Package: rtForecastEval (via r-universe)

May 29, 2026

Title Evaluate the Discrepancy between Two Real-Time Updated Probabilistic Forecasts

Version 0.0.0.9000

Description Methods from Yeh, Rice, and Dubin (2022, doi:10.1080/00031305.2021.1967781; arXiv:2010.00781) for comparing two continuously updated probabilistic forecasts under squared (Brier) loss: pointwise loss and variance, a global delta test (Monte Carlo p-values), simulation designs, and a naive pointwise band plot.

URL https://github.com/chikuang/rtForecastEval

BugReports https://github.com/chikuang/rtForecastEval/issues

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Imports dplyr (>= 1.0.1), ggplot2 (>= 3.3.2), stats (>= 4.0.3), tidyr (>= 1.1.1), RSpectra (>= 0.16.0), rlist (>= 0.4.6.1), purrr (>= 0.3.4), MASS (>= 7.3-52)

Suggests pkgload (>= 1.2.0), knitr (>= 1.28.0), readr (>= 1.3.1), reshape2 (>= 1.4.4), rmarkdown (>= 2.1.0), gridExtra (>= 2.3.0), tibble (>= 3.0.3), sde

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libxml2-dev

Repository https://chikuang.r-universe.dev

Date/Publication 2026-04-14 21:10:11 UTC

RemoteUrl https://github.com/chikuang/rtforecasteval

RemoteRef HEAD

RemoteSha e62298f2bcc493661433f7130c9d53dac1f4fc87

Contents

---

rtForecastEval-package

rtForecastEval: Compare real-time probabilistic forecasts

Description

Methods from Yeh, Rice, and Dubin (2022, doi:10.1080/00031305.2021.1967781; arXiv:2010.00781) for comparing two continuously updated probabilistic forecasts under squared (Brier) loss: pointwise loss and variance, a global delta test (Monte Carlo p-values), simulation designs, and a naive pointwise band plot.

Author(s)

Maintainer: Chi-Kuang Yeh <chi-kuang.yeh@uwaterloo.ca>

Other contributors:

- Gregory Rice [contributor, thesis advisor]
- Joel A. Dubin [contributor, thesis advisor]

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). Evaluating real-time probabilistic forecasts with application to National Basketball Association outcome prediction. *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: https://arxiv.org/abs/2010.00781 (PDF: https://arxiv.org/pdf/2010.00781.pdf).

See Also

Useful links:

- https://github.com/chikuang/rtForecastEval
- Report bugs at https://github.com/chikuang/rtForecastEval/issues

calc_eig *Leading eigenvalues for the delta test covariance*

Description

Forms the empirical covariance matrix of either centered or non-centered forecast differences across time points (see `diff_cent` vs `diff_non_cent` in the paper), and returns the leading eigenvalues scaled by one over `nsamp`, for use with `calc_pval()`. This matches the construction in the replication `utility.R` (`calc_p_val` / `eigs_sym`).

Usage

```
calc_eig(df, n_eig = 10, ngame, nsamp, grid = "grid", cent = FALSE)
```

Arguments

`df`	Data frame containing `grid` and a column `diff_cent` or `diff_non_cent` (vector differences of forecasts across games, aligned within each time point).
`n_eig`	Number of leading eigenvalues to extract (dimension D in the paper).
`ngame`	Number of independent games (used for scaling inner products).
`nsamp`	Number of time grid points.
`grid`	Name of the time grid column.
`cent`	If `TRUE`, use `diff_cent` (centered differences); if `FALSE`, use `diff_non_cent`.

Value

A numeric vector of length `n_eig` of eigenvalues (descending).

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: https://arxiv.org/abs/2010.00781

calc_L_s2 *Pointwise mean loss difference and variance*

Description

For each time grid point, estimates the pointwise difference in expected squared (Brier) loss between two probabilistic forecasts, and the variance factor used for inference (Yeh, Rice, and Dubin, 2022). With default loss `L(x,y) = (x-y)^2`, the function computes the mean over games of `L(Y, phat_1) - L(Y, phat_2)` at each time, and the influence-style terms `si` used to form sigma-squared over four, matching the paper replication code.

Usage

```
calc_L_s2(
  df,
  pA = "phat_1",
  pB = "phat_2",
  Y = "Y",
  grid = "grid",
  L = function(x, y) (x - y)^2
)
```

Arguments

`df`	A data frame with one row per (game × time) in long format.
`pA, pB`	Column names for the two forecast vectors (probabilities between 0 and 1).
`Y`	Column name for the binary outcome (0/1).
`grid`	Column name for the normalized time grid between 0 and 1.
`L`	Loss function; default is squared error, \(L(x,y)=(x-y)^2\).

Value

A tibble with one row per `grid` value and columns `L` (mean loss difference), `sigma2` (variance factor), and `n` (number of rows).

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: https://arxiv.org/abs/2010.00781

`calc_pval` *Monte Carlo p-value and quantiles for the delta test*

Description

Given leading eigenvalues of the covariance operator used in the delta test (from `calc_eig()`) and the observed statistic \(Z\) (from `calc_Z()`), draws a Monte Carlo sample from the weighted sum of chi-square(1) variables and returns the right-tail *p*-value and quantiles of the null distribution (as in the paper's replication code).

Usage

```
calc_pval(Z, eig, quan, n_MC = 5000)
```

Arguments

Z	Observed test statistic from calc_Z().
eig	Numeric vector of leading eigenvalues (length n_eig).
quan	Probabilities for which to return quantiles of the simulated null statistic (e.g. c(0.90, 0.95, 0.99)).
n_MC	Number of Monte Carlo draws (default 5000).

Value

A list with p_val (one-sided p-value) and quantile (named vector of quantiles at quan).

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: https://arxiv.org/abs/2010.00781

---

calc_Z *Delta test statistic for comparing two forecasting methods*

Description

Computes the global test statistic Z for comparing two real-time forecasters under squared loss: Z equals (n_game / n_samp) times the sum over grid points of the squared pointwise mean loss differences, matching the implementation in the paper replication code (utility.R).

Usage

```
calc_Z(
  df,
  pA = "phat_1",
  pB = "phat_2",
  Y = "Y",
  grid = "grid",
  nsamp,
  ngame,
  L = function(x, y) (x - y)^2
)
```

Arguments

df	Data frame containing forecasts, outcomes, and grid.
pA, pB	Names of columns with the two probability forecasts.
Y	Name of the binary outcome column.
grid	Name of the column with normalized times between 0 and 1.

nsamp	Number of distinct time points (length of the grid).
ngame	Number of independent replicates (e.g. games).
L	Loss function; default is squared error.

Value

A single numeric value of the test statistic Z.

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: https://arxiv.org/abs/2010.00781

---

df_gen *Simulate real-time probabilistic forecasts (paper designs)*

Description

Generates replicated "games" and two competing forecast trajectories on a uniform grid from 0 to 1, following the simulation constructions used in Yeh, Rice, and Dubin (2022) (see `sanity_generator` / simulation scripts in the replication repository). Ornstein–Uhlenbeck (`type = "OU"`) or Brownian motion (`type = "BM"`) noise can drive the latent processes; outputs include binary outcomes `Y` and forecast probabilities `phat_A`, `phat_B` suitable for `calc_Z()`, `calc_eig()`, etc.

Usage

```
df_gen(N, Ngame, type = c("OU", "BM"), a = 1, b = 0.27)
```

Arguments

N	Number of interior time steps (grid has `N+1` points from 0 to 1).
Ngame	Number of independent replicates (games).
type	`"OU"` (default) or `"BM"` for the innovation process used in the bivariate construction.
a, b	Constants controlling the drift of the latent trajectory (see replication code).

Details

Requires the **sde** package (Suggests) for Brownian paths.

Value

A tibble with columns including `game`, `grid`, `Y`, `phat_A`, `phat_B`, and latent `W1`, `W2`, etc., in long format.

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: `https://arxiv.org/abs/2010.00781`

`lin_interp` *Linear interpolation of a forecast trajectory onto a grid*

Description

Maps a vector of probabilities `prob` observed at times `grid` onto an equally spaced grid of length `length(grid)` between 0 and 1 using linear interpolation (`stats::approx`). Useful when aligning ESPN or model outputs to the common grid used in the NBA analysis (see replication `pre_process` / `utility.R`).

Usage

```
lin_interp(prob, grid, outcome)
```

Arguments

`prob`	Numeric vector of forecast probabilities.
`grid`	Numeric vector of time points (same length as `prob`) between 0 and 1.
`outcome`	Scalar binary outcome \(Y\) to attach to every interpolated row.

Value

A tibble with columns `phat_approx`, `grid`, and `Y`.

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: `https://arxiv.org/abs/2010.00781`

plot_pcb — *Naive pointwise confidence band for mean loss difference*

Description

Plots the pointwise mean loss difference from `calc_L_s2()` (column `L`) against time, with a normal-theory band using `sigma2` and the point sample size `n`. This is the **naive** \(t\)-style band from the paper; global inference uses `calc_Z()` / `calc_pval()` instead.

Usage

```
plot_pcb(
  df,
  grid = "grid",
  L = "L",
  var = "sigma2",
  title = "Pointwise mean loss difference (A vs B)",
  subtitle = "Naive normal band (95%); use calc_pval() for global test",
  caption = NULL,
  xlab = "Normalized time (grid)",
  ylab = "Mean squared loss difference"
)
```

Arguments

`df`	Output of `calc_L_s2()` (or compatible tibble with `grid`, `L`, `sigma2`, `n`).
`grid`	Name of the x-axis column (default `"grid"`).
`L`	Name of the pointwise mean difference column (default `"L"`).
`var`	Name of the variance column (default `"sigma2"`).
`title`, `subtitle`, `caption`	Passed to `ggplot2::labs()`. Defaults describe the plot when `NULL` (some labels are omitted if `NULL`).
`xlab`, `ylab`	Axis labels (see `ggplot2::labs()`).

Details

This plot summarizes **relative skill** (loss difference) over time — not a classical **calibration** plot (forecast vs observed event rate). For a simple reliability-style view from the same long-format data, see the package vignette example that bins `phat` and plots mean `Y` vs mean forecast.

Value

A **ggplot2** object.

References

Yeh, C.-K., Rice, G., & Dubin, J. A. (2022). *The American Statistician*, *76*, 214–223. doi:10.1080/00031305.2021.1967781

Preprint: https://arxiv.org/abs/2010.00781

Index