Package ‘tmle’
March 26, 2018

Version 1.3.0-1
Date 2018-3-15
Title Targeted Maximum Likelihood Estimation
Author Susan Gruber [aut, cre],
Mark van der Laan [aut]
Maintainer Susan Gruber <sgruber@cal.berkeley.edu>
Copyright Copyright 2012. The Regents of the University of California
(Regents). All Rights Reserved.
Depends SuperLearner
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 2018-03-25 23:09:56 UTC

R topics documented:

  tmle-package ......................................................... 2
Targeted Maximum Likelihood Estimation with Super Learning

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

<table>
<thead>
<tr>
<th>Package:</th>
<th>tmle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.3.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2018-01-31</td>
</tr>
<tr>
<td>License:</td>
<td>BSD</td>
</tr>
</tbody>
</table>

Author(s)

Susan Gruber, in collaboration with Mark van der Laan.

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

References

calcParameters


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=z, needed for CDE estimation
Delta indicator of missing outcome. 1 - observed, 0 - missing
g1W censoring mechanism estimates, P(A = 1|W) ∗ P(Delta = 1|A, W)
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

```r
calcSigma(hAV, gAW, Y, Q, mAV, covar.MSM, covar.MSMA0, covar.MSMA1, I.V, Delta, ub, id, family)
```
**estimateG**

**Arguments**

- `hAV`: values used in numerator of weights applied to the estimation procedure
- `gAVW`: \( p(A = a| V, W, T) \) \( p(Delta = 1| A, V, W, T) \)
- `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

---

**estimateG**  
*Estimate Treatment or Missingness Mechanism*

**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**

```
estimateG(d, g1W, gform, SL.library, id, V, verbose, message, outcome, newdata=d)
```
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 $A \cdot W$, (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', 'SL.step', 'SL.glm.interaction')
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.

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

Author(s)

Susan Gruber

See Also

tmle, estimateQ, calcParameters, tmleMSM, calcSigma

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.
estimateQ

Usage

```r
estimateQ(Y, Z, A, W, Delta, Q, Qbounds, Qform, maptoystar, SL.library, cvQinit,
family, id, V, verbose)
```

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.step`, `SL.glm.interaction`). 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=FALSE
- **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

Value

- **Q**: `n x 3` 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, `n x 5` 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, estimateQ, calcParameters, tmleMSM, calcSigma
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

Source


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, pDelta, depsilon, max_iter, gbounds, Qbounds)
summary.tmle

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>continuous or binary outcome variable</td>
</tr>
<tr>
<td>A</td>
<td>binary treatment indicator, 1 - treatment, 0 - control</td>
</tr>
<tr>
<td>Delta</td>
<td>indicator of missing outcome. 1 - observed, 0 - missing</td>
</tr>
<tr>
<td>Q</td>
<td>a 3-column matrix ( Q(A, W), Q(1, W), Q(0, W) )</td>
</tr>
<tr>
<td>g1W</td>
<td>treatment mechanism estimates, ( P(A = 1</td>
</tr>
<tr>
<td>pDelta1</td>
<td>censoring mechanism estimates, a 2-column matrix ( P(Delta = 1</td>
</tr>
<tr>
<td>depsilon</td>
<td>step size for delta moves, set to 0.001</td>
</tr>
<tr>
<td>max_iter</td>
<td>maximum number of iterations before terminating without convergence</td>
</tr>
<tr>
<td>gbounds</td>
<td>bounds on the propensity score for untreated subjects</td>
</tr>
<tr>
<td>Qbounds</td>
<td>alpha bounds on the logit scale</td>
</tr>
</tbody>
</table>

Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>psi</td>
<td>effect estimate (on the transformed scale for continuous outcomes)</td>
</tr>
<tr>
<td>IC</td>
<td>influence function</td>
</tr>
<tr>
<td>conv</td>
<td>TRUE if procedure converged, FALSE otherwise</td>
</tr>
</tbody>
</table>

Author(s)

Susan Gruber

See Also

tmle,

summary.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 $g$
- **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.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.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

Author(s)

Susan Gruber
See Also

tmle

Examples

```r
# 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+1", g1W=rep(.5, n))
summary(result)
```

Description

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

Usage

```r
## 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, p-values, 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)
- **g.Avcoef**: coefficient of each term in model for h(A,V)
- **g.Avtype**: description of estimation procedure for h(A,V)
- **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
- **psi.Qinit**: MSM parameter estimates based on initial (untargeted) estimated Q

Author(s)

Susan Gruber

See Also

- tmle

---

**tmle**

Targeted Maximum Likelihood Estimation

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` for treatment, `A=0` for control), and `W` is a matrix or data frame 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

tmle(y, a, w, Z=NULL, Delta = rep(1,length(y)), Q = NULL, Q.Z1 = NULL, Qform = NULL, Q.bounds = NULL, Q.SL.library = c("SL.glm", "SL.step", "SL.glm.interaction"), cvQinit = FALSE, gW = NULL, gform = NULL, gbound = 0.025, pZ1= NULL, g.Zform = NULL, pDelta1 = NULL, g.Deltaform = NULL, g.SL.library = c("SL.glm", "SL.step", "SL.glm.interaction"), family = "gaussian", fluctuation = "logistic", alpha = 0.995, id=1:length(y), V = 5, verbose = 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 \(Q\)
cvQinit logical, if TRUE, estimates cross-validated predicted values using discrete super learning, default=FALSE
gW 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
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 \(Z\sim A+ W\), 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, \(P(Delta = 1|Z = 0, A = 0, W), P(Delta = 1|Z = 0, A = 1, W), P(Delta = 1|Z = 1, A = 0, W), P(Delta = 1|Z = 1, A = 1, W)\)
optional regression formula of the form Delta~A+W, if specified this overrides the call to SuperLearner

optional vector of prediction algorithms to use for SuperLearner estimation of gW or pDelta1

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

‘logistic’ (default), or ‘linear’

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

optional subject identifier

Number of cross-validation folds for Super Learning to estimate Q and g

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

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.

Q.SL.library defaults to (‘SL.glm’, ‘SL.step’, ‘SL.glm.interaction’)

g.SL.library Defaults to (‘SL.glm’, ‘SL.step’, ‘SL.glm.interaction’)

This choice is simply because these algorithms are included in the base R installation. See SuperLearner help files for further information.

Value

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

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

$Z  intermediate covariate assignment estimate (when applicable). A list with three items: $Z$ is 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 $ when glm used, g$Z$type estimation procedure

$Delta  missingness mechanism estimate. A list with three items: $Delta$ is 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.) $Delta$coef the coefficients for the model for $ 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.tmle, estimateQ, estimateG, calcParameters, oneStepATT, tmleMSM, calcSigma*
Examples

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
result1 <- tmle(Y, A, W)
summary(result1)

# 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(.5*W[,1] + .4*W[,2] + .3*W[,3]))
# Y <- rbinom(n, 1, plogis(A + .2*W[,1] + .2*W[,2] + .1*W[,3]))
result2 <- tmle(Y, A, W, family="binomial", Qform=Y~A+W+W+R+W, gform=A+W+W+R+W)
summary(result2)

# Example 3: Population mean outcome
# User-supplied (misspecified) model for Q,
# Super learner called to estimate g, g.Delta
# approx. 20% missing at random
Y <- W[,1] + W[,2]^2 + rnorm(n)
Delta <- rbinom(n, 1, 1/(1+exp(-1.7-W[,1])))
result3 <- tmle(Y, A=NULL, W,Delta=NULL, Qform="Y-A+W+W+W", gform="A+W+W+W")
print(result3)

# Example 4: Controlled direct effect
# User-supplied models for g, g.Z
A <- rbinom(n, 1,.5)
Z <- rbinom(n, 1, plogis(.5*A + .1*W[,1]))
Y <- 1 + A + 10*Z + W[,1]+ rnorm(n)
total.effect <- tmle(Y, A, W, gform="A-1")
print(CDE)
print(total-effect)
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 \((Y, A, W, \text{MSM})\), 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. 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

```
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", "SL.step", "SL.glm.interaction"),
  cvQinit = FALSE, hAV = NULL, hAVform = NULL, gW = NULL,
  gform = NULL, pDelta1 = NULL, g.Deltaform = NULL,
  g.SL.library = c("SL.glm", "SL.step", "SL.glm.interaction"),
  ub = 1/0.025, family = "gaussian", fluctuation = "logistic",
  alpha = 0.995, id = 1:length(Y), V_SL = 5, inference = TRUE, verbose = 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 \(nx2\) 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=FALSE
optional \( nx \times 2 \) 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 \( nx \times 2 \) matrix of all 1s

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

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

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

optional \( nx \times 2 \) matrix of conditional probabilities for missingness mechanism, \( P(Delta = 1|A = 0, V, W, T), P(Delta = 1|A = 1, V, W, T) \).

optional regression formula of the form \( Delta \sim A + W \), if specified this overrides the call to SuperLearner

optional vector of prediction algorithms to use for SuperLearner estimation of \( gW \) or \( pDelta1 \)

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

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

‘logistic’ (default), or ‘linear’

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

optional subject identifier

number of cross-validation folds for Super Learner estimation of \( Q \) and \( g \)

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

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

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.

Defaults to (‘SL.glm’, ‘SL.step’, ‘SL.glm.interaction’)

Defaults to (‘SL.glm’, ‘SL.step’, ‘SL.glm.interaction’)

This choice is simply because these algorithms are included in the base R installation. See SuperLearner help files for further information.

MSM parameter estimate

variance covariance matrix

standard errors extracted from sigma

two-sided p-value
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)
```r
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, MSM = "A+V", Qform = Y - ., gform = A~1,
    hAVform = A~1, family = "binomial")
print(result.ex1)

# 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(100)
n <- 500
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=Y.1, V.2, A=Y1, A.2, Y=Y.1, Y.2)

# change dataset from wide to long format
longd <- rehape(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
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)
print(result.ex2)

# Example 3: Introduce 20% missingness in example 2 data
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, MSM = "A + V + I(V^2) + T", Qform = Y - A + V, gform = A ~ W,
    g.Deltaform = Delta~ 1, id=longd$id, verbose = TRUE)
print(result.ex3)
```

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, 6, 7, 15
  calcSigma, 4, 4, 6, 7, 15, 19

  estimateG, 4, 5, 5, 7, 15, 19
  estimateQ, 4–6, 6, 15, 19

  fev, 8

  methods, 9, 11

  oneStepATT, 8, 15

  print.summary.tmle (summary.tmle), 9
  print.summary.tmleMSM (summary.tmleMSM), 11
  print.tmle (summary.tmle), 9
  print.tmleMSM (summary.tmleMSM), 11

  summary.tmle, 9, 15
  summary.tmleMSM, 11, 19

  tmle, 3–7, 9, 11, 12, 19, 21
  tmle-package, 2
  tmleMSM, 3–7, 12, 15, 16, 21
  tmleNews, 20