Package ‘tmle’

October 15, 2019

Version 1.4.0.1
Date 2019-10-15
Title Targeted Maximum Likelihood Estimation
Author Susan Gruber [aut, cre],
Mark van der Laan [aut],
Chris Kennedy [ctr]
Maintainer Susan Gruber <sgruber@cal.berkeley.edu>
Copyright Copyright 2012. The Regents of the University of California
(Regents). All Rights Reserved.
Depends glmnet, SuperLearner (>= 2.0)
Suggests dbarts (>= 0.9-12), gam (>= 1.15)
Description Targeted maximum likelihood estimation of point treatment effects (Targeted Maximum Likelihood Learning, The International Journal of Biostatistics, 2(1), 2006. This version automatically estimates the additive treatment effect among the treated (ATT) and among the controls (ATC). The tmle() function calculates the adjusted marginal difference in mean outcome associated with a binary point treatment, for continuous or binary outcomes. Relative risk and odds ratio estimates are also reported for binary outcomes. Missingness in the outcome is allowed, but not in treatment assignment or baseline covariate values. The population mean is calculated when there is missingness, and no variation in the treatment assignment. The tmleMSM() function estimates the parameters of a marginal structural model for a binary point treatment effect. Effect estimation stratified by a binary mediating variable is also available. An ID argument can be used to identify repeated measures. Default settings call 'SuperLearner' to estimate the Q and g portions of the likelihood, unless values or a user-supplied regression function are passed in as arguments.
License BSD_3_clause + file LICENSE | GPL-2
URL https://CRAN.R-project.org/package=tmle
NeedsCompilation no
Repository CRAN
Date/Publication 2019-10-15 15:40:02 UTC
R topics documented:

tmle-package .......................................................... 2
calcParameters ......................................................... 3
calcSigma ............................................................... 4
estimateG ............................................................... 6
estimateQ ............................................................... 7
fev ................................................................. 8
oneStepATT ........................................................... 9
summary.tmle ......................................................... 10
summary.tmleMSM .................................................. 12
tmle .............................................................. 13
tmle.SL.dbarts2 .................................................... 18
tmleMSM ............................................................ 20
tmleNews ............................................................ 24

Index 25

Package: tmle
Type: Package
Version: 1.4.0.1
Date: 2019-10-15
License: BSD | GPL-2

Description

Targeted maximum likelihood estimation of marginal treatment effect of a binary point treatment on a continuous or binary outcome, adjusting for baseline covariates (ATE: entire population, ATT: treated population, ATC: control population). Missingness in the outcome is accounted for in the estimation procedure. The population mean outcome is calculated when there is missingness and no treatment. Controlled direct effect estimation is available, and MSM parameter estimation for binary point treatment effects. Optional data-adaptive estimation of $Q$ and $g$ portions of the likelihood using the SuperLearner package is strongly encouraged.

Details

Author(s)

Susan Gruber, in collaboration with Mark van der Laan.

Maintainer: Susan Gruber, <sgruber@cal.berkeley.edu>
References


See Also

tmle, tmleMSM

calcParameters  Calculate Parameter Estimates (calcParameters)

Description

An internal function called by the tmle function to calculate the population mean effect when there is missingness in the data, but no treatment assignment. When observations are in treatment and control groups, estimates the additive treatment effect among the entire population (ATE), among the treated (ATT), and among the controls (ATC). If the outcome is binary, also the relative risk and odds ratio parameters. P-values and 95% confidence intervals are also calculated (on the log scale for RR and OR).

Usage

calcParameters(Y, A, I.Z, Delta, g1W, g0W, Q, mu1, mu0, id, family)
Arguments

- **Y**: continuous or binary outcome variable
- **A**: binary treatment indicator, 1 - treatment, 0 - control
- **I.Z**: indicator \( Z=\zeta \), needed for CDE estimation
- **Delta**: indicator of missing outcome. 1 - observed, 0 - missing
- **g1W**: censoring mechanism estimates, \( P(A = 1|W) \times P(Delta = 1|A, W) \)
- **g0W**: censoring mechanism estimates, \( P(A = 0|W) \times P(Delta = 1|A, W) \)
- **Q**: a 3-column matrix \( Q(A, W), Q(1, W), Q(0, W) \)
- **mu1**: targeted estimate of \( E(Y|A = 1, W) \)
- **mu0**: targeted estimate of \( E(Y|A = 0, W) \)
- **id**: subject identifier
- **family**: family specification for regressions, generally ‘gaussian’ for continuous outcomes, ‘binomial’ for binary outcomes

Value

- **EY1**: Population mean outcome estimate, variance, p-value, 95% confidence interval (missingness only, no treatment assignment), or NULL
- **ATE**: additive treatment effect estimate, variance, p-value, 95% confidence interval, or NULL
- **RR**: relative risk estimate, p-value, 95% confidence interval, log(RR), variance(log(RR)), or NULL
- **OR**: odds ratio estimate, p-value, 95% confidence interval, log(OR), variance(log(OR)), or NULL

Author(s)

Susan Gruber

See Also

tmle, estimateQ, estimateG, tmleMSM, calcSigma

calcSigma: Calculate Variance-Covariance Matrix for MSM Parameters (calcSigma)

Description

An internal function called by the tmleMSM function to calculate the variance-covariance matrix of the parameter estimates based on the influence curve of the specified MSM.
Usage

calcSigma(hAV, gAVW, Y, Q, mAV, covar.MSM, covar.MSMA0, covar.MSMA1, I.V, Delta, ub, id, family)

Arguments

- **hAV**: values used in numerator of weights applied to the estimation procedure
- **gAVW**: \( p(A = a|V, W, T) \ast p(Delta = 1|A, V, W, T) \)
- **Y**: continuous or binary outcome variable
- **Q**: estimated \( P(Y|A, V, W, T, Delta = 1) \), typically targeted values \( Q^* \) are passed in
- **mAV**: predicted values for \( EY_1 \) from the MSM using the targeted estimates for \( psi \)
- **covar.MSM**: covariate values used as predictors for the MSM when \( A=a \)
- **covar.MSMA0**: covariate values used as predictors for the MSM when \( A=0 \)
- **covar.MSMA1**: covariate values used as predictors for the MSM when \( A=1 \)
- **I.V**: indicator that observation is in stratum of interest
- **Delta**: indicator of missing outcome. 1 - observed, 0 - missing
- **ub**: upper bound on weights
- **id**: subject identifier
- **family**: ‘gaussian’ for continuous outcomes, ‘binomial’ for binary outcomes

Value


Author(s)

Susan Gruber

See Also

tmle, estimateQ, estimateG, tmleMSM
**Description**

An internal function called by the `tmle` function to obtain an estimate of conditional treatment assignment probabilities \( P(A = 1|W) \), and conditional probabilities for missingness, \( P(Delta = 1|A, W) \). The estimate can be based on user-supplied values, a user-supplied regression formula, or a data-adaptive super learner fit. If the `SuperLearner` package is not available, and there are no user-specifications, estimation is carried out using main terms regression with `glm`. These main terms-based estimates may yield poor results.

**Usage**

```r
estimateG(d, g1W, gform, SL.library, id, V, verbose, message, outcome, newdata=d, discreteSL)
```

**Arguments**

- `d`: dataframe with binary dependent variable in the first column, predictors in remaining columns
- `g1W`: vector of values for \( P(A = 1|W) \), \( P(Z = 1|A, W) \), or \( P(Delta = 1|Z, A, W) \)
- `gform`: regression formula of the form \( A \sim W \), (dependent variable is one of \( A, Z, D \)) if specified this overrides the call to `SuperLearner`
- `SL.library`: vector of prediction algorithms used by `SuperLearner`, default value is (`SL.glm`, `tmle.SL.dbarts.k.5`, `SL.gam`)
- `id`: subject identifier
- `V`: Number of cross validation folds for Super Learning
- `verbose`: status messages printed if set to TRUE
- `message`: text specifies whether treatment or missingness mechanism is being estimated
- `outcome`: \( A, D, Z \) to indicate which quantity is being estimated.
- `newdata`: optional dataset to be used for prediction after fitting on `d`.
- `discreteSL`: If true, returns discrete SL estimates, otherwise ensemble estimates. Ignored when SL is not used.

**Value**

- `g1W`: a vector containing values for \( P(A = 1|W) \), matrix for \( P(Z = 1|A, W) \), evaluated at \( A=0, A=1 \), or matrix \( P(Delta = 1|Z, A, W) \) evaluated at \((0,0), (0,1), (1,0), (1,1)\)
- `coef`: coefficients for each term in the working model used for estimation if `glm` was used
- `type`: estimation procedure
estimateQ

Author(s)
Susan Gruber

See Also
tmle, estimateQ, calcParameters, tmleMSM, calcSigma

Initial Estimation of Q portion of the Likelihood

Description
An internal function called by the tmle function to obtain an initial estimate of the Q portion of the likelihood based on user-supplied matrix values for predicted values of (counterfactual outcomes) Q(0,W), Q(1,W), or a user-supplied regression formula, or based on a data-adaptively selected SuperLearner fit. In the absence of user-supplied values, a user-supplied regression formula takes precedence over data-adaptive super-learning. The default is to return cross-validated predictions.

Usage
estimateQ(Y, Z, A, W, Delta, Q, Qbounds, Qform, maptoYstar, SL.library, cvQinit, family, id, V, verbose, discreteSL)

Arguments

Y continuous or binary outcome variable
Z optional binary indicator for intermediate covariate for controlled direct effect estimation
A binary treatment indicator, 1 - treatment, 0 - control
W vector, matrix, or dataframe containing baseline covariates
Delta indicator of missing outcome. 1 - observed, 0 - missing
Q 3-column matrix (Q(A,W), Q(0,W), Q(1,W))
Qbounds Bounds on predicted values for Q, set to alpha for logistic fluctuation, or range(Y) if not user-supplied
Qform regression formula of the form Y~A+W
maptoYstar if TRUE indicates continuous Y values should be shifted and scaled to fall between (0,1)
SL.library specification of prediction algorithms, default is (‘SL.glm’, ‘SL.glmnet’, ‘tmle.SL.dbarts2’). In practice, including more prediction algorithms in the library improves results.
cvQinit logical, whether or not to estimate cross-validated values for initial Q, default=TRUE
family family specification for regressions, generally ‘gaussian’ for continuous outcomes, ‘binomial’ for binary outcomes
id
subject identifier

V
Number of cross-validation folds for Super Learning

verbose
status message printed if set to TRUE

discreteSL
If true, returns discrete SL estimates, otherwise ensemble estimates. Ignored when SL is not used.

Value

Q
nx3 matrix, columns contain the initial estimate of \[ Q(A, W) = E(Y | A = a, W), Q(0, W) = E(Y | A = 0, W), Q(1, W) = E(Y | A = 1, W) \]. For controlled direct estimation, nx5 matrix, \[ E(Y | Z, A, W) \], evaluated at \((z, a), (0, 0), (0, 1), (1, 0), (1, 1)\) on scale of linear predictors

Qfamily
‘binomial’ for targeting with logistic fluctuation, ‘gaussian’ for linear fluctuation

coef
coefficients for each term in working model used for initial estimation of Q if glm used.

type
type of estimation procedure

Author(s)
Susan Gruber

See Also
tmle, estimateG, calcParameters, tmleMSM, calcSigma

---

**fev**

*Forced Expiratory Volume (FEV) Data (fev)*

Description

Sample of 654 youths, aged 3 to 19, in the area of East Boston during middle to late 1970’s. Interest concerns the relationship between smoking and FEV. Since the study is necessarily observational, statistical adjustment via regression models clarifies the relationship.

Usage
data(fev)

Format

A data frame with 654 observations on the following 5 variables.

- age  a numeric vector
- fev  a numeric vector
- ht   a numeric vector
- sex  a numeric vector
- smoke a numeric vector
Calculate Additive treatment effect among the treated (oneStepATT)

Description
An internal function called by the tmle function to calculate the additive treatment effect among the treated (ATT) using a universal least favorable submodel (on the transformed scale if outcomes are continuous). The function is called a second time with updated arguments to calculate the additive treatment effect among the controls (ATC). Missingness in the outcome data is allowed.

Usage
oneStepATT(Y, A, Delta, Q, g1W, pDelta1, depsilon, max_iter, gbounds, Qbounds)

Arguments
- **Y**: continuous or binary outcome variable
- **A**: binary treatment indicator, 1 - treatment, 0 - control
- **Delta**: indicator of missing outcome. 1 - observed, 0 - missing
- **Q**: a 3-column matrix \((Q(A,W), Q(1,W), Q(0,W))\)
- **g1W**: treatment mechanism estimates, \(P(A = 1|W)\)
- **pDelta1**: censoring mechanism estimates, a 2-column matrix \([P(Delta = 1|A = 0, W), P(Delta = 1|A = 1, W)]\)
- **depsilon**: step size for delta moves, set to 0.001
- **max_iter**: maximum number of iterations before terminating without convergence
- **gbounds**: bounds on the propensity score for untreated subjects
- **Qbounds**: alpha bounds on the logit scale

Value
- **psi**: effect estimate (on the transformed scale for continuous outcomes)
- **IC**: influence function
- **conv**: TRUE if procedure converged, FALSE otherwise

Author(s)
Susan Gruber

See Also
tmle,
Summary of the Results of a Call to the tmle Routine

Description

These functions are all methods for class tmle, tmle.list, summary.tmle, summary.tmle.list objects.

Usage

```r
## S3 method for class 'tmle'
summary(object, ...)
## S3 method for class 'tmle.list'
summary(object, ...)
## S3 method for class 'tmle'
print(x, ...)
## S3 method for class 'tmle.list'
print(x, ...)
## S3 method for class 'summary.tmle'
print(x, ...)
## S3 method for class 'summary.tmle.list'
print(x, ...)
```

Arguments

- `object` - an object of class tmle or tmle.list.
- `x` - an object of class tmle or tmle.list for summary functions, class summary.tmle or summary.tmle.list for print functions.
- `...` - currently ignored.

Details

`print.tmle` prints the estimate, variance, p-value, and 95% confidence interval only. `print.summary.tmle`, called indirectly by entering the command `summary(result)` (where result has class tmle), outputs additional information. Controlled direct effect estimates have class tmle.list, a list of two objects of class tmle. The first item corresponds to $Z = 0$, the second to $Z = 1$.

Value

- `estimates` - list of parameter estimates, p-values, and 95% confidence intervals
- `Qmodel` - working model used to obtain initial estimate of $Q$ portion of the likelihood, if glm used
- `Qterms` - terms in the model for $Q$
- `Qcoef` - coefficient of each term in model for $Q$
- `gmodel` - model used to estimate treatment mechanism $g$
summary.tmle

gterms terms in the treatment mechanism model
gcoef coefficient of each term in model for treatment mechanism
gtype description of estimation procedure for treatment mechanism, e.g. "SuperLearner"
gdiscreteSL flag indicating whether discrete SL or ensemble SL was used for treatment mechanism estimation
g.Zmodel model used to estimate intermediate variable assignment mechanism g.Z
g.Zterms terms in the intermediate mechanism model
g.Zcoef coefficient of each term in model for intermediate mechanism
g.Ztype description of estimation procedure for intermediate variable
g.ZdiscreteSL flag indicating whether discrete SL or ensemble SL was used for intermediate variable estimation
g.Deltamodel model used to estimate missingness mechanism g.Delta
g.Deltaterms terms in the missingness mechanism model
g.Deltacoef coefficient of each term in model for missingness mechanism
g.Deltatype description of estimation procedure for missingness
g.DeltadiscreteSL flag indicating whether discrete SL or ensemble SL was used for missingness estimation

Author(s)

Susan Gruber

See Also

tmle

Examples

# generate data
set.seed(10)
n <- 500
W <- matrix(rnorm(n*3), ncol=3)
A <- rbinom(n,1, 1/(1+exp(-(.1*W[,1] - .1*W[,2] + .5*W[,3]))))
colnames(W) <- paste("W",1:3, sep="")

result <- tmle(Y,A,W, Qform="Y~A+W1", g1W=rep(.5, n))
summary(result)
summary.tmleMSM

Summarization of the results of a call to the tmleMSM function

Description

These functions are all methods for class tmleMSM, summary.tmleMSM objects.

Usage

## S3 method for class 'tmleMSM'
summary(object, ...)
## S3 method for class 'tmleMSM'
print(x, ...)
## S3 method for class 'summary.tmleMSM'
print(x, ...)

Arguments

object an object of class tmleMSM.
x an object of class tmleMSM for summary functions, class summary.tmleMSM for print functions.
... currently ignored.

Details

print.tmleMSM prints the estimate, standard error, p-value, and 95% confidence interval only. print.summary.tmleMSM, called indirectly by entering the command summary(result) (where result has class tmleMSM), outputs additional information.

Value

estimates matrix of MSM parameter estimates, standard errors, pvalues, upper and lower bounds on 95% confidence intervals
sigma variance-covariance matrix
Qmodel working model used to obtain initial estimate of Q portion of the likelihood, if glm used
Qterms terms in the model for Q
Qcoef coefficient of each term in model for Q
gmodel model used to estimate treatment mechanism g
gterms terms in the treatment mechanism model
gcoef coefficient of each term in model for treatment mechanism
gtype description of estimation procedure for treatment mechanism, e.g. "SuperLearner"
g.AVmodel model used to estimate h(A,V) (or h(A,T))
g.AVterms terms in the model for h(A,V)
Description

Targeted maximum likelihood estimation of parameters of a marginal structural model, and of marginal treatment effects of a binary point treatment on an outcome. In addition to the additive treatment effect, risk ratio and odds ratio estimates are reported for binary outcomes. The tmle function is generally called with arguments \((Y, A, W)\), where \(Y\) is a continuous or binary outcome variable, \(A\) is a binary treatment variable, \((A=1 \text{ for treatment}, A=0 \text{ for control})\), and \(W\) is a matrix or dataframe of baseline covariates. The population mean outcome is calculated when there is no variation in \(A\). If values of binary mediating variable \(Z\) are supplied, estimates are returned at each level of \(Z\). Missingness in the outcome is accounted for in the estimation procedure if missingness indicator \(\Delta\) is 0 for some observations. Repeated measures can be identified using the \(id\) argument.

Usage

```r
tmle(Y, A, W, Z=NULL, Delta = rep(1,length(Y)), Q = NULL, Q.Z1 = NULL, Qform = NULL, Qbounds = NULL, Q.SL.library = c("SL.glm", "tmle.SL.dbarts2", "SL.glmnet"), cvQinit = TRUE, g1W = NULL, gform = NULL, gbound = 0.025, pZ1=NULL, g.Zform = NULL, pDelta1 = NULL, g.Deltaform = NULL, g.SL.library = c("SL.glm", "tmle.SL.dbarts.k.5", "SL.gam"), family = "gaussian", fluctuation = "logistic", alpha = 0.9995, id=1:length(Y), V = 5, verbose = FALSE, Q.discreteSL=FALSE, g.discreteSL=FALSE, prescreenW.g=TRUE, min.retain = 2, target.gwt = TRUE, automate=FALSE)
```
Arguments

Y  continuous or binary outcome variable
A  binary treatment indicator, 1 - treatment, 0 - control
W  vector, matrix, or dataframe containing baseline covariates
Z  optional binary indicator for intermediate covariate for controlled direct effect estimation
Delta  indicator of missing outcome or treatment assignment. 1 - observed, 0 - missing
Q  optional $nx2$ matrix of initial values for $Q$ portion of the likelihood, $(E(Y|A = 0, W), E(Y|A = 1, W))$
Q.Z1  optional $nx2$ matrix of initial values for $Q$ portion of the likelihood, $(E(Y|Z = 1, A = 0, W), E(Y|Z = 1, A = 1, W))$. (When specified, values for $E(Y|Z = 0, A = 0, W), E(Y|Z = 0, A = 1, W)$ are passed in using the $Q$ argument
Qform  optional regression formula for estimation of $E(Y|A, W)$, suitable for call to glm
Q.bounds  vector of upper and lower bounds on $Y$ and predicted values for initial $Q$. Defaults to the range of $Y$, widened by 10% of the min and max values.
Q.SL.library  optional vector of prediction algorithms to use for SuperLearner estimation of initial $Q$
Qinit  logical, if TRUE, estimates cross-validated predicted values, default=TRUE
g1W  optional vector of conditional treatment assignment probabilities, $P(A = 1|W)$
gform  optional regression formula of the form $A~W$, if specified this overrides the call to SuperLearner
gbound  value between (0,1) for truncation of predicted probabilities. See Details section for more information
pZ1  optional $nx2$ matrix of conditional probabilities $P(Z = 1|A = 0, W), P(Z = 1|A = 1, W)$
g.Zform  optional regression formula of the form $Z~A+W$, if specified this overrides the call to SuperLearner
pDelta1  optional matrix of conditional probabilities for missingness mechanism, $nx2$ when $Z$ is NULL $P(Delta = 1|A = 0, W), P(Delta = 1|A = 1, W)$. $nx4$ otherwise, $P(Delta = 1|Z = 0, A = 0, W), P(Delta = 1|Z = 0, A = 1, W), P(Delta = 1|Z = 1, A = 0, W), P(Delta = 1|Z = 1, A = 1, W)$
g.Deltaform  optional regression formula of the form $Delta~A+W$, if specified this overrides the call to SuperLearner
g.SL.library  optional vector of prediction algorithms to use for SuperLearner estimation of $g1W$ or $pDelta1$
fFamily  family specification for working regression models, generally ‘gaussian’ for continuous outcomes (default), ‘binomial’ for binary outcomes
fluctuation  ‘logistic’ (default), or ‘linear’
alpha  used to keep predicted initial values bounded away from (0,1) for logistic fluctuation
id

V

verbose

Q.discreteSL

g.discreteSL

prescreenW.g

min.retain

target.gwt

automate

details

value

estimates

---

**id**

Optional subject identifier

**V**

Number of cross-validation folds for estimating Q, and for super learner estimation of g

**verbose**

Status messages printed if set to TRUE (default=FALSE)

**Q.discreteSL**

If TRUE, discreteSL is used instead of ensemble SL. Ignored when SL not used to estimate Q

**g.discreteSL**

If TRUE, discreteSL is used instead of ensemble SL. Ignored when SL not used to estimate Q

**prescreenW.g**

Screen covariates before estimating g in order to retain only those associated with Stage 1 residuals

**min.retain**

Minimum number of covariates to retain when prescreening covariates for g. Ignored when prescreenW.g=FALSE

**target.gwt**

When TRUE, move g from denominator of clever covariate to the weight when fitting epsilon

**automate**

When TRUE, all tuning parameters are set to their default values. Number of cross validation folds and truncation level for g are set data-adaptively based on sample size (see details).

---

**Details**

**gbounds**

Defaults to (0.025, 0.975) for treatment effect estimates. If only one value is provided, symmetric truncation levels are assumed. Bounds default to (0.025, 1) for estimating the population mean outcome.

**W**

Should only contain covariates that are factors when Super Learner is not used to estimate Q or g.

Controlled direct effects are estimated when binary covariate Z is non-null. The tmle function returns an object of class tmle.list, a list of two items of class tmle. The first corresponds to estimates obtained when Z is fixed at 0, the second corresponds to estimates obtained when Z is fixed at 1.

When automate = TRUE the sample size determines the number of cross validation folds, V, and the bound on g as follows: When \( n \leq 100 \), \( V=20 \) and \( gbd = 0.1 \); When \( 100 < n \leq 500 \), \( V=10 \), \( gbd = 0.05 \); When \( 500 < n \leq 1000 \), \( V=5 \), \( gbd = 0.025 \); When \( 1000 < n \leq 10000 \), \( V=3 \), \( gbd = 0.025 \); When \( n > 10000 \), \( V=2 \), \( gbd = 0.01 \).

**Value**

**estimates**

List with elements EY1 (population mean), ATE (additive treatment effect), ATT (additive treatment effect among the treated), ATC (additive treatment effect among the controls), RR (relative risk), OR (odds ratio). Each element in the estimates of these is itself a list containing

- psi - Parameter estimate
- pvalue - Two-sided p-value
- CI - 95% confidence interval
- var.psi - Influence-curve based variance of estimate (ATE parameter only)
- log.psi - Parameter estimate on log scale (RR and OR parameters)
• var.log.psi - Influence-curve based variance of estimate on log scale (RR and OR parameters)

Qinit

initial estimate of Q. Qinit$coef are the coefficients for a glm model for Q, if applicable. Qinit$Q is an nx2 matrix, where n is the number of observations. Columns contain predicted values for Q(0,W),Q(1,W) using the initial fit. Qinit$type is method for estimating Q

Qstar

targeted estimate of Q, an nx2 matrix with predicted values for Q(0,W),Q(1,W) using the updated fit

g
treatment mechanism estimate. A list with four items: g$g1W contains estimates of P(A=1|W) for each observation, g$coef the coefficients for the model for g when glm used, g$type estimation procedure, g$discreteSL flag

g.Z

intermediate covariate assignment estimate (when applicable). A list with four items: g.Z$g1W an nx2 matrix containing values of P(Z=1|A=1,W), P(Z=1|A=0,W) for each observation, g.Z$coef the coefficients for the model for g when glm used, g.Z$type estimation procedure, g.Z$discreteSL flag

g.Delta

missingness mechanism estimate. A list with four items: g.Delta$g1W an nx4 matrix containing values of P(Delta=1|Z,A,W) for each observation, with (Z=0,A=0), (Z=0,A=1), (Z=1,A=0),(Z=1,A=1). (When Z is NULL, columns 3 and 4 are duplicates of 1 and 2.) g.Delta$coef the coefficients for the model for g when glm used, g.Delta$type estimation procedure, g.Delta$discreteSL flag

gbound

bounds used to truncate g

W.retain

names of covariates used to model the components of g

Author(s)

Susan Gruber <sgruber@cal.berkeley.edu>, in collaboration with Mark van der Laan.

References


**See Also**

`summary.tmle, estimateQ, estimateG, calcParameters, oneStepATT, tmleMSM, calcSigma`

**Examples**

```r
library(tmle)
set.seed(1)

n <- 250
W <- matrix(rnorm(n*3), ncol=3)
A <- rbinom(n, 1, 1/(1+exp(-(.2*W[,1] - .1*W[,2] + .4*W[,3])))))

# Example 1. Simplest function invocation
# SuperLearner called to estimate Q, g
# Delta defaults to 1 for all observations
## Not run:
result1 <- tmle(Y,A,W)
summary(result1)
## End(Not run)

# Example 2:
# User-supplied regression formulas to estimate Q and g
# binary outcome
n <- 250
W <- matrix(rnorm(n*3), ncol=3)
colnames(W) <- paste("W",1:3, sep="")
A <- rbinom(n,1, plogis(0.6*W[,1] +0.4*W[,2] + 0.5*W[,3]))
Y <- rbinom(n,1, plogis(A + 0.2*W[,1] + 0.1*W[,2] + 0.2*W[,3]^2 ))
result2 <- tmle(Y,A,W, family="binomial", Qform=Y~A+W1+W2+W3, gform=A~W1+W2+W3)
summary(result2)

## Not run:
# Example 3: Population mean outcome
# User-supplied (misspecified) model for Q,
# Super learner called to estimate g, g.Delta
# V set to 2 for demo, not recommended at this sample size
# approx. 20
Y <- W[,1] + W[,2]^2 + rnorm(n)
Delta <- rbinom(n, 1, 1/(1+exp(-(1.7-1*W[,1])))))
result3 <- tmle(Y,A=NULL,W, Delta=Delta, Qform="Y~A+W1+W2+W3", V=2)
print(result3)

# Example 4: Controlled direct effect
# User-supplied models for g, g.Z
# V set to 2 for demo, not recommended at this sample size
A <- rbinom(n,1,.5)
Z <- rbinom(n, 1, plogis(.5*A + .1*W[,1]))
```
\[ Y \leftarrow 1 + A + 10 \times Z + W[,1] + \text{rnorm(n)} \]

\[ \text{CDE} \leftarrow \text{tmle}(Y, A, W, Z, \text{gform}="A\sim1", \text{g.Zform} = "Z \sim A + W1", \text{V}=2) \]

\[ \text{print(CDE)} \]

\[ \text{total.effect} \leftarrow \text{tmle}(Y, A, W, \text{gform}="A\sim1") \]

\[ \text{print(total.effect)} \]

## End(Not run)

---

tmle.SL.dbarts2

Super Learner wrappers for modeling and prediction using \textit{bart} in the dbarts package

---

**Description**

These functions are used internally, not typically called by the user

**Usage**

\[
\text{tmle.SL.dbarts2}(Y, X, newX, \text{family}, \text{obsWeights}, \text{id}, \text{sigest} = \text{NA}, \text{sigdf} = 3, \text{sigquant} = 0.90, k = 2, \text{power} = 2.0, \text{base} = 0.95, \text{binaryOffset} = 0.0, \text{ntree} = 200, \text{ndpost} = 1000, \text{nskip} = 100, \text{printevery} = 100, \text{keepevery} = 1, \text{keeptrainfits} = \text{TRUE}, \text{usequants} = \text{FALSE}, \text{numcut} = 100, \text{printcutoffs} = 0, \text{nthread} = 1, \text{keepcall} = \text{TRUE,verbose} = \text{FALSE}, \ldots) \
\]

\[
\text{tmle.SL.dbarts.k.5}(Y, X, newX, \text{family}, \text{obsWeights}, \text{id}, \text{sigest} = \text{NA}, \text{sigdf} = 3, \text{sigquant} = 0.90, k = 0.5, \text{power} = 2.0, \text{base} = 0.95, \text{binaryOffset} = 0.0, \text{ntree} = 200, \text{ndpost} = 1000, \text{nskip} = 100, \text{printevery} = 100, \text{keepevery} = 1, \text{keeptrainfits} = \text{TRUE}, \text{usequants} = \text{FALSE}, \text{numcut} = 100, \text{printcutoffs} = 0, \text{nthread} = 1, \text{keepcall} = \text{TRUE,verbose} = \text{FALSE}, \ldots) \
\]

## S3 method for class 'tmle.SL.dbarts2'

\[ \text{predict(object, newdata, \text{family}, \ldots)} \]

**Arguments**

- **Y** Dependent variable
- **X** Predictor covariate matrix or data frame used as training set
- **newX** Predictor covariate matrix or data frame for which predictions should be made
- **family** Regression family, ‘gaussian’ or ‘binomial’
- **obsWeights** observation-level weights
- **id** idid to group observations, not used
- **sigest** An estimate of error variance. See \textit{bart} documentation
- **sigdf** Degrees of freedom for error variance prior. See \textit{bart} documentation
- **sigquant** Quantile of error variance prior. See \textit{bart} documentation
- **k** Tuning parameter that controls smoothing. Larger values are more conservative, see Details
power  Power parameter for tree prior
base   Base parameter for tree prior
binaryOffset Allows fits with probabilities shrunk towards values other than 0.5. See bart documentation
ntree  Number of trees in the sum-of-trees formulation
ndpost Number of posterior draws after burn in
nskip  Number of MCMC iterations treated as burn in
printevery How often to print messages
keepevery Every keepevery draw is kept to be returned to the user
keeptrainfits If TRUE the draws of \( f(x) \) for \( x \) corresponding to the rows of \( x.train \) are returned
usequants Controls how tree decisions rules are determined. See bart documentation
numcut Maximum number of possible values used in decision rules
printcutoffs Number of cutoff rules to print to screen. 0 prints nothing
nthread Integer specifying how many threads to use
keepcall Returns the call to BART when TRUE
verbose Ignored for now
... Additional arguments passed on to plot or control functions
object object of type tmle.SL.dbarts2
newdata matrix or dataframe used to get predictions from the fitted model

Details
tmle.SL.dbarts2 is in the default library for estimating \( Q \). It uses the default setting in the dbarts package, \( k = 2 \). tmle.SL.dbarts.k.5 is used to estimate the components of \( g \). It sets \( k = 0.5 \), to avoid shrinking predicted values too far from \((0, 1)\). See bart documentation for more information.

Value

objectan object of type tmle.SL.dbarts2 used internally by Super Learner

Author(s)
Chris Kennedy and Susan Gruber

See Also

SuperLearner
Targeted Maximum Likelihood Estimation of Parameter of MSM

Description

Targeted maximum likelihood estimation of the parameter of a marginal structural model (MSM) for binary point treatment effects. The `tmleMSM` function is minimally called with arguments `(Y, A, W, MSM)`, where `Y` is a continuous or binary outcome variable, `A` is a binary treatment variable, (`A=1` for treatment, `A=0` for control), and `W` is a matrix or dataframe of baseline covariates. MSM is a valid regression formula for regressing `Y` on any combination of `A, V, W, T`, where `V` defines strata and `T` represents the time at which repeated measures on subjects are made. Missingness in the outcome is accounted for in the estimation procedure if missingness indicator `Delta` is 0 for some observations. Repeated measures can be identified using the `id` argument.

Usage

```r
tmleMSM(Y, A, W, V, T = rep(1, length(Y)), Delta = rep(1, length(Y)), MSM, v = NULL, Q = NULL, Qform = NULL, Qbounds = c(-Inf, Inf), Q.SL.library = c("SL.glm", "tmle.SL.dbarts", "SL.glmnet"), cvQinit = TRUE, hAV = NULL, hAVform = NULL, g1W = NULL, gform = NULL, pDelta1 = NULL, g.Deltaform = NULL, g.SL.library = c("SL.glm", "tmle.SL.dbarts.k.5", "SL.gam"), ub = 1/0.025, family = "gaussian", fluctuation = "logistic", alpha = 0.995, id = 1:length(Y), V_SL = 5, inference = TRUE, verbose = FALSE, Q.discreteSL = FALSE, g.discreteSL = FALSE)
```

Arguments

- **Y**: continuous or binary outcome variable
- **A**: binary treatment indicator, 1 - treatment, 0 - control
- **W**: vector, matrix, or dataframe containing baseline covariates. Factors are not currently allowed.
- **V**: vector, matrix, or dataframe of covariates used to define strata
- **T**: optional time for repeated measures data
- **Delta**: indicator of missing outcome or treatment assignment. 1 - observed, 0 - missing
- **MSM**: MSM of interest, specified as valid right hand side of a regression formula (see examples)
- **v**: optional value defining the strata of interest (`V = v`) for stratified estimation of MSM parameter
- **Q**: optional `n x 2` matrix of initial values for `Q` portion of the likelihood, \( (E(Y|A = 0, W), E(Y|A = 1, W)) \)
- **Qform**: optional regression formula for estimation of `E(Y|A, W)`, suitable for call to `glm`
- **Qbounds**: vector of upper and lower bounds on `Y` and predicted values for initial `Q`
Q.SL.library
optional vector of prediction algorithms to use for SuperLearner estimation of initial Q

cvQinit
logical, if TRUE, estimates cross-validated predicted values using discrete super learning, default=TRUE

hAV
optional $nx2$ matrix used in numerator of weights for updating covariate and the influence curve. If unspecified, defaults to conditional probabilities $P(A = 1|V)$ or $P(A = 1|T)$, for repeated measures data. For unstabilized weights, pass in an $nx2$ matrix of all 1s

hAVform
optional regression formula of the form $A\sim V+T$, if specified this overrides the call to SuperLearner

g1W
optional vector of conditional treatment assignment probabilities, $P(A = 1|W)$

gform
optional regression formula of the form $A\sim W$, if specified this overrides the call to SuperLearner

pDelta1
optional $nx2$ matrix of conditional probabilities for missingness mechanism, $P(Delta = 1|A = 0, V, W, T), P(Delta = 1|A = 1, V, W, T)$.

g.Deltaform
optional regression formula of the form $Delta\sim A+W$, if specified this overrides the call to SuperLearner

g.SL.library
optional vector of prediction algorithms to use for SuperLearner estimation of $g1W$ or $pDelta1$

ub
upper bound on observation weights. See Details section for more information

family
family specification for working regression models, generally ‘gaussian’ for continuous outcomes (default), ‘binomial’ for binary outcomes, ‘logistic’ (default), or ‘linear’

alpha
used to keep predicted initial values bounded away from (0,1) for logistic fluctuation

id
optional subject identifier

V_SL
number of cross-validation folds for Super Learner estimation of Q and g

inference
if TRUE, variance-covariance matrix, standard errors, pvalues, and 95% confidence intervals are calculated. Setting to FALSE saves a little time when bootstrapping.

verbose
status messages printed if set to TRUE (default=FALSE)

Q.discreteSL
If true, use discrete SL to estimate Q, otherwise ensembleSL by default. Ignored when SL is not used.

g.discreteSL
If true, use discrete SL to estimate components of g, otherwise ensembleSL by default. Ignored when SL is not used.

Details

$ub$ bounds the IC by bounding the factor $h(A, V)/[g(A, V, W)P(Delta = 1|A, V, W)]$ between 0 and $ub$, default value = 1/0.025.
### Value

- **psi**: MSM parameter estimate.
- **sigma**: variance covariance matrix.
- **se**: standard errors extracted from sigma.
- **pvalue**: two-sided p-value.
- **lb**: lower bound on 95% confidence interval.
- **ub**: upper bound on 95% confidence interval.
- **epsilon**: fitted value of epsilon used to target initial Q.
- **psi.Qinit**: MSM parameter estimate based on untargeted initial Q.
- **Qstar**: targeted estimate of Q, an $nx2$ matrix with predicted values for $Q(0,W), Q(1,W)$ using the updated fit.
- **Qinit**: initial estimate of Q, $Qinit$coef are the coefficients for a glm model for Q, if applicable. $Qinit$Q is an $nx2$ matrix, where n is the number of observations. Columns contain predicted values for $Q(0,W), Q(1,W)$ using the initial fit. $Qinit$type is method for estimating Q.
- **g**: treatment mechanism estimate. A list with three items: $g$g1W contains estimates of $P(A = 1 | W)$ for each observation, $g$coef the coefficients for the model for g when glm used, $g$type estimation procedure.
- **g.AV**: estimate for $h(A,V)$ or $h(A,T)$. A list with three items: $g.AV$g1W an $nx2$ matrix containing values of $P(A = 0 | V, T), P(A = 1 | V, T)$ for each observation, $g.AV$coef the coefficients for the model for g when glm used, $g.AV$type estimation procedure.
- **g_Delta**: missingness mechanism estimate. A list with three items: $g_Delta$g1W an $nx2$ matrix containing values of $P(Delta = 1 | A, V, W, T)$ for each observation, $g_Delta$coef the coefficients for the model for g when glm used, $g_Delta$type estimation procedure.

### Author(s)
Susan Gruber <sgruber@cal.berkeley.edu>, in collaboration with Mark van der Laan.

### References


### See Also
summary.tmleMSM, estimateQ, estimateG, calcSigma, tmle
Examples

library(tmle)
# Example 1. Estimating MSM parameter with correctly specified regression formulas
# MSM: psi0 + psi1*A + psi2*V + psi3*A*V (saturated)
# true parameter value: psi = (0, 1, -2, 0.5)
# generate data
set.seed(100)
n <- 1000
W <- matrix(rnorm(n*3), ncol = 3)
colnames(W) <- c("W1", "W2", "W3")
V <- rbinom(n, 1, 0.5)
A <- rbinom(n, 1, 0.5)
Y <- rbinom(n, 1, plogis(A - 2*V + 0.5*A*V))
result.ex1 <- tmleMSM(Y, A, W, V, MSM = "A*V", Qform = Y~., gform = A~1,
hAVform = A~1, family = "binomial")
print(result.ex1)
## Not run:
# Example 2. Repeated measures data, two observations per id
# (e.g., crossover study design)
# MSM: psi0 + psi1*A + psi2*V + psi3*V^2 + psi4*T
# true parameter value: psi = (-2, 1, 0, -2, 0)
# generate data in wide format (id, W1, Y(t), W2(t), V(t), A(t))
set.seed(10)
n <- 250
id <- rep(1:n)
W1 <- rbinom(n, 1, 0.5)
W2.1 <- rnorm(n)
W2.2 <- rnorm(n)
V.1 <- rnorm(n)
V.2 <- rnorm(n)
A.1 <- rbinom(n, 1, plogis(0.5 + 0.3 * W2.1))
A.2 <- 1-A.1
Y.1 <- -2 + A.1 - 2*V.1^2 + W2.1 + rnorm(n)
Y.2 <- -2 + A.2 - 2*V.2^2 + W2.2 + rnorm(n)
d <- data.frame(id, W1, W2=W2.1, W2.2, V=V.1, V.2, A=A.1, A.2, Y=Y.1, Y.2)
# change dataset from wide to long format
longd <- reshape(d,
varying = cbind(c(3, 5, 7, 9), c(4, 6, 8, 10)),
idvar = "id",
direction = "long",
timevar = "T",
new.row.names = NULL,
sep = "")
# misspecified model for initial Q, partial misspecification for g.
# V_SL set to 2 to save time, not recommended at this sample size
result.ex2 <- tmleMSM(Y = longd$Y, A = longd$A, W = longd[,c("W1", "W2")], V = longd$V,
T = longd$T, MSM = "A + V + I(V^2) + T", Qform = Y ~ A + V, gform = A ~ W,
id = longd$id, V_SL=2)
print(result.ex2)
# Example 3: Introduce 20
# V_SL set to 2 to save time, not recommended at this sample size
Delta <- rbinom(nrow(longd), 1, 0.8)
result.ex3 <- tmleMSM(Y = longd$Y, A = longd$A, W = longd[, c("W1", "W2")], V = longd$V, T = longd$T,
                   Delta = Delta, MSM = "A + V + I(V^2) + T", Qform = Y ~ A + V, gform = A ~ W2,
                   g.Deltaform = Delta - 1, id = longd$id, verbose = TRUE, V_SL = 2)
print(result.ex3)

## End(Not run)

---

**tmleNews**  
*Show the NEWS file (tmleNews)*

**Description**

Shows recent changes and bug fixes documented in the tmle package NEWS file.

**Usage**

`tmleNews(...)`

**Arguments**

...  
additional arguments passed to `RShowDoc`

**Value**

NONE

**Author(s)**

Susan Gruber

**See Also**

tmle, tmleMSM
Index

Topic datasets
  fev, 8
  calcParameters, 3, 7, 8, 17
  calcSigma, 4, 7, 8, 17, 22
  estimateG, 4, 5, 6, 8, 17, 22
  estimateQ, 4, 5, 7, 17, 22
  fev, 8
  methods, 10, 12
  oneStepATT, 9, 17
  predict.tmle.SL.dbarts2
    (tmle.SL.dbarts2), 18
  print.summary.tmle (summary.tmle), 10
  print.summary.tmleMSM
    (summary.tmleMSM), 12
  print.tmle (summary.tmle), 10
  print.tmleMSM(summary.tmleMSM), 12
  summary.tmle, 10, 17
  summary.tmleMSM, 12, 22
  SuperLearner, 19
  tmle, 3–5, 7–9, 11, 13, 22, 24
  tmle-package, 2
  tmle.SL.dbarts.k.5 (tmle.SL.dbarts2), 18
  tmle.SL.dbarts2, 18
  tmleMSM, 3–5, 7, 8, 13, 17, 20, 24
  tmleNews, 24