Package ‘RCAL’

March 6, 2019

Title Regularized Calibrated Estimation

Version 1.0

Author Zhiqiang Tan

Maintainer Zhiqiang Tan <ztan@stat.rutgers.edu>

URL http://www.stat.rutgers.edu/~ztan

Description
Regularized calibrated estimation for causal inference and missing-data problems with high-dimensional data, based on Tan (2017) <arXiv:1710.08074> and Tan (2018) <arXiv:1801.09817>.

Depends R (>= 3.5.0), trust

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-06 17:32:58 UTC

R topics documented:

`RCAL-package` *RCAL: Regularized calibrated estimation*

Description

Regularized calibrated estimation for causal inference and missing-data problems with high-dimensional data.

Details

The R package `RCAL` - version 1.0 can be used for two main tasks:

- to estimate the mean of an outcome in the presence of missing data,
- to estimate the average treatment effects in causal inference.

There are 3 high-level functions provided for the first task:

- `mn.nreg`: inference using non-regularized calibrated estimation,
- `mn.regu.cv`: inference using regularized calibrated estimation based on cross validation,
- `mn.regu.path`: inference using regularized calibrated estimation along a regularization path.

The first function `mn.nreg` is appropriate only in relatively low-dimensional settings, whereas the functions `mn.regu.cv` and `mn.regu.path` are designed to deal with high-dimensional data (namely, the number of covariates close to or greater than the sample size). In parallel, there are 3 functions for the second task, `ate.nreg`, `ate.regu.cv`, and `ate.regu.path`. These functions can also be used to perform inference for the average treatment effects on the treated or on the untreated. Currently, the treatment is assumed to be binary (i.e., untreated or treated). Extensions to multi-valued treatments will be incorporated in later versions.

The package also provides lower-level functions, including `glm.nreg` to implement non-regularized M-estimation and `glm.regu` to implement Lasso regularized M-estimation for fitting generalized linear models currently with continuous or binary outcomes. The latter function `glm.regu` uses an active-set descent algorithm, which enjoys a finite termination property for solving least-squares Lasso problems.

See the the vignette for more details.

`ate.aipw` *Augmented inverse probability weighted estimation of population means*

Description

This function implements augmented inverse probability weighted (IPW) estimation of average treatment effects (ATEs), provided both fitted propensity scores and fitted values from outcome regression.

Usage

```
ate.aipw(y, tr, mfp, mfo, off = NULL)
```

Arguments

`y`	An n x 1 vector of observed outcomes.
`tr`	An n x 1 vector of treatment indicators (=1 if treated or 0 if untreated).
`mfp`	An n x 2 matrix of fitted propensity scores for untreated (first column) and treated (second column).
`mfo`	An n x 2 matrix of fitted values from outcome regression, for untreated (first column) and treated (second column).
`off`	A 2 x 1 vector of offset values (e.g., the true values in simulations) used to calculate the z-statistics.

Value

`one`	A 2 x 1 vector of direct IPW estimates of 1.
`ipw`	A 2 x 1 vector of ratio IPW estimates of means.
`or`	A 2 x 1 vector of outcome regression estimates of means.
`est`	A 2 x 1 vector of augmented IPW estimates of means.
`var`	The estimated variances associated with the augmented IPW estimates of means.
`ze`	The z-statistics for the augmented IPW estimates of means, compared to `off`.
`diff`	The augmented IPW estimate of ATE.
`diff.var`	The estimated variance associated with the augmented IPW estimate of ATE.
`diff.ze`	The z-statistic for the augmented IPW estimate of ATE.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

`ate.ipw` *Inverse probability weighted estimation of average treatment effects*

Description

This function implements inverse probability weighted (IPW) estimation of average treatment effects (ATEs), provided fitted propensity scores.

Usage

```
ate.ipw(y, tr, mfp)
```

Arguments

`y`	An n x 1 vector of observed outcomes.
`tr`	An n x 1 vector of treatment indicators (=1 if treated or 0 if untreated).
`mfp`	An n x 2 matrix of fitted propensity scores for untreated (first column) and treated (second column).

Value

`one`	The direct IPW estimates of 1.
`est`	The ratio IPW estimates of means.
`diff`	The ratio IPW estimate of ATE.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

`ate.nreg` *Model-assisted inference for average treatment effects without regularization*

Description

This function implements model-assisted inference for average treatment effects, using non-regularized calibrated estimation.

Usage

```
ate.nreg(y, tr, x, ploss = "cal", yloss = "gaus", off = NULL)
```

Arguments

- `y` An n x 1 vector of observed outcomes.
- `tr` An n x 1 vector of treatment indicators (=1 if treated or 0 if untreated).
- `x` An n x p matix of covariates, used in both propensity score and outcome regression models.
- `ploss` A loss function used in propensity score estimation (either "ml" or "cal").
- `yloss` A loss function used in outcome regression (either "gaus" for continuous outcomes or "ml" for binary outcomes).
- `off` A 2 x 1 vector of offset values (e.g., the true values in simulations) used to calculate the z-statistics from augmented IPW estimation.

Details

For calibrated estimation, two sets of propensity scores are separately estimated for the untreated and treated as discussed in Tan (2017, 2019). See also **Details** for `mn.nreg`.

Value

- `ps` A list containing the results from fitting the propensity score model by `glm.nreg`.
- `mfp` An n x 2 matrix of fitted propensity scores for untreated (first column) and treated (second column).
- `or` A list containing the results from fitting the outcome regression model by `glm.nreg`.
- `mfo` An n x 2 matrix of fitted values from outcome regression, for untreated (first column) and treated (second column).
- `est` A list containing the results from augmented IPW estimation by `ate.aipw`.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

# include only 10 covariates
x2 <- x[,1:10]
```

```
ate.cal <- ate.nreg(y, tr, x2, ploss="cal", yloss="gaus")
matrix(unlist(ate.cal$est), ncol=2, byrow=TRUE,
dimnames=list(c("one", "ipw", "or", "est", "var", "ze",
"diff.est", "diff.var", "diff.ze"), c("untreated", "treated")))
```

`ate.regu.cv` *Model-assisted inference for average treatment effects based on cross validation*

Description

This function implements model-assisted inference for average treatment effects, using regularized calibrated estimation based on cross validation.

Usage

```
ate.regu.cv(fold, nrho = NULL, rho.seq = NULL, y, tr, x,
  ploss = "cal", yloss = "gaus", off = NULL, ...)
```

Arguments

`fold`	A vector of length 2 giving the fold numbers for cross validation in propensity score estimation and outcome regression respectively.
`nrho`	A vector of length 2 giving the numbers of tuning parameters searched in cross validation.
`rho.seq`	A list of two vectors giving the tuning parameters in propensity score estimation (first vector) and outcome regression (second vector).
`y`	An n x 1 vector of obseved outcomes.
`tr`	An n x 1 vector of treatment indicators (=1 if treated or 0 if untreated).
`x`	An n x p matix of covariates, used in both propensity score and outcome regression models.
`ploss`	A loss function used in propensity score estimation (either "ml" or "cal").
`yloss`	A loss function used in outcome regression (either "gaus" for continuous outcomes or "ml" for binary outcomes).
`off`	A 2 x 1 vector of offset values (e.g., the true values in simulations) used to calculate the z-statistics from augmented IPW estimation.
`...`	Additional arguments to `glm.regu.cv`.

Details

For calibrated estimation, two sets of propensity scores are separately estimated for the untreated and treated as discussed in Tan (2017, 2019). See also **Details** for `mn.regu.cv`.

Value

ps	A list containing the results from fitting the propensity score model by glm.regu.cv.
mfp	An n x 2 matrix of fitted propensity scores for untreated (first column) and treated (second column).
or	A list containing the results from fitting the outcome regression model by glm.regu.cv.
mfo	An n x 2 matrix of fitted values from outcome regression, for untreated (first column) and treated (second column).
est	A list containing the results from augmented IPW estimation by ate.aipw.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

ate.cv.rcal <- ate.regu.cv(fold=5*c(1,1), nrho=(1+10)*c(1,1), rho.seq=NULL, y, tr, x,
                           ploss="cal", yloss="gaus")

matrix(unlist(ate.cv.rcal$est), ncol=2, byrow=TRUE,
dimnames=list(c("one", "ipw", "or", "est", "var", "ze",
"diff.est", "diff.var", "diff.ze"), c("untreated", "treated")))
```

---

ate.regu.path — *Model-assisted inference for average treatment effects along regularization paths*

Description

This function implements model-assisted inference for average treatment effects, using regularized calibrated estimation along regularization paths for propensity score (PS) estimation while based on cross validation for outcome regression (OR).

Usage

```
ate.regu.path(fold, nrho = NULL, rho.seq = NULL, y, tr, x,
  ploss = "cal", yloss = "gaus", off = NULL, ...)
```

Arguments

`fold`	A vector of length 2, with the second component giving the fold number for cross validation in outcome regression. The first component is not used.
`nrho`	A vector of length 2 giving the number of tuning parameters in a regularization path for PS estimation and that in cross validation for OR.
`rho.seq`	A list of two vectors giving the tuning parameters for propensity score estimation (first vector) and outcome regression (second vector).
`y`	An n x 1 vector of observed outcomes.
`tr`	An n x 1 vector of treatment indicators (=1 if treated or 0 if untreated).
`x`	An n x p matix of covariates, used in both propensity score and outcome regression models.
`ploss`	A loss function used in propensity score estimation (either "ml" or "cal").
`yloss`	A loss function used in outcome regression (either "gaus" for continuous outcomes or "ml" for binary outcomes).
`off`	A 2 x 1 vector of offset values (e.g., the true values in simulations) used to calculate the z-statistics from augmented IPW estimation.
`...`	Additional arguments to `glm.regu.cv` and `glm.regu.path`.

Details

See **Details** for `ate.regu.cv`.

Value

`ps`	A list of 2 objects, giving the results from fitting the propensity score model by `glm.regu.path` for untreated (first) and treated (second).
`mfp`	A list of 2 matrices of fitted propensity scores, along the PS regularization path, for untreated (first matrix) and treated (second matrix).
`or`	A list of 2 lists of objects for untreated (first) and treated (second), where each object gives the results from fitting the outcome regression model by `glm.regu.cv` for a PS tuning parameter.
`mfo`	A list of 2 matrices of fitted values from outcome regression based on cross validation, along the PS regularization path, for untreated (first matrix) and treated (second matrix).
`est`	A list containing the results from augmented IPW estimation by `ate.aipw`.
`rho`	A vector of tuning parameters leading to converged results in propensity score estimation.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

ate.path.rcal <- ate.regu.path(fold=5*c(0,1), nrho=(1+10)*c(1,1), rho.seq=NULL, y, tr, x,
                               ploss="cal", yloss="gaus")
ate.path.rcal$est
```

---

glm.nreg — *Non-regularied M-estimation for fitting generalized linear models*

Description

This function implements non-regularizd M-estimation for fitting generalized linear models with continuous or binary responses, including maximum likelihood, calibrated estimation, and covariate-balancing estimation in the latter case of fitting propensity score models.

Usage

```
glm.nreg(y, x, iw = NULL, loss = "cal", init = NULL)
```

Arguments

`y`	An n x 1 response vector.
`x`	An n x p matix of covariates, excluding a constant.
`iw`	An n x 1 weight vector.
`loss`	A loss function used, which can be specified as "gaus" for continuous responses, or "ml", "cal", or "bal" for binary responses.
`init`	A $(p+1)$ x 1 vector of initial values (the intercept and coefficients).

Details

Least squares estimation is implemented by calling `lm` for continuous responses (`loss`="gaus"). For binary responses, maximum likelihood estimation (`loss`="ml") is implemented by calling `glm`. Calibrated estimation (`loss`="cal") is implemented by using a trust-region algorithm in the R package **trust** to minimize the calibration loss, i.e., (8) in Tan (2017). Covariate-balancing estimation (`loss`="bal") in Imai and Ratkovic (2014) is implemented by using **trust** to minimize (38) in Tan (2017).

Value

`coef`	The $(p + 1)$ x 1 vector of estimated intercept and coefficients.
`fit`	The n x 1 vector of fitted values.
`conv`	Logical; 1 if loss="gaus" for continuous responses or convergence is obtained within 1000 iterations by `glm` with loss="ml" or `trust` with loss="cal" or "bal" for binary responses.

References

Imai, K. and Ratkovic, M. (2014) Covariate balancing propensity score, *Journal of the Royal Statistical Society*, Ser. B, 76, 243-263.

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

# include only 10 covariates
x2 <- x[,1:10]

ps.ml <- glm.nreg(y=tr, x=x2, loss="ml")
check.ml <- mn.ipw(x2, tr, ps.ml$fit)
check.ml

ps.cal <- glm.nreg(y=tr, x=x2, loss="cal")
check.cal <- mn.ipw(x2, tr, ps.cal$fit)
check.cal  # should be numerically 0

ps.bal <- glm.nreg(y=tr, x=x2, loss="bal")
check.bal <- mn.ipw(x2, tr, ps.bal$fit)
check.bal
```

glm.regu	*Regularied M-estimation for fitting generalized linear models with a fixed tuning parameter*

Description

This function implements regularized M-estimation for fitting generalized linear models with continuous or binary responses for a fixed choice of tuning parameters.

Usage

```
glm.regu(y, x, iw = NULL, loss = "cal", init = NULL, rhos,
  test = NULL, offs = NULL, id = NULL, Wmat = NULL, Rmat = NULL,
  zzs = NULL, xxs = NULL, n.iter = 100, eps = 1e-06, bt.lim = 3,
  nz.lab = NULL, pos = 10000)
```

Arguments

`y`	An n x 1 response vector.
`x`	An n x p matix of covariates, excluding a constant.
`iw`	An n x 1 weight vector.
`loss`	A loss function, which can be specified as "guas" for continuous responses, or "ml" or "cal" for binary respones.
`init`	A $(p+1)$ x 1 vector of initial values (the intercept and coefficients).
`rhos`	A p x 1 vector of Lasso tuning parameters, usually a constant vector, associated with the p coefficients.
`test`	A vector giving the indices of observations between 1 and n which are included in the test set.
`offs`	An n x 1 vector of offset values, similarly as in `glm`.
`id`	An argument which can be used to speed up computation.
`Wmat`	An argument which can be used to speed up computation.
`Rmat`	An argument which can be used to speed up computation.
`zzs`	An argument which can be used to speed up computation.
`xxs`	An argument which can be used to speed up computation.
`n.iter`	The maximum number of iterations allowed. An iteration is defined by computing an quadratic approximation and solving a least-squares Lasso problem.
`eps`	The tolerance at which the difference in the objective (loss plus penalty) values is considered close enough to 0 to declare convergence.
`bt.lim`	The maximum number of backtracking steps allowed.
`nz.lab`	A p x 1 logical vector (useful for simulations), indicating which covariates are included when calculating the number of nonzero coefficients. If `nz.lab=NULL`, then `nz.lab` is reset to a vector of 0s.
`pos`	A value which can be used to facilitate recording the numbers of nonzero coefficients with or without the restriction by `nz.lab`. If `nz.lab=NULL`, then `pos` is reset to 1.

Details

For continuous responses, this function uses an active-set descent algorithm (Osborne et al. 2000; Yang and Tan 2018) to solve the least-squares Lasso problem. For binary responses, regularized calibrated estimation is implemented using the Fisher scoring descent algorithm in Tan (2017), whereas regularized maximum likelihood estimation is implemented in a similar manner based on quadratic approximation as in the R package **glmnet**.

Value

`iter`	The number of iterations performed up to `n.iter`.
`conv`	1 if convergence is obtained, 0 if exceeding the maximum number of iterations, or -1 if exceeding maximum number of backtracking steps.
`nz`	A value defined as (nz0 * `pos` + nz1) to record the numbers of nonzero coefficients without or with the restriction (denoted as nz0 and nz1) by `nz.lab`. If `nz.lab=NULL`, then nz1 is 0, `pos` is 1, and hence `nz` is nz0.
`inter`	The estimated intercept.
`bet`	The p x 1 vector of estimated coefficients, excluding the intercept.
`fit`	The vector of fitted values in the training set.
`eta`	The vector of linear predictors in the training set.
`tau`	The p x 1 vector of generalized signs, which should be -1 or 1 for a negative or positive estimate and between -1 and 1 for a zero estimate.
`obj.train`	The average loss in the training set.
`pen`	The Lasso penalty of the estimates.
`obj`	The average loss plus the Lasso penalty.
`fit.test`	The vector of fitted values in the test set.
`eta.test`	The vector of linear predictors in the test set.
`obj.test`	The average loss in the test set.
`id`	This can be re-used to speed up computation.
`Wmat`	This can be re-used to speed up computation.
`Rmat`	This can be re-used to speed up computation.
`zzs`	This can be re-used to speed up computation.
`xxs`	This can be re-used to speed up computation.

References

Osborne, M., Presnell, B., and Turlach, B. (2000) A new approach to variable selection in least squares problems, *IMA Journal of Numerical Analysis*, 20, 389-404.

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Yang, T. and Tan, Z. (2018) Backfitting algorithms for total-variation and empirical-norm penalized additive modeling with high-dimensional data, *Stat*, 7, e198.

Tibshirani, R. (1996) Regression shrinkage and selection via the Lasso, *Journal of the Royal Statistical Society*, Ser. B, 58, 267-288.

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

### Example 1: linear regression
# rhos should be a vector of length p, even though a constant vector
out.rgaus <- glm.regu(y[tr==1], x[tr==1,], rhos=rep(.05,p), loss="gaus")

# the intercept
out.rgaus$inter

# the estimated coefficients and generalized signs; the first 10 are shown
cbind(out.rgaus$bet, out.rgaus$tau)[1:10,]

# the number of nonzero coefficients
out.rgaus$nz

### Example 2: logistic regression using likelihood loss
out.rml <- glm.regu(tr, x, rhos=rep(.01,p), loss="ml")
out.rml$inter
cbind(out.rml$bet, out.rml$tau)[1:10,]
out.rml$nz

### Example 3: logistic regression using calibration loss
out.rcal <- glm.regu(tr, x, rhos=rep(.05,p), loss="cal")
out.rcal$inter
cbind(out.rcal$bet, out.rcal$tau)[1:10,]
out.rcal$nz
```

`glm.regu.cv` *Regularied M-estimation for fitting generalized linear models based on cross validation*

Description

This function implements regularized M-estimation for fitting generalized linear models with binary or contiunous responses based on cross validation.

Usage

```
glm.regu.cv(fold, nrho = NULL, rho.seq = NULL, y, x, iw = NULL,
  loss = "cal", n.iter = 100, eps = 1e-06, tune.fac = 0.5,
  tune.cut = TRUE, ann.init = TRUE, nz.lab = NULL, permut = NULL)
```

Arguments

`fold` — A fold number used for cross validation.

`nrho` — The number of tuning parameters searched in cross validation.

`rho.seq` — A vector of tuning parameters searched in cross validation. If both `nrho` and `rho.seq` are specified, then `rho.seq` overrides `nrho`.

`y` — An n x 1 response vector.

`x` — An n x p matix of covariates, excluding a constant.

`iw` — An n x 1 weight vector.

`loss` — A loss function, which can be specified as "guas" for continuous responses, or "ml" or "cal" for binary respones.

`n.iter` — The maximum number of iterations allowed as in `glm.regu`.

`eps` — The tolerance used to declare convergence as in `glm.regu`.

`tune.fac` — The multiplier (factor) used to define `rho.seq` if only `nrho` is specified.

`tune.cut` — Logical; if `TRUE`, all smaller tuning parameters are skipped once non-convergence is found with a tuning parameter.

`ann.init` — Logical; if `TRUE`, the estimates from the previous tuning parameter are used as the inital values when fitting with the current tuning parameter.

`nz.lab` — A p x 1 logical vector (useful for simulations), indicating which covariates are included when calculating the number of nonzero coefficients.

`permut` — An n x 1 vector, giving a random permutation of the integers from 1 to n, which is used in cross validation.

Details

Cross validation is performed as described in Tan (2017, 2019). If not specified by users, the sequence of tuning parameters searched is defined as a geometric series of length `nrho`, starting from the value which yields a zero solution, and then decreasing by a factor `tune.fac` successively.

After cross validation, two tuning parameters are selected. The first and default choice is the value yielding the smallest average test loss. The second choice is the largest value giving the average test loss within one standard error of the first choice (Hastie, Tibshirani, and Friedman 2016).

Value

`permut` — An n x 1 vector, giving the random permutation used in cross validation.

`rho` — The vector of tuning parameters, searched in cross validation.

`non.conv` — A vector indicating the non-convergene status found or imputed if `tune.cut=TRUE`, for the tuning parmaters in cross validation. For each tuning parameter, 0 indicates convergence, 1 non-convergence if exceeding `n.iter`, 2 non-convergence if exceeding `bt.lim`.

`err.ave` — A vector giving the averages of the test losses in cross validation.

`err.sd` — A vector giving the standard deviations of the test losses in cross validation.

`sel.rho` — A vector of two selected tuning parameters by cross validation; see **Details**.

`sel.nz`	A vector of numbers of nonzero coefficients estimated for the selected tuning parameters.
`sel.bet`	The $(p+1)$ x 2 vector of estimated intercept and coefficients.
`sel.fit`	The n x 2 vector of fitted values.

References

Hastie, T., Tibshirani, R., and Friedman. J. (2016) *The Elements of Statistical Learning* (second edition), Springer: New York.

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

### Example 1: Regularized maximum likelihood estimation of propensity scores
ps.cv.rml <- glm.regu.cv(fold=5, nrho=1+10, y=tr, x=x, loss="ml")
ps.cv.rml$rho
ps.cv.rml$err.ave
ps.cv.rml$err.sd
ps.cv.rml$sel.rho
ps.cv.rml$sel.nz

fp.cv.rml <- ps.cv.rml $sel.fit[,1]
check.cv.rml <- mn.ipw(x, tr, fp.cv.rml)
check.cv.rml$est

### Example 2: Regularized calibrated estimation of propensity scores
ps.cv.rcal <- glm.regu.cv(fold=5, nrho=1+10, y=tr, x=x, loss="cal")
ps.cv.rcal$rho
ps.cv.rcal$err.ave
ps.cv.rcal$err.sd
ps.cv.rcal$sel.rho
ps.cv.rcal$sel.nz

fp.cv.rcal <- ps.cv.rcal $sel.fit[,1]

check.cv.rcal <- mn.ipw(x, tr, fp.cv.rcal)
check.cv.rcal$est
```

glm.regu.path *Regularied M-estimation for fitting generalized linear models along a regularization path*

Description

This function implements regularized M-estimation for fitting generalized linear models with binary or contiunous responses along a regularization path.

Usage

```
glm.regu.path(nrho = NULL, rho.seq = NULL, y, x, iw = NULL,
  loss = "cal", n.iter = 100, eps = 1e-06, tune.fac = 0.5,
  tune.cut = TRUE, ann.init = TRUE, nz.lab = NULL)
```

Arguments

`nrho`	The number of tuning parameters in a regularization path.
`rho.seq`	A vector of tuning parameters in a regularization path. If both `nrho` and `rho.seq` are specified, then `rho.seq` overrides `nrho`.
`y`	An n x 1 response vector.
`x`	An n x p matix of covariates, excluding a constant.
`iw`	An n x 1 weight vector.
`loss`	A loss function, which can be specified as "guas" for continuous responses, or "ml" or "cal" for binary respones.
`n.iter`	The maximum number of iterations allowed as in `glm.regu`.
`eps`	The tolerance used to declare convergence as in `glm.regu`.
`tune.fac`	The multiplier (factor) used to define rho.seq if only `nrho` is specified.
`tune.cut`	Logical; if `TRUE`, all smaller tuning parameters are skipped once non-convergence is found with a tuning parameter.
`ann.init`	Logical; if `TRUE`, the estimates from the previous tuning parameter are used as the inital value when fitting with the current tuning parameter.
`nz.lab`	A p x 1 logical vector (useful for simulations), indicating which covariates are included when calculating the number of nonzero coefficients.

Details

If not specified by users, the sequence of tuning parameters (i.e., the regularization path) is defined as a geometric series of length `nrho`, starting from the value which yields a zero solution, and then decreasing by a factor `tune.fac` successively.

Value

rho	The vector of tuning parameters included in the regularization path.
non.conv	A vector indicating the non-convergene status found or imputed if tune.cut=TRUE, along the regularization path. For each tuning parameter, 0 indicates convergence, 1 non-convergence if exceeding n.iter, 2 non-convergence if exceeding bt.lim.
nz.all	A vector giving the numbers of nonzero coefficients estimated, along the regularization path.
bet.all	A matrix giving estimated intercept and coefficients, column by column, along the regularization path.
fit.all	A matrix giving fitted values, column by column, along the regularization path.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

### Example 1: linear regression
out.rgaus.path <- glm.regu.path(rho.seq=c(.01, .02, .05, .1, .2, .5), y=y[tr==1], x=x[tr==1,],
                                loss="gaus")

# the estimated intercept and coefficients; the first 10 are shown
out.rgaus.path$bet.all[1:10,]

### Example 2: logistic regression using likelihood loss
out.rml.path <- glm.regu.path(rho.seq=c(.002, .005, .01, .02, .05, .1), y=tr, x=x, loss="ml")
out.rml.path$bet.all[1:10,]

### Example 3: logistic regression using calibration loss
out.rcal.path <- glm.regu.path(rho.seq=c(.005, .01, .02, .05, .1, .2), y=tr, x=x, loss="cal")
out.rcal.path$bet.all[1:10,]
```

`mn.aipw` *Augmented inverse probability weighted estimation of population means*

Description

This function implements augmented inverse probability weighted (IPW) estimation of population means with missing data, provided both fitted propensity scores and fitted values from outcome regression.

Usage

```
mn.aipw(y, tr, fp, fo, off = 0)
```

Arguments

`y`	An n x 1 vector of outcomes with missing data.
`tr`	An n x 1 vector of non-missing indicators (=1 if `y` is observed or 0 if `y` is missing).
`fp`	An n x 1 vector of fitted propensity scores.
`fo`	An n x 1 vector of fitted values from outcome regression.
`off`	An offset value (e.g., the true value in simulations) used to calculate the z-statistic.

Value

`one`	The direct IPW estimate of 1.
`ipw`	The ratio IPW estimate.
`or`	The outcome regression estimate.
`est`	The augmented IPW estimate.
`var`	The estimated variance associated with the augmented IPW estimate.
`ze`	The z-statistic for the augmented IPW estimate, compared to `off`.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

mn.ipw *Inverse probability weighted estimation of population means*

Description

This function implements inverse probability weighted (IPW) estimation of population means with missing data, provided fitted propensity scores.

Usage

```
mn.ipw(y, tr, fp)
```

Arguments

`y`	An n x 1 vector of outcomes with missing data.
`tr`	An n x 1 vector of non-missing indicators (=1 if `y` is observed or 0 if `y` is missing).
`fp`	An n x 1 vector of fitted propensity scores.

Details

The ratio IPW estimate is the direct IPW estimate divided by that with `y` replaced by a vector of 1s. The latter is referred to as the direct IPW estimate of 1.

Value

`one`	The direct IPW estimate of 1.
`est`	The ratio IPW estimate.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

mn.nreg	*Model-assisted inference for population means without regularization*

Description

This function implements model-assisted inference for population means with missing data, using non-regularized calibrated estimation.

Usage

```
mn.nreg(y, tr, x, ploss = "cal", yloss = "gaus", off = 0)
```

Arguments

y	An n x 1 vector of outcomes with missing data.
tr	An n x 1 vector of non-missing indicators (=1 if y is observed or 0 if y is missing).
x	An n x p matix of covariates (excluding a constant), used in both propensity score and outcome regression models.
ploss	A loss function used in propensity score estimation (either "ml" or "cal").
yloss	A loss function used in outcome regression (either "gaus" for continuous outcomes or "ml" for binary outcomes).
off	An offset value (e.g., the true value in simulations) used to calculate the z-statistic from augmented IPW estimation.

Details

Two steps are involved in this function: first fitting propensity score and outcome regression models and then applying the augmented IPW estimator for a population mean. For ploss="cal", calibrated estimation is performed similarly as in Tan (2017, 2019), but without regularization. The method then leads to model-assisted inference, in which confidence intervals are valid if the propensity score model is correctly specified but the outcome regression model may be misspecified. With linear outcome models, the inference is also doubly robust (Kim and Haziza 2014; Vermeulen and Vansteelandt 2015). For ploss="ml", maximum likelihood estimation is used (Robins et al. 1994). In this case, standard errors are in general conservative if the propensity score model is correctly specified but the outcome regression model may be misspecified.

Value

ps	A list containing the results from fitting the propensity score model by glm.nreg.
fp	The n x 1 vector of fitted propensity scores.
or	A list containing the results from fitting the outcome regression model by glm.nreg.
fo	The n x 1 vector of fitted values from outcome regression.
est	A list containing the results from augmented IPW estimation by mn.aipw.

References

Kim, J.K. and Haziza, D. (2014) Doubly robust inference with missing data in survey sampling, *Statistica Sinica*, 24, 375-394.

Robins, J.M., Rotnitzky, A., and Zhao, L.P. (1994) Estimation of regression coefficients when some regressors are not always observed, *Journal of the American Statistical Association*, 89, 846-866.

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Vermeulen, K. and Vansteelandt, S. (2015) Bias-reduced doubly robust estimation, *Journal of the American Statistical Association*, 110, 1024-1036.

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

# missing data
y[tr==0] <- NA

# include only 10 covariates
x2 <- x[,1:10]

mn.cal <- mn.nreg(y, tr, x2, ploss="cal", yloss="gaus")
unlist(mn.cal$est)
```

`mn.regu.cv` *Model-assisted inference for population means based on cross validation*

Description

This function implements model-assisted inference for population means with missing data, using regularized calibrated estimation based on cross validation.

Usage

```
mn.regu.cv(fold, nrho = NULL, rho.seq = NULL, y, tr, x,
  ploss = "cal", yloss = "gaus", off = 0, ...)
```

Arguments

`fold`	A vector of length 2 giving the fold numbers for cross validation in propensity score estimation and outcome regression respectively.
`nrho`	A vector of length 2 giving the numbers of tuning parameters searched in cross validation.
`rho.seq`	A list of two vectors giving the tuning parameters in propensity score estimation (first vector) and outcome regression (second vector).
`y`	An n x 1 vector of outcomes with missing data.
`tr`	An n x 1 vector of non-missing indicators (=1 if `y` is observed or 0 if `y` is missing).
`x`	An n x p matix of covariates, used in both propensity score and outcome regression models.
`ploss`	A loss function used in propensity score estimation (either "ml" or "cal").
`yloss`	A loss function used in outcome regression (either "gaus" for continuous outcomes or "ml" for binary outcomes).
`off`	An offset value (e.g., the true value in simulations) used to calculate the z-statistic from augmented IPW estimation.
`...`	Additional arguments to `glm.regu.cv`.

Details

Two steps are involved in this function: first fitting propensity score and outcome regression models and then applying the augmented IPW estimator for a population mean. For `ploss="cal"`, regularized calibrated estimation is performed with cross validation as described in Tan (2017, 2019). The method then leads to model-assisted inference, in which confidence intervals are valid with high-dimensinoal data if the propensity score model is correctly specified but the outcome regression model may be misspecified. With linear outcome models, the inference is also doubly robust. For `ploss="ml"`, regularized maximum likelihood estimation is used (Belloni et al. 2014; Farrell 2015). In this case, standard errors are only shown to be valid if both the propensity score model and the outcome regression model are correctly specified.

Value

`ps`	A list containing the results from fitting the propensity score model by `glm.regu.cv`.
`fp`	The n x 1 vector of fitted propensity scores.
`or`	A list containing the results from fitting the outcome regression model by `glm.regu.cv`.
`fo`	The n x 1 vector of fitted values from outcome regression.
`est`	A list containing the results from augmented IPW estimation by `mn.aipw`.

References

Belloni, A., Chernozhukov, V., and Hansen, C. (2014) Inference on treatment effects after selection among high-dimensional controls, *Review of Economic Studies*, 81, 608-650.

Farrell, M.H. (2015) Robust inference on average treatment effects with possibly more covariates than observations, *Journal of Econometrics*, 189, 1-23.

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

# missing data
y[tr==0] <- NA

mn.cv.rcal <- mn.regu.cv(fold=5*c(1,1), nrho=(1+10)*c(1,1), rho.seq=NULL, y, tr, x,
                         ploss="cal", yloss="gaus")
unlist(mn.cv.rcal$est)
```

---

mn.regu.path — *Model-assisted inference for population means along a regularization path*

Description

This function implements model-assisted inference for population means with missing data, using regularized calibrated estimation along a regularization path for propensity score (PS) estimation while based on cross validation for outcome regression (OR).

Usage

```
mn.regu.path(fold, nrho = NULL, rho.seq = NULL, y, tr, x,
  ploss = "cal", yloss = "gaus", off = 0, ...)
```

Arguments

fold	A vector of length 2, with the second component giving the fold number for cross validation in outcome regression. The first component is not used.
nrho	A vector of length 2 giving the number of tuning parameters in a regularization path for PS estimation and that in cross validation for OR.

`rho.seq`	A list of two vectors giving the tuning parameters for propensity score estimation (first vector) and outcome regression (second vector).
`y`	An n x 1 vector of outcomes with missing data.
`tr`	An n x 1 vector of non-missing indicators (=1 if `y` is observed or 0 if `y` is missing).
`x`	An n x p matix of covariates, used in both propensity score and outcome regression models.
`ploss`	A loss function used in propensity score estimation (either "ml" or "cal").
`yloss`	A loss function used in outcome regression (either "gaus" for continuous outcomes or "ml" for binary outcomes).
`off`	An offset value (e.g., the true value in simulations) used to calculate the z-statistic from augmented IPW estimation.
`...`	Additional arguments to `glm.regu.cv` and `glm.regu.path`.

Details

See **Details** for `mn.regu.cv`.

Value

`ps`	A list containing the results from fitting the propensity score model by `glm.regu.path`.
`fp`	The matrix of fitted propensity scores, column by column, along the PS regularization path.
`or`	A list of objects, each giving the results from fitting the outcome regression model by `glm.regu.cv` for a PS tuning parameter.
`fo`	The matrix of fitted values from outcome regression based on cross validation, column by column, along the PS regularization path.
`est`	A list containing the results from augmented IPW estimation by `mn.aipw`.
`rho`	A vector of tuning parameters leading to converged results in propensity score estimation.

References

Tan, Z. (2017) Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data, arXiv:1710.08074.

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Examples

```
data(simu.data)
n <- dim(simu.data)[1]
p <- dim(simu.data)[2]-2

y <- simu.data[,1]
```

```
tr <- simu.data[,2]
x <- simu.data[,2+1:p]
x <- scale(x)

# missing data
y[tr==0] <- NA

mn.path.rcal <- mn.regu.path(fold=5*c(0,1), nrho=(1+10)*c(1,1), y=y, tr=tr, x=x,
                             ploss="cal", yloss="gaus")
mn.path.rcal$est
```

---

simu.data *Simulated data*

Description

A dataset simulated as in Tan (2019), Section 4.

Usage

```
data(simu.data)
```

Format

A data matrix with 800 rows and 202 columns.

Details

The dataset is generated as follows, where `y`, `tr`, and `x` represent an outcome, a treatment, and covariates respectively.

```
library(MASS)

###
mt0 <- 1-pnorm(-1)
mt1 <- dnorm(-1)
mt2 <- -(2*pnorm(-1)-1)/2 - dnorm(-1) +1/2
mt3 <- 3*dnorm(-1)
mt4 <- -3/2*(2*pnorm(-1)-1) - 4*dnorm(-1) +3/2

m.z1 <- mt0 + 2*mt1 + mt2
v.z1 <- mt0 + 4*mt1 + 6*mt2 + 4*mt3 + mt4
v.z1 <- v.z1 + 1 + 2*(mt1 + 2*mt2 + mt3)

sd.z1 <- sqrt(v.z1 -m.z1^2)
###
```

```
set.seed(123)

n <- 800
p <- 200

noise <- rnorm(n)

covm <- matrix(1,p,p)
for (i1 in 1:p)
  for (i2 in 1:p) {
    covm[i1,i2] <- 2^(-abs(i1-i2))
  }
x <- mvrnorm(n, mu=rep(0,p), Sigma=covm)

# transformation
z <- x
for (i in 1:4) {
  z[,i] <- ifelse(x[,i]>-1,x[,i]+(x[,i]+1)^2,x[,i])
  z[,i] <- (z[,i]-m.z1) /sd.z1   # standardized
}

# treatment
eta <- 1+ c( z[,1:4] %*% c(1, .5, .25, .125) )
tr <- rbinom(n, size=1, prob=expit(eta))

# outcome
eta.y <- c( z[,1:4] %*% c(1, .5, .25, .125) )
y <- eta.y + noise

# save
simu.data <- cbind(y, tr, x)
save(simu.data, file="simu.data.rda")
```

References

Tan, Z. (2019) Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data, *Annals of Statistics*, to appear (preprint arXiv:1801.09817).

Index