Package ‘PheNorm’

October 12, 2022

Type  Package
Title  Unsupervised Gold-Standard Label Free Phenotyping Algorithm for EHR Data
Version  0.1.0
Description  The algorithm combines the most predictive variable, such as count of the main International Classification of Diseases (ICD) codes, and other Electronic Health Record (EHR) features (e.g. health utilization and processed clinical note data), to obtain a score for accurate risk prediction and disease classification. In particular, it normalizes the surrogate to resemble gaussian mixture and leverages the remaining features through random corruption denoising. Background and details about the method can be found at Yu et al. (2018) <doi:10.1093/jamia/ocx111>.
License  GPL-3
Encoding  UTF-8
RoxygenNote  7.1.1
URL  https://github.com/celehs/PheNorm
BugReports  https://github.com/celehs/PheNorm/issues
Suggests  knitr, rmarkdown, testthat
VignetteBuilder  knitr
NeedsCompilation  no
Author  Sheng Yu [aut],
        Victor Castro [aut],
        Clara-Lea Bonzel [aut, cre],
        Molei Liu [aut],
        Chuan Hong [aut],
        Tianxi Cai [aut],
        PARSE LTD [aut]
Maintainer  Clara-Lea Bonzel <clbonzel@hsph.harvard.edu>
Repository  CRAN
Date/Publication  2021-01-07 13:50:05 UTC
R topics documented:

*PheNorm.Prob*

Fit the phenotyping algorithm *PheNorm* using EHR features

Description

The function requires as input: *a surrogate, such as the ICD code *the healthcare utilization It can leverage other EHR features (optional) to assist risk prediction.

Usage

```
PheNorm.Prob(
  nm.logS.ori,
  nm.utl,
  dat,
  nm.X = NULL,
  corrupt.rate = 0.3,
  train.size = 10 * nrow(dat)
)
```

Arguments

- `nm.logS.ori` name of the surrogates (log(ICD+1), log(NLP+1) and log(ICD+NLP+1))
- `nm.utl` name of healthcare utilization (e.g. note count, encounter_num etc)
- `dat` all data columns need to be log-transformed and need column names
- `nm.X` additional features other than the main ICD and NLP
- `corrupt.rate` rate for random corruption denoising, between 0 and 1, default value=0.3
- `train.size` size of training sample, default value 10 * nrow(dat)

Value

list containing probability and beta coefficient

Examples

```r
## Not run:
set.seed(1234)
fit.dat <- read.csv("https://raw.githubusercontent.com/celehs/PheNorm/master/data-raw/data.csv")
fit.phenorm=PheNorm.Prob("ICD", "utl", fit.dat, nm.X = NULL,
  corrupt.rate=0.3, train.size=nrow(fit.dat));
head(fit.phenorm$probs)
## End(Not run)
```
Index

PheNorm.Prob, 2