Package 'alpaca'

May 15, 2019

Type Package

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

Version 0.3

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 an analytical bias-correction for binary choice models (logit and probit) derived by Fernandez-Val and Weidner (2016) <doi:10.1016/j.jeconom.2015.12.014>.

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 6.1.1

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 2019-05-14 22:10:03 UTC

R topics documented:

`alpaca` *alpaca: A package for fitting glm's with high-dimensional k-way fixed effects*

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 an analytical bias-correction for binary choice models (logit and probit) derived by Fernandez-Val and Weidner (2016).

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 two-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 Weinder (2016) to obtain bias-corrected estimates of the structural parameters and is currently restricted to logit and probit models.

Usage

```
biasCorr(object = NULL, L = 0L)
```

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.

Value

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

References

Czarnowske, D. and Stammann, A. (2019). "Binary Choice Models with High-Dimensional Individual and Time Fixed Effects". ArXiv e-prints.

Fernandez-Val, I. and Weidner, M. (2016). "Individual and time effects in nonlinear panel models with large N, T". Journal of Econometrics, 192(1), 291-312.

Fernandez-Val, I. and Weidner M. (2018). "Fixed effects estimation of large-t panel data models". Annual Review of Economics, 10, 109-138.

Hahn, J. and Kuersteiner, G. (2011). "Bias reduction for dynamic nonlinear panel models with fixed effects". Econometric Theory, 27(6), 1152-1191.

Neyman, J. and Scott, E. L. (1948). "Consistent estimates based on partially consistent observations". Econometrica, 16(1), 1-32.

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)

# Compute bias-corrected average partial effects
mod.ape <- getAPEs(mod.bc)
summary(mod.ape)
```

`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

```
## 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.feglm *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

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

Arguments

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

Value

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

See Also

feglm

---

feglm *Efficiently fit glm's with high-dimensional k-way fixed effects*

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

```
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 ~ 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 ~ 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. Similiar 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 $\boldsymbol{\beta} = \mathbf{0}$.
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.

Value

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.

Stammann, A., F. Heiss, and D. McFadden (2016). "Estimating Fixed Effects Logit Models with Large Panel Data". Working paper.

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

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)
summary(mod)
```

`feglm.nb` *Efficiently fit negative binomial glm's with high-dimensional k-way fixed effects*

Description

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

Usage

```
feglm.nb(formula = NULL, data = NULL, beta.start = NULL,
  eta.start = NULL, init.theta = NULL, link = c("log", "identity",
  "sqrt"), control = NULL)
```

Arguments

formula	see feglm.
data	see feglm.
beta.start	see feglm.
eta.start	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".
control	see feglm.

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.

Stammann, A., F. Heiss, and D. McFadden (2016). "Estimating Fixed Effects Logit Models with Large Panel Data". Working paper.

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

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: $\|\|\mathbf{v}_i - \mathbf{v}_{i-1}\|\|_2 < tol\|\|\mathbf{v}_{i-1}\|\|$. Default is `1.0e-05`.
`rho.tol`	tolerance level for the stephalving in the maximization routine. Stephalving only takes place if the deviance in iteration r is larger than the one of the previous iteration. If this is the case, $\|\|\boldsymbol{\beta}_r - \boldsymbol{\beta}_{r-1}\|\|_2$ is halfed until the deviance is less or numerically equal compared to the deviance of the previous iteration. Stephalving fails if the the following condition holds: $\rho < tol$, where ρ is the stepcorrection factor. If stephalving 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 `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`	depreacted; termination conditions is now similar to `glm`.
`...`	arguments passed to the deprecated function `feglmControl`.

Value

The function `feglmControl` returns a named list of control parameters.

See Also

feglm

`fitted.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

`getAPEs` *Compute average partial effects after fitting binary choice models with two-way error component*

Description

`getAPEs` 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, 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 follows: $(n^* - n)/(n^* - 1)$, 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.
weak.exo	logical indicating if some of the regressors are assumed to be weakly exogenous (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

Cruz-Gonzalez, M., Fernandez-Val, I., and Weidner, M. (2017). "Bias corrections for probit and 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 Individual and Time Fixed Effects". ArXiv e-prints.

Fernandez-Val, I. and Weidner, M. (2016). "Individual and time effects in nonlinear panel models with large N, T". Journal of Econometrics, 192(1), 291-312.

Fernandez-Val, I. and Weidner M. (2018). "Fixed effects estimation of large-t panel data models". Annual Review of Economics, 10, 109-138.

Neyman, J. and Scott, E. L. (1948). "Consistent estimates based on partially consistent observations". Econometrica, 16(1), 1-32.

See Also

biasCorr, 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)

# Compute average partial effects
mod.ape <- getAPEs(mod)
```

```
summary(mod.ape)
```

`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 i if $\|\|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i-1}\|\|_2 < tol\|\|\boldsymbol{\alpha}_{i-1}\|\|_2$. Default is `1.0e-08`.

Value

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

References

Gaure, S. (n. d.). "Multicollinearity, identification, and estimable functions". Unpublished.

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 *Predict method for* feglm *fits*

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 to 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.

See Also

feglm

print.APEs *Print* APEs

Description

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") -
...	other arguments.

See Also

getAPEs

`print.feglm` *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") -
...	other arguments.

See Also

feglm

`print.summary.APEs` *Print* `summary.APEs`

Description

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

Usage

```
## 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") -
...	other arguments.

See Also

getAPEs

---

print.summary.feglm *Print* summary.feglm

Description

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

Usage

```
## 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") -
...	other arguments.

See Also

feglm

---

simGLM *Generate an artificial data set for some GLM's with two-way fixed effects*

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 ($\boldsymbol{\eta}$) is generated as follows:

$$\eta_{it} = \mathbf{x}_{it}'\boldsymbol{\beta} + \alpha_i + \gamma_t \,,$$

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

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

Usage

```
simGLM(n = NULL, t = NULL, seed = NULL, model = 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".

Value

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

See Also

feglm

---

summary.APEs *Summarizing models of class* APEs

Description

Summary statistics for objects of class "APEs".

Usage

```
## 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 *Summarizing models of class* feglm

Description

Summary statistics for objects of class "feglm".

Usage

```
## S3 method for class 'feglm'
summary(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 refered 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

Cameron, C., J. Gelbach, and D. Miller (2011). "Robust Inference With Multiway Clustering". Journal of Business & Economic Statistics 29(2).

See Also

feglm

vcov.feglm *Extract estimates of the covariance matrix*

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

```
## 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 refered 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

Cameron, C., J. Gelbach, and D. Miller (2011). "Robust Inference With Multiway Clustering". Journal of Business & Economic Statistics 29(2).

See Also

feglm

Index