Package 'OnlineSurr'

April 21, 2026

Type Package

Title Surrogate Evaluation for Jointly Longitudinal Outcome and Surrogate

Version 0.0.3

Description
Tools for surrogate evaluation in longitudinal studies using state-space models. The package estimates treatment effects over time with and without adjustment for surrogate information, summarizes the proportion of treatment effect explained by a longitudinal surrogate, quantifies uncertainty via bootstrap resampling, and provides plotting and summary utilities for fitted models.

License GPL (>= 3)

Encoding UTF-8

Imports kDGLM (>= 1.2.14), dplyr, ggplot2, tidyr, rlang, Rfast, stats, latex2exp, Rdpack

Suggests knitr, rmarkdown

RdMacros Rdpack

RoxygenNote 7.3.3

VignetteBuilder knitr

URL https://silvaneojunior.github.io/OnlineSurr/

Repository CRAN

BugReports https://github.com/silvaneojunior/OnlineSurr/issues

Depends R (>= 4.1.0)

NeedsCompilation no

Author Silvaneo dos Santos Jr. [aut, cre], Layla Parast [aut]

Maintainer Silvaneo dos Santos Jr. <silvaneojunior@utexas.edu>

Date/Publication 2026-04-21 18:02:37 UTC

Contents

`fit.surr` *Fit marginal and conditional state-space models for longitudinal surrogate evaluation*

Description

Fits two Gaussian state-space models (Dynamic Linear Models) to jointly longitudinal outcome data: (i) a marginal model for the outcome trajectory given treatment and time, and (ii) a conditional model that additionally adjusts for a user-specified surrogate structure. The function returns per-time treatment-effect estimates from both models and subject-level bootstrap draws obtained via subject-level resampling.

Usage

```
fit.surr(
  formula,
  id,
  surrogate,
  treat,
  data = NULL,
  time = NULL,
  N.boots = 2000,
  verbose = 1,
  D.local = 0.8
)
```

Arguments

`formula`	An object of class `formula` describing the fixed-effects mean structure for the primary outcome. The left-hand side must be the outcome variable. Internally, the right-hand side is augmented to include treatment-by-time fixed effects.
`id`	A variable (unquoted) identifying subjects. Each subject must have at most one measurement per `time` value.
`surrogate`	A formula describing the surrogate structure to be included in the conditional model. May be provided either as a `formula` (e.g., `~ s1 + s2`) or as a string that can be coerced to a formula.

`treat` A variable (unquoted) indicating treatment assignment. Must encode exactly two treatment levels after coercion to a factor.

`data` A `data.frame` containing all variables referenced in `formula`, `id`, `treat`, `surrogate`, and (optionally) `time`.

`time` Optional variable (unquoted) giving the measurement time index. Must be numeric and equally spaced across observed time points. If `NULL`, an equally spaced within-subject index is created in the current row order (with a warning).

`N.boots` Integer number of subject-level bootstrap replicates. Each replicate resamples subjects with replacement and recombines subject-specific sufficient quantities to form bootstrap draws of the fixed effects.

`verbose` Logical scalar indicating whether to print progress information during model fitting. If `TRUE`, progress updates are shown; if `FALSE`, no progress output is produced.

`D.local` Numeric, a number between 0 and 1 indicating the discount factor to be used for the random effect block. This factor controls how smooth the random effect evolve over time. A discount factor of 1 means that the random effects do not change over time, so that each individual has its own local level, but that level is the same for all times. A discount factor of 0 is not acceptable (the kDGLM package will replace it by 1), but values closer to 0 imply in a more flexible dynamic. See (West and Harrison 1997) or the appendix in (dos Santos Jr. and Parast 2026) for instructions on how to specify the discount factor.

Details

The implementation follows a two-model decomposition used for estimating longitudinal treatment effects and surrogate-adjusted (residual) treatment effects in a state-space framework.

See (dos Santos Jr. and Parast 2026) for details on the methodology.

See (West and Harrison 1997) for best practices on model specification in the state-space model setting.

Data requirements. The data must have at most one row per subject-time pair; time must be numeric and equally spaced (or omitted, in which case an index is created). Treatment and subject identifiers are coerced to factors with sorted levels.

Model structure. The marginal model includes treatment-by-time fixed effects and a subject-specific random-walk component to capture within-subject correlation. The conditional model adds the user-specified surrogate structure to the design, and checks that treatment is not a linear combination of the surrogate design (rank check).

Bootstrap. Subjects are resampled with replacement. Subject-specific filtered quantities are computed once and recombined in each bootstrap iteration to reduce computational cost, consistent with a subject-level nonparametric bootstrap strategy for replicated time series.

Value

An object of class `"fitted_onlinesurr"`: a named list with elements `$Marginal` and `$Conditional`. Each of these contains:

- point: the point estimate vector of the fixed effects (excluding subject-specific random-walk states) at the final time point.
- smp: a matrix of bootstrap draws for those fixed effects, with one column per bootstrap replicate.

The object also includes:

- T: number of unique time points.
- N: number of subjects.
- n.fixed: number of fixed-effect coefficients implied by formula for a single subject prior to stacking across subjects.

Examples

```
## Not run:
# data columns: y (outcome), id (subject id), trt (0/1 or two-level factor),
# time (numeric equally spaced), s1 and s2 (surrogates)

fit <- fit.surr(
  formula    = y ~ 1, # baseline fixed effects; function adds trt*time terms
  id         = id,
  surrogate  = ~ s1 + s2,
  treat      = trt,
  data       = dat,
  time       = time,
  N.boots    = 500
)

# Access point estimates and bootstrap samples
fit$Marginal$point
fit$Conditional$smp[, 1:10]

## End(Not run)
```

---

lagged *Compute lagged values of a vector*

Description

Returns a lagged version of x by shifting its values forward by k positions and padding the first k entries with zeros.

Usage

```
lagged(x, k = 1)
```

Arguments

x	A vector to be lagged.
k	A non-negative integer giving the lag order.

Details

This function is intended for use in model formulas when delayed effects of a predictor should be included explicitly.

Value

A vector of the same length as x, with the first k values set to 0 and the remaining values taken from x shifted by k positions.

Examples

```
x <- 1:5

lagged(x)
lagged(x, k = 2)
```

---

plot.fitted_onlinesurr

Plot time-varying PTE measures and treatment effects from a "fitted_onlinesurr" *object*

Description

Produces a ggplot2 figure showing, over time, either the Local PTE (LPTE), the Cumulative PTE (CPTE), or the marginal and residual treatment effects $\Delta(t)$ and $\Delta_R(t)$ (labeled Δ and Δ_R in the plot). Point estimates are taken from object$Marginal$point and object$Conditional$point, with uncertainty bands computed from the stored bootstrap draws.

Usage

```
## S3 method for class 'fitted_onlinesurr'
plot(x, type = "LPTE", conf.level = 0.95, one.sided = TRUE, ...)
```

Arguments

x	A "fitted_onlinesurr" object, typically returned by fit.surr. It must contain $T, $n.fixed, and the components $Marginal and $Conditional, each with point and smp.
type	Character string specifying what to plot. One of "LPTE", "CPTE", or "Delta" (case-insensitive). "Delta" plots both $\Delta(t)$ and $\Delta_R(t)$ with separate colors.

conf.level	Numeric in $(0, 1)$ giving the confidence level for the plotted intervals. Default is 0.95.
one.sided	Logical; if TRUE (default), uses signif.level = (1-conf.level)/2 when taking quantiles, so each tail excludes 1-conf.level (i.e., a wider interval than the usual two-sided conf.level interval). This is convenient when visually assessing one-sided surrogate validation criteria. If FALSE, uses the standard two-sided construction signif.level = 1-conf.level.
...	Additional arguments (currently unused) included for S3 method compatibility.

Details

The function extracts time-indexed treatment-effect estimates $\Delta(t)$ (marginal) and $\Delta_R(t)$ (residual/conditional) from the fitted object, along with bootstrap draws for each. It then constructs:

- **LPTE:** $\mathrm{LPTE}(t) = 1 - \Delta_R(t)/\Delta(t)$.
- **CPTE:** $\mathrm{CPTE}(t) = 1 - \sum_{u \le t} \Delta_R(u) / \sum_{u \le t} \Delta(u)$.
- **Delta:** plots $\Delta(t)$ and $\Delta_R(t)$ directly.

Point estimates are plotted as points; intervals are empirical quantile intervals computed from the bootstrap sample matrices stored in object.

Value

A ggplot object.

Examples

```
## Not run:
fit <- fit.surr(y ~ 1,
  id = id, surrogate = ~ s1 + s2, treat = trt,
  data = dat, time = time, N.boots = 2000
)

plot(fit, type = "LPTE")
plot(fit, type = "CPTE", conf.level = 0.90, one.sided = FALSE)
plot(fit, type = "Delta")

## End(Not run)
```

---

s *Construct a B-spline basis matrix*

Description

Build a B-spline basis for a numeric vector using a Cox-de Boor style recursion. By default, the function constructs a cubic spline basis (P = 3) and chooses the number of basis functions from the number of unique values in x.

Usage

```
s(
  x,
  P = 3,
  K = min(7, max(3, floor(log2(length(unique(x))))),
  limits = c(NA, NA),
  knots = "eq"
)
```

Arguments

x	A numeric vector of predictor values.
P	A non-negative integer giving the spline degree. P = 3 corresponds to a cubic B-spline basis.
K	An integer giving the number of basis functions to return. The default increases slowly with the number of unique values in x.
limits	A numeric vector of length 2 giving the lower and upper boundary limits for the spline basis. Missing values are replaced by min(x) and max(x).
knots	Either a numeric vector of knot locations, or one of "eq" or "quantile". If "eq", knots are placed uniformly between limits[1] and limits[2]. If "quantile", knots are placed at equally spaced empirical quantiles of x.

Details

Boundary limits are taken from x unless supplied explicitly. Knot locations may be given directly as a numeric vector, or generated either at equally spaced locations ("eq") or at empirical quantiles ("quantile").

The returned basis has length(x) rows and k columns.

When knots is generated internally, the function first creates K - P + 1 knot locations and then augments them with repeated boundary knots so the recursion can be evaluated.

Value

A numeric matrix with one row per element of x and one column per spline basis function.

Examples

```
x <- seq(0, 1, length.out = 10)

# Default cubic basis
B <- s(x)
dim(B)

# Equally spaced knots with custom basis size
B2 <- s(x, K = 5, knots = "eq")

# Quantile-based knots
B3 <- s(x, knots = "quantile")
```

summary.fitted_onlinesurr *Summarize a* "fitted_onlinesurr" *object*

Description

Prints a human-readable report for an object of class "fitted_onlinesurr" returned by fit.surr. The report includes marginal and conditional treatment-effect estimates at a selected time point (or cumulatively up to that time), an estimate of the LPTE/CPTE, and a time-homogeneity test of the LPTE.

Usage

```
## S3 method for class 'fitted_onlinesurr'
summary(object, t = object$T, cumulative = T, signif.level = 0.05, ...)
```

Arguments

object	A "fitted_onlinesurr" object.
t	Integer time index at which to evaluate treatment effects and the PTE. If cumulative = TRUE, effects are aggregated over times 1:t. If cumulative = FALSE, effects are evaluated at time t only.
cumulative	Logical; if TRUE (default), the report uses cumulative (up to time t) marginal and conditional treatment effects. If FALSE, the report uses the effects at time t only.
signif.level	Numeric in $(0, 1)$ giving the significance level for the time-homogeneity test that is reported (e.g., via time_homo_test).
...	Additional arguments passed to downstream summary/print utilities (if any).

Details

The "fitted_onlinesurr" object stores point estimates and bootstrap samples for marginal and surrogate-adjusted (conditional) models in object$Marginal and object$Conditional.

Value

No return value. Called for its side effect of printing a summary report.

Examples

```
## Not run:
fit <- fit.surr(y ~ 1,
  id = id, surrogate = ~ s1 + s2, treat = trt,
  data = dat, time = time, N.boots = 2000
)

# Cumulative up to time 5
```

```
summary(fit, t = 5, cumulative = TRUE, signif.level = 0.05)

# Time-specific at time 5
summary(fit, t = 5, cumulative = FALSE)

## End(Not run)
```

`time_homo_test` *Test time-homogeneity of the PTE*

Description

Tests the null hypothesis that the LPTE is constant over time. The test is based on the difference between the conditional and marginal treatment-effect trajectories implied by a fitted `"fitted_onlinesurr"` object, standardized by an estimated covariance, and uses a max-type statistic to control the family wise error across time points.

Usage

```
time_homo_test(model, signif.level = 0.05, N.boots = 50000)
```

Arguments

`model`	A fitted object of class `"fitted_onlinesurr"`, typically returned by `fit.surr`. Must contain `$T`, `$n.fixed`, and the elements `$Marginal` and `$Conditional` with `point` and `smp` components.
`signif.level`	Numeric in (0,1) giving the test significance level used to form the critical value from the bootstrap distribution. Default is `0.05`.
`N.boots`	Integer number of Monte Carlo draws used to approximate the null distribution of the max standardized deviation statistic and to compute the p-value. Default is `50000`.

Details

See (dos Santos Jr. and Parast 2026) for the theoretical details about this test.

Notes:

- The function assumes the first `T` time-specific treatment-effect parameters are stored contiguously at the beginning of `model$Marginal$point` and `model$Conditional$point` (and similarly for `smp`). It uses the index `1:(n.fixed)` as implemented in the code: `1:(T + n.fixed - T)`.
- `N.boots` here is a Monte Carlo size for the null simulation (distinct from the bootstrap size used when fitting `model`).

Value

A named list with:

- `T`: the observed test statistic (maximum absolute standardized deviation).
- `T.crit`: the 1-signif.level critical value.
- `p.value`: the Monte Carlo p-value `mean(T_null > T_obs)`.

Examples

```
## Not run:
fit <- fit.surr(y ~ 1,
  id = id, surrogate = ~ s1 + s2, treat = trt,
  data = dat, time = time, N.boots = 2000
)

time_homo_test(fit, signif.level = 0.05, N.boots = 50000)

## End(Not run)
```

Index