

Package ‘gemmR’

August 24, 2015

Type Package

Title General Monotone Model

Version 1.3-3

Date 2015-08-23

LazyData yes

Description An R-language implementation of the General Monotone Model proposed by Michael Dougherty and Rick Thomas. It is a procedure for estimating weights for a set of independent predictors that minimize the rank-order inversions between the model predictions and some outcome.

License GPL-3

Depends Rcpp(>= 0.11)

Suggests knitr

LinkingTo Rcpp

VignetteBuilder knitr

BugReports <https://github.com/jchrascz/gemmR/issues>

Collate 'gemmquick.R' 'RcppExports.R'

NeedsCompilation yes

Author Jeffrey Chrabaszcz [aut, cre],

Joe Tidwell [aut],

Michael Dougherty [aut]

Maintainer Jeffrey Chrabaszcz <jchrabaszcz@gmail.com>

R topics documented:

gemmR-package	2
convergencePlot	3
culture	3
gemm	4
gemmEst	6
gemmFitRcppI	8
genAlg	8
list2gemm	9
predict.gemm	10
tauTest	11
Index	12

gemmR-package

General Monotone Model in R (gemmR)

Description

The gemmR package provides functions for fitting and analyzing General Monotone Models (GeMM). The GeMM is a statistical algorithm for predicting rank orders from a set of k predictors. As shown in Dougherty and Thomas (2012, Psychological Review), GeMM is unaffected by any monotonic transformation of the criterion variable, unaffected by non-linear monotone relationships, and shows better power and predictive accuracy than Least-Squares regression procedures in a variety of contexts. The primary gemmR function is [gemm](#).

Details

Package: gemmR
Type: Package
Version: 1.3.01
Date: 2014-09-08
License: GPL-3

license

This file is part of gemmR. gemmR is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

gemmR is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with gemmR. If not, see <http://www.gnu.org/licenses/>.

Author(s)

Jeffrey S. Chrabaszcz
Joe Tidwell
Michael R. Dougherty

References

Dougherty, M. R., & Thomas, R. P. (2012). Robust decision making in a nonlinear world. *Psychological Review*, 119(2), 321.

convergencePlot	<i>convergencePlot function for visualizing genetic algorithm in gemmR</i>
-----------------	--

Description

This function plots convergence for gemm objects that were called with the "check.convergence" argument.

Usage

```
convergencePlot(beta, fit.metric, ...)
```

Arguments

beta	matrix of BIC values for the best model in each generation.
fit.metric	selected metric for a given model, specified when <code>gemm</code> is run.
...	Additional arguments to be passed to lower level plotting functions.

Details

Generally does not work well with `n.chains < 10`.

Examples

```
data(mtcars)
gemm.model <- gemm(mpg ~ disp + hp, data = mtcars, check.convergence = TRUE,
  seed.metric = FALSE, n.chains = 3, n.gens = 3, n.beta = 200)
with(gemm.model, convergencePlot(converge.fit.metric, fit.metric))
```

culture	<i>Geographic and cultural predictors of country homicide</i>
---------	---

Description

The culture data contains 92 rows and 5 columns from a study on the predictors of homicide rate by country.

Usage

```
culture
```

Format

An object of class `data.frame` containing the following columns:

country A factor vector containing the country names correspond to each row of observations.
murder.rate Numeric vector giving murder rate as expected annual murders per 10,000 people.
pasture Numeric vector giving the percentage of geographic area used as pasture.
gini Numeric vector of Gini coefficients, a measure of income disparity.
gnp Numeric vector of gross national products in US dollars.

Source

1. Dougherty, M.R., Thomas, R.P., Brown, R.P., Chrabaszcz, J.S., and Tidwell, J.W. (2014). An introduction to the general monotone model with application to two problematic datasets. Sociological Methods.
2. Brown, R. P., & Osterman, L. L. (2012). Culture of Honor, Violence. The Oxford handbook of evolutionary perspectives on violence, homicide, and war, 218.

gemm

Fit General Monotone Models.

Description

gemm is used to fit general monotone models. By default, the function will generate metric weights that minimize rank order inversion between the model predictions and a response variable, subject to a parsimony correction. Optional argument passed to [gemmEst](#).

Usage

```
## S3 method for class formula
gemm(formula, data=list(), ...)

## Default S3 method:
gemm(x, k.pen, n.chains = 4, fit.metric = "bic", ...)
```

Arguments

formula	an object of the class formula .
data	optional data frame, list, or environment.
x	must be data frame, first column is treated as dependent variable. Not needed if formula is supplied.
k.pen	vector of integers giving penalty equivalent to main effects for any interaction terms. Calculated by gemm.formula or defaults to no special interaction weighting.
n.chains	number or replications of the sampling process, used to assess various starting conditions
fit.metric	Value to optimize with genetic algorithm. Currently accepts “bic”, “aic”, and “tau”.
...	Additional arguments to be passed to lower level fitting functions (see below).

Details

Models for gemm are constructed with syntax similar to [lm](#). By default, gemm will use random search to minimize penalized rank order inversions between model predictions and a response variable. This is accomplished by generating candidate weights using [genAlg](#) and deriving a BIC based on transformed Kendall’s tau using [gemmFitRcppI](#). gemm may take some time to run depending on the complexity of the model. Methods for standard goodness-of-fit functions are also available and are run by [summary.gemm](#).

Value

A list with class "gemm" containing the following components:

date	system time and date for model completion.
call	the matched call.
coefficients	matrix of best weights with one row for each chain.
fitted.values	model predictions for criterion generated from weights associated with best chain.
residuals	metric values for response minus fitted.
rank.residuals	rank response minus rank criterion.
bic	vector of Bayesian Information Criteria for estimation sample of each chain.
aic	vector of Aikaike Information Criteria for estimation sample of each chain.
r	vector of Pearson's r values for estimation sample of each chain.
tau	vector of Kendall's tau for estimation sample of each chain.
metric.betas	regression weights derived using lm .
p.vals	p-values associated with ordinary least squares regression weights.
model	data frame including modeled data.
fit.metric	sorting metric used.
cross.val.bic	vector of Bayesian Information Criteria for cross-validation sample of each chain.
cross.val.aic	vector of Aikaike Information Criteria for cross-validation sample of each chain.
cross.val.r	vector of Pearson's r values for cross-validation sample of each chain.
cross.val.tau	vector of Kendall's tau for cross-validation sample of each chain.
converge.fit.metric	matrix of "fit.metric" values with generations by chains.
converge.beta	matrix derived weights for each generation within each chain, column for each predictor.
converge.r	generations by chains matrix of Pearson's r.
formula	formula object used.

References

Dougherty, M. R., & Thomas, R. P. (2012). Robust decision making in a nonlinear world. *Psychological review*, 119(2), 321.

See Also

[genAlg](#) for search, [gemmFitRcppI](#) for fitting routine, [tauTest](#) for $O(N \log N)$ scale Kendall's tau, [convergencePlot](#) for the optional plot pane when `check.convergence = TRUE`. [gemmEst](#) may be useful in some circumstances when formula input and interaction terms are not needed.

Examples

```
## Not run:
data(culture)
gemm.model <- gemm(mpg ~ disp + cyl, data = mtcars, check.convergence = TRUE)
print(gemm.model)
plot(gemm.model)

## End(Not run)
```

gemmEst

*Fit General Monotone Models.***Description**

gemmEst is called by [gemm](#) to fit general monotone models.

Usage

```
gemmEst(input.data, output = "gemmr", n.beta = 8000, n.chains = n.chains,
n.gens = 10, save.results = FALSE, k.pen = k.pen, seed.metric = TRUE,
check.convergence = FALSE, roe = FALSE, fit.metric = fit.metric,
correction = "knp", oclo = TRUE, isTauB = FALSE)
```

Arguments

input.data	must be data frame, first column is treated as dependent variable.
output	string argument for use in naming file output. gemmEst may write a .RData file in the current working directory each time the function is called.
n.beta	number of beta vectors to generate per generation.
n.chains	number of times the fitting process will be repeated.
n.gens	number of generations per chain.
save.results	logical value to determine whether the resulting gemm object is saved to a .RData file.
k.pen	penalty term for BIC, as calculated by gemm.formula .
seed.metric	logical value to control whether genAlg is seeded with OLS regression weights or random values.
check.convergence	logical value to indicate whether BIC for each generation is retained, mostly useful for checking performance of genAlg .
roe	logical value to determine whether region of equivalence data are retained.
fit.metric	value used to order models.
correction	placeholder for correction transformations on the fit statistic, (r or tau).
oclo	logical for <i>Ordered Constrained Linear Optimization</i> . If TRUE secondary sort based on r performed on matrix of beta weights in order to maximize linear fit, given the ordered fits determined by fit.metric.
isTauB	logical for whether to include ties in the denominator of the tau calculation.

Details

Formula syntax and interaction penalty terms can be avoided by fitting data directly using gemmEst.

Value

A list with class "gemm" containing the following components:

date	system time and date for model completion.
call	the matched call.
coefficients	matrix of best weights with one row for each chain.
fitted.values	model predictions for criterion generated from weights associated with best chain.
residuals	metric values for response minus fitted.
rank.residuals	rank response minus rank criterion.
bic	vector of Bayesian Information Criteria for estimation sample of each chain.
r	vector of Pearson's r values for estimation sample of each chain.
tau	vector of Kendall's tau for estimation sample of each chain.
tau.par	vector containing the correction, incorrect, criterion ties, predictor ties, and both between the weighted cues and outcome used for model fit.
metric.betas	regression weights derived using lm .
p.vals	p-values associated with ordinary least squares regression weights.
model	data frame including modeled data.
fit.metric	sorting metric used.
cross.val.bic	vector of Bayesian Information Criteria for cross-validation sample of each chain.
cross.val.r	vector of Pearson's r values for cross-validation sample of each chain.
cross.val.tau	vector of Kendall's tau for cross-validation sample of each chain.
cross.val.tau.par	vector containing the correction, incorrect, criterion ties, predictor ties, and both between the weighted cues and outcome used for crossvalidation.
converge.fit.metric	matrix of "fit.metric" values with generations by chains.
converge.beta	matrix derived weights for each generation within each chain, column for each predictor.
converge.r	generations by chains matrix of Pearson's r.
formula	formula object used.

References

Dougherty, M. R., & Thomas, R. P. (2012). Robust decision making in a nonlinear world. *Psychological review*, 119(2), 321.

See Also

[gemm](#) for normal use. [genAlg](#) for search. [gemmFitRcppI](#) for fitting routine, [tauTest](#) for $O(N \log N)$ scale Kendall's tau, [convergencePlot](#) for the optional plot pane when `check.convergence = TRUE`.

Examples

```
data(mtcars)
gemm.model <- gemm(mpg ~ disp + cyl, data = mtcars,
  check.convergence = TRUE, n.chains = 3, n.gens = 3, n.beta = 200)
print(gemm.model)
plot(gemm.model)
```

gemmFitRcppI

gemmFitRcppI

Description

This function is run each generation to calculate fit between weighted predictors and response based on `fit.metric`, (weights generated by [genAlg](#)), and criterion. Calls a number of Rcpp functions that are hidden from R.

Usage

```
gemmFitRcppI(n, betas, data, p, kCor, correction, isTauB)
```

Arguments

<code>n</code>	double, number of observations in dataset.
<code>betas</code>	numeric matrix of random weights.
<code>data</code>	numeric matrix, dataset constructed based on original gemm.formula call. First column is criterion variable.
<code>p</code>	double, number of potential predictors in data.
<code>kCor</code>	numeric vector, penalty term for given vector of weights. Will always equal number of non-zero weights unless interactions are included.
<code>correction</code>	character string, names correction to tau for BIC calculation. Currently set as "knp".
<code>isTauB</code>	boolean to set BIC computation based on tau-a or tau-b.

Value

<code>fit</code>	list of named vectors for BICtau, BICr, tau, and r.
------------------	---

See Also

[gemm](#) for full model-fitting function, [tauTest](#) for quick Kendall's tau.

genAlg

genAlg

Description

Genetic algorithm for [gemmR](#).

Usage

```
genAlg(metricbeta, nbeta, nsuperelites, p, gens, bestmodels, seedmetric)
```


Arguments

metricbeta	Weights derived using multiple regression. Overwritten if seedmetric is FALSE.
nbeta	Number of candidate weight vectors in each generation.
nsuperelites	Number of candidate weight vectors to involve in permutation for reps > 1.
p	Number of potential predictors.
gens	Generation number. For gens == 1, entirely new weights are generated. When gens > 1, bestmodels are used to generate permutations.
bestmodels	Matrix of best candidate weight vectors from previous generation.
seedmetric	If TRUE, multiple regression weights are used to seed the genetic algorithm. Otherwise, random weights are used.

Details

Currently has fixed scaling factors so predictors should be normalized. Heavily seeded with zero values to interact properly with AIC/BIC calculation.

Value

allbetas	returns a matrix of candidate weights with rows for each predictor and columns for each unique vector of betas.
----------	---

See Also

[gemm](#) for full model-fitting function, [tauTest](#) for quick Kendall's tau.

Examples

```
p <- 4
gen.alg <- genAlg(matrix(rnorm(p), nrow = p), 5, 2, p, 1, matrix(1), TRUE)
```

list2gemm	<i>Collapse a uniform list of gemm objects.</i>
-----------	---

Description

list2gemm is used to generate a normal [gemm](#) object from a list of gemm objects with identical input options and n.chains = 1.

Usage

```
list2gemm(gemm.list)
```

Arguments

gemm.list	a uniform list of gemm objects, as generated by a parallelizing function.
-----------	---

Details

Will generally only be used in conjunction with a parallel processing package, otherwise equivalent to specifying higher n.chains argument in [gemm](#).

Value

Output is a standard gemm object.

See Also

[gemm](#)

Examples

```
library(gemmR)
data(mtcars)
fit <- list()
for (i in 1:3) {
  fit[[i]] <- gemm(mpg ~ disp + cyl, data = mtcars, n.chains = 1, n.gens = 3, n.beta = 200)
}
gemm.model <- list2gemm(fit)
summary(gemm.model)
```

predict.gemm

Predict method for general monotone models.

Description

predict.gemm is used to get fitted values from gemm objects. Can also be used to get correct, incorrect, and ties for all paired comparisons of cases in model.

Usage

```
## S3 method for class gemm
predict(object, newdata = NULL, tie.struct = FALSE, ...)
```

Arguments

object	an object of the class gemm .
newdata	optional data frame, currently not functional.
tie.struct	option to return corrects, incorrects, and ties based on fitted values.
...	other arguments to internal functions.

Details

Will eventually include functionality to generate estimates based on new data.

Value

Output depends on tie.struct argument. Output is either:

y	matrix of fitted values or a data.frame of correct, incorrect, and tied pairs.
---	--

See Also

[gemm](#)

Examples

```
data(mtcars)
gemm.model <- gemm(mpg ~ disp + cyl, data = mtcars, check.convergence = TRUE, n.beta = 200)
predict(gemm.model, tie.struct = TRUE)
```

tauTest	<i>quick Kendall's tau</i>
---------	----------------------------

Description

This is a better-scaling implementation of Kendall's tau than is used by [cor.test](#).

Usage

```
tauTest(arr1, arr2, length)
```

Arguments

arr1	numeric vector
arr2	numeric vector
length	length of arr1 or arr2, (both must be the same length).

Value

Returns a list with the following components:

tau-a	Kendall's tau-a, which includes tied pairs of all types in the denominator.
tau-b	Kendall's tau-b, which does not penalize for ties.
n.pairs	Number of paired comparisons.
n.ties.1	Number of ties in arr1.
n.ties.2	Number of ties in arr2.
n.ties.both	Number of ties for both arr1 and arr2.
n.dis	Total discordances for all paired comparisons.
n.con	Total concordances for all paired comparisons.

Index

*Topic **ordinal, regression**

- gemm, [4](#)
- gemmEst, [6](#)
- list2gemm, [9](#)
- predict.gemm, [10](#)

*Topic **package**

- gemmR-package, [2](#)

*Topic

- gemm, [4](#)
- gemmEst, [6](#)
- gemmR-package, [2](#)
- list2gemm, [9](#)
- predict.gemm, [10](#)

convergencePlot, [3](#), [5](#), [7](#)

cor.test, [11](#)

culture, [3](#)

deviance.gemm (gemm), [4](#)

formula, [4](#), [5](#), [7](#)

gemm, [2](#), [3](#), [4](#), [6–10](#)

gemm.formula, [4](#), [6](#), [8](#)

gemmEst, [4–6](#), [6](#)

gemmFitRcppI, [4](#), [5](#), [7](#), [8](#)

gemmR, [8](#)

gemmR (gemmR-package), [2](#)

gemmR-package, [2](#)

genAlg, [4–8](#), [8](#)

list2gemm, [9](#)

lm, [4](#), [5](#), [7](#)

logLik.gemm (gemm), [4](#)

nobs.gemm (gemm), [4](#)

plot.gemm (gemm), [4](#)

predict.gemm, [10](#)

print.gemm (gemm), [4](#)

print.summary.gemm (gemm), [4](#)

summary.gemm, [4](#)

summary.gemm (gemm), [4](#)

tauTest, [5](#), [7–9](#), [11](#)