Package ‘multisite.accuracy’

July 31, 2024

Type Package

Title Estimation of Accuracy in Multisite Machine-Learning Models

Version 1.3

Date 2024-07-31

Description The effects of the site may severely bias the accuracy of a multisite machine-learning model, even if the analysts removed them when fitting the model in the 'training set' and applying the model in the 'test set' (Solanes et al., Neuroimage 2023, 265:119800). This simple R package estimates the accuracy of a multisite machine-learning model unbiasedly, as described in (Solanes et al., Psychiatry Research: Neuroimaging 2021, 314:111313). It currently supports the estimation of sensitivity, specificity, balanced accuracy (for binary or multinomial variables), the area under the curve, correlation, mean squarer error, and hazard ratio for binomial, multinomial, gaussian, and survival (time-to-event) outcomes.

License GPL-3

Imports coxme, lme4, lmerTest, logistf, metafor, pROC, survival

NeedsCompilation no

Author Joaquim Radua [aut, cre] (<https://orcid.org/0000-0003-1240-5438>)

Maintainer Joaquim Radua <quimradua@gmail.com>

Repository CRAN

Date/Publication 2024-07-31 18:10:02 UTC

Contents

multisite.accuracy *Estimate accuracy in multisite machine learning studies*

Description

Function to estimate the accuracy of the predictions of a multisite machine-learning model, controlling the effects of the site.

Usage

```
multisite.accuracy(y, y.pred, site, estimate = c("auc", "bac", "cor", "hr", "mse"),
                   site.method = "covar", mixed = FALSE, min.n = 10, ...)
```

Arguments

`y`	actual value of the variable that you want to predict. For `"auc"` estimates, it must be a binary vector. For `"bac"` estimates, it must be either a binary vector or a factor. For `"cor"` and `"mse"` estimates, it must be a numeric vector. For `"hr"` estimates, it must be an object of class `"Surv"`.
`y.pred`	predicted value of the variable. For `"auc"`, `"cor"`, `"hr"`, and `"mse"` estimates, it must be a numeric vector. For `"bac"` estimates, it must be either a binary vector or a factor with the same levels as `"y"`.
`site`	vector with the name of the site for each observation. Ignored if `site.method` is `"none"`.
`estimate`	`"auc"` (area under the ROC curve, when `"y"` is a binary variable and `"y.pred"` is a continous variable), `"bac"` (balanced accuracy, along with sensitivity and specificity when both `"y"` and `"y.pred"` are binary variables; or along with the sensitivity to detect each group when both `"y"` and `"y.pred"` are factors), `"cor"` and `"mse"` (correlation and mean squared error, when both `"y"` and `"y.pred"` are numeric variables), `"hr"` (hazard ratio, when `"y"` is an object of class `"Surv"` and `"y.pred"` is a numeric variable).
`site.method`	`"covar"` (site as covariate, e.g., using `lm`), `"meta"` (meta-analysis across sites), or `"none"` (no control of the effects of the site).
`mixed`	try to fit mixed-effects models, only for `site.method` = `"covar"` and `estimate` = `"cor"`, `"mse"`, or `"hr"`.
`min.n`	sites below this minimum sample size will be ignored.
`...`	further arguments for `logistf`, `lm`, `rma`, etc.

Details

We refer the reader to the publication below for details on the calculations.

Value

A data frame with the estimated accuracy, the methods used, and any warning or error.

References

Solanes, A., Palau, P., Fortea, L., Salvador, R., Gonzalez-Navarro, L., Llach, C.D., Valenti, M., Vieta, E., Radua, J. (2021) Biased accuracy in multisite machine-learning studies due to incomplete removal of the effects of the site. *Psychiatry Research: Neuroimaging*, 314:111313. Solanes, A., Gosling, C.J., Fortea, L., Ortuno, M., Lopez-Soley, E., Llufriu, S., Madero, S., Martinez-Heras, E., Pomarol-Clotet, E., Solana, E., Vieta, E., Radua, J. (2021) Removing the effects of the site in brain imaging machine-learning - Measurement and extendable benchmark. *Neuroimage*, 265:119800.

Examples

```
for (real_effects in c(FALSE, TRUE)) {
  for (eos in c(FALSE, TRUE)) {

    # Simulate data
    site = rep(c("Site A", "Site B"), 200)
    y = c(scale(rnorm(400)))
    y.pred = c(scale(residuals(lm(rnorm(400) ~ y))))
    # If real effects:
    if (real_effects) {
      y.pred = c(scale(y.pred + y))
    }
    # If effects of the site:
    if (eos) {
      y = c(scale(y + 2 * (site == "Site B")))
      y.pred = c(scale(y.pred + 2 * (site == "Site B")))
    }
    cat("\n=== Real effects:", ifelse(real_effects, "yes", "no"),
        "\n=== Effects of the site:", ifelse(eos, "yes", "no"), "\n\n")

    # Numeric: without real effects, mse.pred should not be < mse.mean and cor should be ~0
    print(rbind(
      multisite.accuracy(y, y.pred, site, "mse", site.method = "covar"),
      multisite.accuracy(y, y.pred, site, "mse", site.method = "meta"),
      multisite.accuracy(y, y.pred, site, "mse", site.method = "none"))[,1:5])
    print(rbind(
      multisite.accuracy(y, y.pred, site, "cor", site.method = "covar"),
      multisite.accuracy(y, y.pred, site, "cor", site.method = "meta"),
      multisite.accuracy(y, y.pred, site, "cor", site.method = "none"))[,1:3])

  }
}
```

Index