

Package ‘regress’

April 19, 2009

Version 1.1-2

Date 2009-2-10

Title Gaussian linear models with linear covariance structure

Author David Clifford <david.clifford@csiro.au>, Peter McCullagh <pmcc@galton.uchicago.edu>

Maintainer David Clifford <david.clifford@csiro.au>

Description Functions to fit Gaussian linear model by maximising the residual log likelihood where the covariance structure can be written as a linear combination of known matrices. Can be used for multivariate models and random effects models. Easy straight forward manner to specify random effects models, including random interactions.

License GPL

URL <http://www.csiro.au>

SystemRequirements

Repository CRAN

Date/Publication 2009-02-10 08:14:40

R topics documented:

print.regress	2
regress	2
summary.regress	5

Index	6
--------------	----------

```
print.regress
```

Print a summary of regress output

Description

Summarises output from `regress`.

Usage

```
print.regress(x,digits=4,fixed.effects=F, ...)
```

Arguments

<code>x</code>	Output from <code>regress</code> .
<code>digits</code>	Number of significant decimal places to print.
<code>fixed.effects</code>	Logical, include the summary of fixed effects also.
<code>...</code>	additional arguments to be passed

Author(s)

David Clifford

```
regress
```

Fit a Gaussian Linear Model with Linear Covariance Structure

Description

Fits Gaussian linear models in which the covariance structure can be expressed as a linear combination of known matrices. For example, block effects models and spatial models that include a nugget effect. Fits model by maximising the residual log likelihood, also known as the REML log likelihood or restricted log likelihood. Uses a Newton-Raphson algorithm to maximise the residual log likelihood.

Usage

```
regress(formula, Vformula, identity=TRUE, start=NULL, fraction=1,
pos, verbose=0, gamVals=NULL, maxcyc=50, tol=1e-4, data, print.level)
```

Arguments

<code>formula</code>	a symbolic description of the model to be fitted. The details of model specification are the same as for <code>lm</code>
<code>Vformula</code>	Specifies the matrices to include in the covariance structure. Each term is either a symmetric matrix, or a factor. Independent Gaussian random effects are included by passing the corresponding block factor.
<code>identity</code>	Logical variable, includes the identity as the final matrix of the covariance structure. Default is TRUE
<code>start</code>	Specify the variance components at which the Newton-Raphson algorithm starts. Default value is <code>rep(var(y), k)</code> .
<code>fraction</code>	The proportion of each step to take. Default value is 1. Useful to prevent taking huge steps in the first few iterations.
<code>pos</code>	logical vector of length <code>k</code> , where <code>k</code> is the number of matrices in the covariance structure. Indicates which variance components are positive (TRUE) and which are real (FALSE). Important for multivariate problems.
<code>verbose</code>	Controls level of time output, takes values 0, 1 or 2, Default is 0, level 1 gives parameter estimates and value of log likelihood at each stage.
<code>gamVals</code>	When <code>k=2</code> , the marginal log likelihood based on the residual configuration statistic (see Tunnicliffe Wilson(1989)), is evaluated first at $(1-\text{gam}) V_1 + \text{gam} V_2$ for each value of <code>gam</code> in <code>gamVals</code> , a set of values from the unit interval. Subsequently the Newton-Raphson algorithm is started at variance components corresponding the the value of <code>gam</code> that has the highest marginal log likelihood. This is overridden if <code>start</code> is specified.
<code>maxcyc</code>	Maximum number of cycles allowed. Default value is 50. A warning is output to the screen if this is reached before convergence.
<code>tol</code>	Convergence criteria. If the change in residual log likelihood for one cycle is less than <code>tol</code> the algorithm finishes. Default value is $1e-4$.
<code>data</code>	an optional data frame containing the variables in the model. By default the variables are taken from <code>'environment(formula)'</code> , typically the environment from which <code>'regress'</code> is called.
<code>print.level</code>	Deprecated

Details

As the code is running it outputs the variance components, and the residual log likelihood at each iteration.

To avoid confusion over terminology. I define variance components to be the multipliers of the matrices and variance parameters to the parameter space over which the Newton-Raphson algorithm is run. I can force a component to be positive be defining the corresponding variance parameter on the log scale.

All output to the screen is for variance components (i.e. the multiples of the matrices). Values for `start` are on the variance component scale. Use `pos` to force certain variance components to be positive.

NOTE: The final stage of the algorithm converts the estimates of the variance components and the Fisher Information to the usual linear scale, i.e. as if `pos` were a vector of zeroes.

NOTE: No `predict` functionality is provided with `regress` due to some ambiguity. Are we predicting conditional on the observed data. Are we predicting observations from the fitted model itself? It is all normal anyway so it is straightforward, see our paper on `regress`.

Value

<code>trace</code>	Matrix with one row for each iteration of algorithm. Each row contains the residual log likelihood, marginal log likelihood, variance parameters and increments.
<code>llik</code>	Value of the marginal log likelihood at the point of convergence.
<code>cycle</code>	Number of cycles to convergence.
<code>rdf</code>	Residual degrees of freedom.
<code>beta</code>	Estimate of the linear effects.
<code>beta.cov</code>	Estimate of the covariance structure for terms in <code>beta</code> .
<code>beta.se</code>	Standard errors for terms in <code>beta</code> .
<code>sigma</code>	Variance component estimates, interpretation does not depend on value of <code>pos</code>
<code>sigma.cov</code>	Covariance matrix for the variance component estimates based on the Fisher Information at the point of convergence.
<code>W</code>	Inverse of covariance matrix at point of convergence.
<code>Q</code>	$I - X^T(X^TWX)^{-1}X^TW$ at point of convergence.
<code>fitted</code>	$X\beta$, the fitted values.
<code>predicted</code>	If <code>identity=TRUE</code> , decompose <code>y</code> into the part associated with the identity and that associated with the rest of the variance structure, this second part is the predicted values. If $\text{Sigma} = V1 + V2$ at point of convergence then $y = V1 W y + V2 W y$ is the decomposition.
<code>pos</code>	Indicator for the scale for each variance parameter.
<code>Vnames</code>	Names associated with each variance component, used in <code>print.regress</code> .
<code>formula</code>	Copy of formula
<code>Vformula</code>	Updated version of <code>Vformula</code> to include identity if necessary
<code>model</code>	Response, covariates and matrices/factors to be used for model fitting

Author(s)

David Clifford, Peter McCullagh.

References

G. Tunnicliffe Wilson, "On the use of marginal likelihood in time series model estimation." JRSS B, Vol 51, No 1, 1989, 15-27. D. Clifford and P. McCullagh, "The regress Library" R News.

Examples

```
## Example of Random Effects model from Venables and Ripley, page 205
library("nlme")
##library("regress")
data(Oats)
names(Oats) <- c("B", "V", "N", "Y")
Oats$N <- as.factor(Oats$N)

## Using regress
oats.reg <- regress(Y~N+V, ~B+I(B:V), identity=TRUE, verbose=1, data=Oats)
summary(oats.reg)

## Using lme
oats.lme <- lme(Y~N+V, random=~1|B/V, data=Oats, method="REML")
summary(oats.lme)
```

summary.regress *Summarise a Gaussian Linear Model from regress*

Description

Summarises output from `regress`.

Usage

```
summary.regress(object, ...)
```

Arguments

<code>object</code>	Output from <code>regress</code> .
<code>...</code>	additional parameters as needed

Details

All summary information included in the `regress` function itself. This function is included for completeness.

Author(s)

David Clifford

Index

*Topic **models**

print.regress, 1
regress, 2
summary.regress, 5

*Topic **multivariate**

print.regress, 1
regress, 2
summary.regress, 5

*Topic **regression**

print.regress, 1
regress, 2
summary.regress, 5

print.regress, 1

regress, 2

summary.regress, 5