

Package ‘fable.bayesRecon’

September 11, 2026

Date 2026-09-11

Title Bayesian Reconciliation in the 'fable' Framework

Version 0.2.1

Description Implements the 'bayesRecon' probabilistic reconciliation methods within the 'fable' framework for hierarchical time series forecasting. Bayesian reconciliation (bayesRecon) methods are accessed via the 'reconcile' verb, following 'fable' conventions. For methodological background, see Corani et al. (2021) <[doi:10.1007/978-3-030-67664-3_13](https://doi.org/10.1007/978-3-030-67664-3_13)>, Zambon et al. (2024a) <[doi:10.1007/s11222-023-10343-y](https://doi.org/10.1007/s11222-023-10343-y)>, Zambon et al. (2024b) <<https://proceedings.mlr.press/v244/zambon24a.html>>, and Carrara et al. (2026) <[doi:10.1016/j.ijforecast.2026.07.003](https://doi.org/10.1016/j.ijforecast.2026.07.003)>.

License LGPL (>= 3)

URL <https://github.com/dazzimonti/fable.bayesRecon>

BugReports <https://github.com/dazzimonti/fable.bayesRecon/issues>

Encoding UTF-8

Imports stats, bayesRecon, fabletools (>= 0.7.0), purrr, vctrs, distributional, dplyr, tsibble, rlang

Suggests testthat (>= 3.0.0), cli, fable, fable.intermittent (>= 0.3.0), ggtime, tsibbledata, tibble, ggplot2, scales, tidyr, knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 4.1)

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Dario Azzimonti [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0001-5080-3061>>), Stefano Damato [aut] (ORCID: <<https://orcid.org/0009-0001-0225-3736>>), Lorenzo Zambon [aut] (ORCID: <<https://orcid.org/0000-0002-8939-993X>>), Chiara Carrara [aut] (ORCID: <<https://orcid.org/0009-0002-7349-6249>>), Giorgio Corani [aut] (ORCID: <<https://orcid.org/0000-0002-1541-8384>>)

Maintainer Dario Azzimonti <dario.azzimonti@gmail.com>
Repository CRAN
Date/Publication 2026-09-11 16:40:02 UTC

Contents

bayesRecon_BUIS	2
bayesRecon_MixCond	3
bayesRecon_t	5
Index	8

bayesRecon_BUIS	<i>BUIS for probabilistic reconciliation of forecasts via conditioning</i>
-----------------	--

Description

Specifies Bottom-Up Importance Sampling (BUIS) reconciliation for use within `reconcile()`. The method uses the Bottom-Up Importance Sampling algorithm to draw samples from the reconciled forecast distribution, obtained via conditioning. Reconciliation is performed when `forecast()` is called on the resulting model. Marginal reconciled forecasts follow a sample distribution.

Usage

```
bayesRecon_BUIS(models, n_samples = 1000)
```

Arguments

- `models` A list of fitted models to reconcile.
- `n_samples` Number of samples to draw from the reconciled distribution.

Value

A model specification of class "bayesRecon_BUIS" (inheriting from "mdl_1st") to be passed to `reconcile`. The reconciliation is performed when `forecast` is called on the resulting mable (a model table). The forecast output is a fable object containing reconciled probabilistic forecasts represented as sample distributions (`dist_sample`).

References

Zambon, L., Azzimonti, D. & Corani, G. (2024). *Efficient probabilistic reconciliation of forecasts for real-valued and count time series*. Statistics and Computing 34 (1), 21. [doi:10.1007/s11222-02310343y](https://doi.org/10.1007/s11222-02310343y).

See Also

```
fabletools::reconcile(), fabletools::aggregate_key(), bayesRecon_MixCond(), bayesRecon::reconc_BUIS()
```

Examples

```

library(fable)
library(fabletools)
library(tsibble)
library(fable.bayesRecon)

# Small hierarchical example using tourism data
tourism_small <- tsibble::tourism |>
dplyr::filter(Region == "Melbourne") |>
fabletools::aggregate_key(Purpose, Trips = sum(Trips))

fit <- tourism_small |>
model(base = ETS(Trips ~ trend("A") + season("A"))) |>
reconcile(recon = bayesRecon_BUIS(base, n_samples = 300))

fc <- forecast(fit, h = "2 years")
fc

```

bayesRecon_MixCond	<i>Probabilistic reconciliation of mixed hierarchies</i>
--------------------	--

Description

bayesRecon_MixCond specifies Mixed-Conditioning (Mix-Cond) reconciliation for use within `reconcile()`. The method uses importance sampling to draw samples from the reconciled forecast distribution, obtained via conditioning, in the case of a mixed hierarchy.

bayesRecon_TDcond specifies Top-Down Conditioning reconciliation for use within `reconcile()`. The method uses a top-down conditioning algorithm: first, upper base forecasts are reconciled via conditioning using only the hierarchical constraints between the upper; then, the bottom distributions are updated via a probabilistic top-down procedure.

Reconciliation is performed when `forecast()` is called on the resulting model. Marginal reconciled forecasts follow a sample distribution.

Usage

```

bayesRecon_MixCond(models, n_samples = 1000, suppress_warnings = TRUE)

bayesRecon_TDcond(
  models,
  n_samples = 1000,
  suppress_warnings = TRUE,
  min_fraction_samples_ok = 0.5
)

```

Arguments

<code>models</code>	A list of fitted models to reconcile.
<code>n_samples</code>	Number of samples to draw from the reconciled distribution.
<code>suppress_warnings</code>	If TRUE, suppress warnings from reconciliation.
<code>min_fraction_samples_ok</code>	Minimum fraction of samples that must be valid for reconciliation to succeed in TDcond(default: 0.5).

Value

A model specification of class "bayesRecon_MixCond" or "bayesRecon_TDcond" (inheriting from "mdl_1st") to be passed to `reconcile`. The reconciliation is performed when `forecast` is called on the resulting mable (a model table). The forecast output is a fable object containing reconciled probabilistic forecasts represented as sample distributions (`dist_sample`).

References

Zambon, L., Azzimonti, D., Rubattu, N., Corani, G. (2024). *Probabilistic reconciliation of mixed-type hierarchical time series*. Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence, PMLR 244:4078-4095. <https://proceedings.mlr.press/v244/zambon24a.html>.

See Also

`fabletools::reconcile()`, `fabletools::aggregate_key()`, `bayesRecon_BUIS()`, `bayesRecon::reconc_MixCond()`, `bayesRecon::reconc_TDcond()`

Examples

```
library(tsibble)
library(dplyr)
library(fable)
library(fabletools)
library(fable.intermittent)

# Mixed hierarchy with integer-valued bottom and one upper aggregate, from the `auto` dataset.
# Forecasts with EMPSD for bottom base and ETS for smooth base forecasts.

# Helper: split a hierarchy into its upper (aggregated) and bottom (leaf) series so that a
# different base model can be fitted at each level.
hier_filter <- function(data, level = c("upper", "bottom")) {
  level <- match.arg(level)
  key_cols <- tsibble::key_vars(data)
  if (level == "upper") {
    dplyr::filter(data, dplyr::if_any(dplyr::all_of(key_cols), fabletools::is_aggregated))
  } else {
    dplyr::filter(data, !dplyr::if_any(dplyr::all_of(key_cols), fabletools::is_aggregated))
  }
}
```

```

}

# Build a small two-level hierarchy from a handful of auto spare-part series.
# The four series become the bottom level; their sum is the upper aggregate.
sel <- c("TS1461", "TS2953", "TS333", "TS2710")
data <- auto |>
  dplyr::filter(series_id %in% sel) |>
  fabletools::aggregate_key(series_id, value = sum(value))

# Hold out the last 6 months for forecasting.
train <- data |> dplyr::filter(index < tsibble::yearmonth("2011 Jul"))

# Base forecasts: a smooth (Gaussian) model on the continuous-looking upper aggregate,
# and the EMPSD non-parametric count model on the intermittent bottom series.
fit_upper <- train |>
  hier_filter("upper") |>
  fabletools::model(base = fable::ETS(value))

fit_bottom <- train |>
  hier_filter("bottom") |>
  fabletools::model(base = fable.intermittent::EMPSD(value))

fit <- dplyr::bind_rows(fit_upper, fit_bottom)
print(fit)

# Reconcile the mixed hierarchy via Mix-Cond and Top-Down conditioning.
fit <- fit |>
  fabletools::reconcile(
    mixcond = fable.bayesRecon::bayesRecon_MixCond(base),
    tdcond = fable.bayesRecon::bayesRecon_TDcond(base)
  )

fc <- fabletools::forecast(fit, h = 6)
print(fc)

if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc |> ggtime::autoplot(data, level=c(95))
}

```

bayesRecon_t

t-Rec: reconciliation via conditioning with uncertain covariance via multivariate t-distribution

Description

Specifies t-Rec reconciliation for use within reconcile().

Reconciles base forecasts by conditioning on the hierarchical constraints. The base forecasts are assumed to be jointly Gaussian, conditionally on the covariance matrix of the forecast errors. A Bayesian approach is adopted to account for the uncertainty of the covariance matrix. An Inverse-Wishart prior is specified on the covariance matrix, leading to a multivariate t-distribution for the base forecasts. The reconciliation via conditioning is in closed-form, yielding a multivariate t reconciled distribution.

Reconciliation is performed when `forecast()` is called on the resulting model. Marginal reconciled forecasts follow a Student-t distribution.

Usage

```
bayesRecon_t(models, ...)
```

Arguments

- | | |
|---------------------|---|
| <code>models</code> | A list of fitted models to reconcile. |
| <code>...</code> | Additional arguments passed to other methods, including: <ul style="list-style-type: none"> • <code>prior</code>: Optional list with entries <code>nu</code> and <code>Psi</code> specifying the parameters of the Inverse-Wishart prior distribution for the covariance matrix. If not provided, the prior is estimated from the data. • <code>freq</code>: Optional frequency of the time series, used for estimating the naive covariance matrix via seasonal naive residuals. If not provided, the frequency is inferred from the data. • <code>criterion</code>: Criterion for estimating the naive covariance matrix (default: "RSS"). • <code>l_shr</code>: Optional shrinkage parameter (between 0 and 1) for the covariance matrix of the residuals. |

Value

A model specification of class "bayesRecon_t" (inheriting from "mdl_lst") to be passed to `reconcile`. The reconciliation is performed when `forecast` is called on the resulting mable (a model table). The forecast output is a fable object containing reconciled probabilistic forecasts represented as sample distributions (`dist_student_t`).

References

Carrara, C., Azzimonti, D., Corani, G., Zambon, L. (2026). *Modeling uncertainty in the covariance matrix for probabilistic forecast reconciliation*. International Journal of Forecasting. In press, [doi:10.1016/j.ijforecast.2026.07.003](https://doi.org/10.1016/j.ijforecast.2026.07.003).

See Also

`fabletools::reconcile()`, `fabletools::aggregate_key()`, `fabletools::min_trace()`, `bayesRecon::reconc_t()`

Examples

```
library(fable)
library(fabletools)
library(tsibble)
library(fable.bayesRecon)

# Small hierarchical example using tourism data
tourism_small <- tsibble::tourism |>
dplyr::filter(Region == "Melbourne") |>
fabletools::aggregate_key(Purpose, Trips = sum(Trips))

fit <- tourism_small |>
model(base = ETS(Trips ~ trend("A") + season("A"))) |>
reconcile(recon = bayesRecon_t(base))

fc <- forecast(fit, h = "2 years")
fc
```

Index

bayesRecon::reconc_BUIS(), [2](#)
bayesRecon::reconc_MixCond(), [4](#)
bayesRecon::reconc_t(), [6](#)
bayesRecon::reconc_TDcond(), [4](#)
bayesRecon_BUIS, [2](#)
bayesRecon_BUIS(), [4](#)
bayesRecon_MixCond, [3](#)
bayesRecon_MixCond(), [2](#)
bayesRecon_t, [5](#)
bayesRecon_TDcond (bayesRecon_MixCond),
[3](#)

dist_sample, [2](#), [4](#)
dist_student_t, [6](#)

fabletools::aggregate_key(), [2](#), [4](#), [6](#)
fabletools::min_trace(), [6](#)
fabletools::reconcile(), [2](#), [4](#), [6](#)
forecast, [2](#), [4](#), [6](#)

reconcile, [2](#), [4](#), [6](#)