Package: BINtools (via r-universe)

October 29, 2026

Title Bayesian BIN (Bias, Information, Noise) Model of Forecasting

Version 0.2.0

Description A recently proposed Bayesian BIN model disentangles the underlying processes that enable forecasters and forecasting methods to improve, decomposing forecasting accuracy into three components: bias, partial information, and noise. By describing the differences between two groups of forecasters, the model allows the user to carry out useful inference, such as calculating the posterior probabilities of the treatment reducing bias, diminishing noise, or increasing information. It also provides insight into how much tamping down bias and noise in judgment or enhancing the efficient extraction of valid information from the environment improves forecasting accuracy. This package provides easy access to the BIN model. For further information refer to the paper Ville A. Satopää, Marat Salikhov, Philip E. Tetlock, and Barbara Mellers (2021) ``Bias, Information, Noise: The BIN Model of Forecasting'' <doi:10.1287/mnsc.2020.3882>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Biarch true

Depends R (>= 3.4.0)

Imports methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), dplyr (>= 1.0.2), tibble (>= 3.0.3), stringi (>= 1.4.6), mvtnorm (>= 1.1.1), combinat (>= 0.0.8), rstantools

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Suggests testthat, knitr, rmarkdown, pacman (>= 0.5.1), RcppParallel (>= 5.0.1)

VignetteBuilder knitr

NeedsCompilation yes

Author Ville Satopää [aut, cre]
(<https://orcid.org/0000-0002-3787-7303>), Marat Salikhov
[aut], Elvira Moreno [aut]

Maintainer Ville Satopää <ville.satopaa@gmail.com>

Config/pak/sysreqs make

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

Date/Publication 2022-04-04 11:40:02 UTC

RemoteUrl https://github.com/cran/BINtools

RemoteRef HEAD

RemoteSha 95b3ff5f661bdfd60c959127919a8f0b621b0062

Contents

BINtools-package 2
complete_summary 3
estimate_BIN 5
simulate_data 6

Index **8**

`BINtools-package` *The 'BINtools' package.*

Description

A DESCRIPTION OF THE PACKAGE

References

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. https://mc-stan.org

`complete_summary` *Summary*

Description

This function uses the return value of a call to the function `estimate_BIN` and produces a full BIN analysis based on that object.

Usage

```
complete_summary(full_bayesian_fit)
```

Arguments

`full_bayesian_fit`
: The return value of a call to function `estimate_BIN`.

Value

List containing the parameter estimates of the model, the posterior inferences, and the analysis of predictive performance.

The elements of the list are as follows.

- Parameter Estimates: Posterior means, standard deviations, and different quantiles of the model parameters and their differences. The parameter values represent the following quantities.
  - mu_star: Base rate of the event outcome in the probit scale. E.g., if mu_star = 0, then, in expectation, Phi(0)*100% = 50% of the events happen, where Phi(.) is the CDF of a standard Gaussian random variable.
  - mu_0: The level of bias in the control group. This can be any number in the real-line. E.g., if mu_0 = 0.1, then the control group believes the base rate to be Phi(mu_star + 0.1).
  - mu_1: The level of bias in the treatment group; otherwise the interpretation is the same as above for mu_0.
  - gamma_0: The level of information in the control group. This is a number between 0 and 1, where 0 represents no information and 1 represents full information.
  - gamma_1: The level of information in the treatment group; otherwise the interpretation is the same as above for gamma_0.
  - delta_0: The level of noise in the control group. This is a positive value, with higher values indicating higher levels of noise. Noise is in the same scale as information. E.g., delta_0 = 1.0 says that the control group uses as many irrelevant signals as there are relevant signals in the universe. In this sense it represents a very high level of noise.
  - delta_1: The level of noise in the treatment group; otherwise the interpretation is the same as above for delta_0.
  - rho_0: The level of within-group dependence between forecasts of the control group. This is a positive value, with higher values indicating higher levels of dependence. The dependence can be interpreted to stem from shared irrelevant (noise) and/or relevant (information) signals.

  - rho_1: The level of within-group dependence between forecasts of the treatment group; otherwise the interpretation is the same as above for rho_0.
  - rho_01: The level of inter-group dependence between forecasts of the control and treatment groups; otherwise the interpretation is the same as above for rho_0.

- Posterior Inferences: Posterior probabilities of events. Compared to the control group, does the treatment group have: (i) less bias, (ii) more information, and (iii) less noise? Intuitively, one can think of these probabilities as the Bayesian analogs of the p-values in classical hypothesis testing – the closer the probability is to 1, the stronger the evidence for the hypothesis.
- Control,Treatment: This compares the control group against the treatment group. The value of the contribution gives
  - the mean Brier score of the control group;
  - the mean Brier score of the treatment group;
  - how the difference can be explained in terms of bias, noise, or information; and
  - in percentage terms, how does the change in bias, noise, or information (from control to treatment group) changes the Brier score.
- Control,Perfect Accuracy: This compares the control group against a treatment group with perfect accuracy; otherwise the interpretation is the same as above for 'Control,Treatment.'

See Also

simulate_data, estimate_BIN

Examples

```
## An example with one group
# a) Simulate synthetic data:
synthetic_data = simulate_data(list(mu_star = -0.8,mu_0 = -0.5,mu_1 = 0.2,gamma_0 = 0.1,
gamma_1 = 0.3,rho_0 = 0.05,delta_0 = 0.1,rho_1 = 0.2, delta_1 = 0.3,rho_01 = 0.05),300,100,0)
# b) Estimate the BIN-model on the synthetic data:
full_bayesian_fit = estimate_BIN(synthetic_data$Outcomes,synthetic_data$Control, warmup = 500,
iter = 1000)
# c) Analyze the results:
complete_summary(full_bayesian_fit)


## An example with two groups
# a) Simulate synthetic data:
synthetic_data = simulate_data(list(mu_star = -0.8,mu_0 = -0.5,mu_1 = 0.2,gamma_0 = 0.1,
gamma_1 = 0.3, rho_0 = 0.05,delta_0 = 0.1, rho_1 = 0.2, delta_1 = 0.3,rho_01 = 0.05), 300,100,100)
# b) Estimate the BIN-model on the synthetic data:
full_bayesian_fit = estimate_BIN(synthetic_data$Outcomes,synthetic_data$Control,
synthetic_data$Treatment, warmup = 500, iter = 1000)
# c) Analyze the results:
complete_summary(full_bayesian_fit)
```

`estimate_BIN` *Estimate a BIN (Bias, Information, Noise) model*

Description

This function allows the user to compare two groups (treatment and control) of forecasters in terms of their bias, information, and noise levels. Model estimation is performed with a Markov Chain Monte Carlo (MCMC) approach called Hamiltonian Monte Carlo.

Usage

```
estimate_BIN(
  Outcomes,
  Control,
  Treatment = NULL,
 initial = list(mu_star = 0, mu_0 = 0, mu_1 = 0, gamma_0 = 0.4, gamma_1 = 0.4, delta_0
    = 0.5, rho_0 = 0.27, delta_1 = 0.5, rho_1 = 0.27, rho_01 = 0.1),
  warmup = 2000,
  iter = 4000,
  seed = 1
)
```

Arguments

`Outcomes`	Vector of binary values indicating the outcome of each event. The j-th entry is equal to 1 if the j-th event occurs and equal to 0 otherwise.
`Control`	List of vectors containing the predictions made for each event by forecasters in the control group. The j-th vector contains predictions for the j-th event.
`Treatment`	(Default:`NULL`) List of vectors containing the predictions made for each event by forecasters in the treatment group. The j-th vector contains predictions for the j-th event.
`initial`	A list containing the initial values for the parameters mu_star,mu_0,mu_1,gamma_0,gamma_1,delta_0,rho rho_01. (Default: `list(mu_star = 0,mu_0 = 0,mu_1 = 0,gamma_0 = 0.4,gamma_1 = 0.4,delta_0 = 0.5,rho_0 = 0.27, delta_1 = 0.5,rho_1 = 0.27,rho_01 = 0.1)`)
`warmup`	The number of initial iterations used for "burnin." These values are not included in the analysis of the model. (Default:`2000`)
`iter`	Total number of iterations. Must be larger than warmup. (Default:`4000`)
`seed`	(Default: `1`)

Value

Model estimation is performed with the statistical programming language called Stan. The return object is a Stan model. This way the user can apply available diagnostics tools in other packages, such as rstan, to analyze the final results.

See Also

simulate_data, complete_summary

Examples

```
## An example with one group
# a) Simulate synthetic data:
synthetic_data = simulate_data(list(mu_star = -0.8,mu_0 = -0.5,mu_1 = 0.2,gamma_0 = 0.1,
gamma_1 = 0.3,rho_0 = 0.05,delta_0 = 0.1,rho_1 = 0.2, delta_1 = 0.3,rho_01 = 0.05),300,100,0)
# b) Estimate the BIN-model on the synthetic data:
full_bayesian_fit = estimate_BIN(synthetic_data$Outcomes,synthetic_data$Control, warmup = 500,
iter = 1000)
# c) Analyze the results:
complete_summary(full_bayesian_fit)


## An example with two groups
# a) Simulate synthetic data:
synthetic_data = simulate_data(list(mu_star = -0.8,mu_0 = -0.5,mu_1 = 0.2,gamma_0 = 0.1,
gamma_1 = 0.3, rho_0 = 0.05,delta_0 = 0.1, rho_1 = 0.2, delta_1 = 0.3,rho_01 = 0.05), 300,100,100)
# b) Estimate the BIN-model on the synthetic data:
full_bayesian_fit = estimate_BIN(synthetic_data$Outcomes,synthetic_data$Control,
synthetic_data$Treatment, warmup = 500, iter = 1000)
# c) Analyze the results:
complete_summary(full_bayesian_fit)
```

---

simulate_data *Simulate Data*

Description

This function allows the user to generate synthetic data of two groups (control and treatment) of forecasters making probability predictions of binary events. The function is mostly useful for testing and illustration purposes.

Usage

```
simulate_data(parameters, N, N_0, N_1, rho_o = 0)
```

Arguments

parameters	A list containing the true values of the parameters: mu_star,mu_0,mu_1,gamma_0,gamma_1,rho_0,delta_ and rho_01
N	Number of events
N_0	Number of forecasters in the control group
N_1	Number of forecasters in the treatment group

`rho_o` The level of dependence between event outcomes. (Default: the events are independent conditional on the model parameter values. This sets `rho_ = 0.0`)

Details

See `complete_summary` for a description of the model parameters. Not all combinations of parameters are possible. In particular, the covariance parameters gamma and rho are dependent on each other and must result in a positive semi-definite covariance matrix for the outcomes and predictions. To find a feasible set of parameters, we recommend users to experiment: begin with the desired levels of mu, gamma, and delta, and values of rho close to zero, and then increase rho until data can be generated without errors.

Value

List containing the simulated data. The elements of the list are as follows.

- Outcomes: Vector containing binary values that indicate the outcome of each event. The j-th entry is equal to 1 if the j-th event occurs and equal to 0 otherwise.
- Control: List of vectors (one for each event) containing probability predictions made by the forecasters in the control group.
- Treatment: List of vectors (one for each event) containing probability predictions made by the forecasters in the treatment group.

See Also

`estimate_BIN`, `complete_summary`

Examples

```
simulate_data(list(mu_star = -0.8,mu_0 = -0.5,mu_1 = 0.2,gamma_0 = 0.1,gamma_1 = 0.3,
rho_0 = 0.05,delta_0 = 0.1,rho_1 = 0.2, delta_1 = 0.3,rho_01 = 0.05), 300,100,100)
```

Index

BINtools *(*BINtools-package*)*, 2
BINtools-package, 2

complete_summary, 3, *6, 7*

estimate_BIN, *4*, 5, *7*

simulate_data, *4*, 6, *6*