control (feglmControl), 9
cfeglm.nb, 8
cfeglmControl, 7, 9, 10
cfelm, 2

cfitted.feglm, 11, 11

cgetAPEs, 4–6, 11, 11, 12, 15, 16, 18
cgetFEs, 13, 14
cglm, 10
cglm.fit, 7
cglm.nb, 9

cpredict.feglm, 14, 14
cprint.APEs, 15, 15
cprint.feglm, 15, 15
cprint.summary.APEs, 16, 16
cprint.summary.feglm, 16, 16

csimGLM, 17, 17
csummary.APEs, 16, 18
csummary.feglm, 16, 18

ctheta.ml, 10

cvcov.feglm, 19, 19, 20