Package: LTCDM (via r-universe)

September 13, 2024

Type Package

Title Latent Transition Cognitive Diagnosis Model with Covariates

Version 1.0.0

Description Implementation of the three-step approach of latent transition cognitive diagnosis model (CDM) with covariates. This approach can be used to assess changes in attribute mastery status and to evaluate the covariate effects on both the initial states and transition probabilities over time using latent logistic regression. Because stepwise approaches often yield biased estimates, correction for classification error probabilities (CEPs) is considered in this approach. The three-step approach for latent transition CDM with covariates involves the following steps: (1) fitting a CDM to the response data without covariates at each time point separately, (2) assigning examinees to latent states at each time point and computing the associated CEPs, and (3) estimating the latent transition CDM with the known CEPs and computing the regression coefficients. The method was proposed in Liang et al. (2023) <doi:10.3102/10769986231163320> and demonstrated using mental health data in Liang et al. (in press; annotated R code and data utilized in this example are available in Mendeley data) <doi:10.17632/kpjp3gnwbt.1>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Imports GDINA,ggplot2, ggpubr, ggsignif

LazyData true

NeedsCompilation no

Author Qianru Liang [aut, cre, cph] (<https://orcid.org/0000-0001-7558-0431>), Jimmy de la Torre [aut] (<https://orcid.org/0000-0002-0893-3863>), Jingping Du [ctb]

Maintainer Qianru Liang <liangqr@jnu.edu.cn>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2024-05-15 20:00:02 UTC

Contents

`cep` *Data Set cep*

Description

The classification error probabilities (CEP) can be obtained in this data example.

Usage

```
cep
```

Format

A list containing:

`cep.matrix` Each of the 4 lists includes two 2 by 2 matrices.

`w` Each of the 4 lists includes two 2005 by 2 matrices.

`mp` Each of the 4 lists includes two 2005 by 4 matrices.

`eap` Each of the 4 lists includes two 2005 by 4 matrices.

`CEP_t` *Compute classification error probabilities for attributes at different time points*

Description

Function to compute classification error probabilities (CEP) for attributes at different time points. Only attribute-level CEP is available for the time being.

Usage

```
CEP_t(fit.object, t, K, N)
```

Arguments

`fit.object`	a list of the G-DINA model objects return from `GDINA` R package at pre-and post-tests.
`t`	the number of time points. This package can only handle two time points can for the time being.
`K`	the number of attributes.
`N`	the number of examinees (observations).

Value

a list with elements

cep.matrix the CEP matrix

w the correction weights

mp the estimated marginal posterior probabilities obtained from `GDINA` R package

eap the estimated EAP of attribute profiles obtained from `GDINA` R package

References

- Liang, Q., de la Torre, J., & Law, N. (2023).Latent transition cognitive diagnosis model with covariates: A three-step approach. *Journal of Educational and Behavioral Statistics*. doi:10.3102/10769986231163320
- Huebner, A., & Wang, C. (2011).A note on comparing examinee classification methods for cognitive diagnosis models. *Educational and Psychological Measurement, 71*, 407-419. doi:10.1177/0013164410388832

Examples

```
if(requireNamespace("GDINA")){
library(GDINA)
# Assuming dat0, dat1, Q, and other necessary data and objects are predefined.
rdmodel <- c("GDINA","GDINA","GDINA","GDINA","GDINA","GDINA","GDINA","GDINA",
"GDINA","GDINA","GDINA","GDINA","GDINA","GDINA","GDINA","GDINA",
"GDINA","GDINA","GDINA","GDINA","RRUM","GDINA","GDINA","GDINA",
"GDINA","LLM","LLM","RRUM","ACDM","GDINA","GDINA","GDINA",
"GDINA","GDINA","GDINA","GDINA","GDINA","GDINA","GDINA","GDINA")
fitrd <- GDINA(dat = dat0, Q = Q, model= rdmodel, mono.constraint = TRUE, verbose=0)

# Obtained the item parameters from Tan et al. (2022)
itemparm.rd = GDINA::extract(fitrd,"catprob.parm")

# Fit the response data at pre-test to the selected models
fit.t1 = GDINA(dat = dat1[,3:42], Q = Q, mono.constraint = TRUE, model = rdmodel,
catprob.parm = itemparm.rd, att.dist = "independent", control=list(maxitr = 0), verbose=0)

# Fit the response data at post-test to the selected models
fit.t2 = GDINA(dat = dat1[,43:82], Q = Q, mono.constraint = TRUE, model = rdmodel,
catprob.parm = itemparm.rd, att.dist = "independent", control=list(maxitr = 0), verbose=0)
fit.object = list()
fit.object[[1]] <- fit.t1
fit.object[[2]] <- fit.t2
t = 2 # the number of time points
K = ncol(Q) # the number of attributes
N = nrow(dat1) # the number of observations

cep = CEP_t(fit.object = fit.object, t = t, K = K, N = N)

# The CEP matrices of the attributes
cep$cep.matrix

}
```

---

dat0 *Data Set dat0*

Description

The dataset of time point 1 used in (Tan et al., 2023).

Usage

```
dat0
```

Format

A data frame with 719 rows and 40 columns.

dat1	*Data Set dat1*

Description

The longitudinal dataset used in this example. Items with a prefix "a" are for the pre-test, and items with a prefix "b" are for the post-test.

Usage

```
dat1
```

Format

A data frame with 2005 rows and 82 columns.

L_step3	*Step 3 estimation for latent logistic regression coefficients*

Description

Function to estimate the latent logistic regression models at the initial state and transition

Usage

```
L_step3(par, z_t1, z_t2, weight, k)
```

Arguments

par	Coefficients of latent logistic regression to be estimated.
z_t1	covariates at Time 1, which has already had the intercept column (1s).
z_t2	covariates at Time 2, which has already had the intercept column (1s).
weight	Correction weight obtained from CEP.
k	The k-th attribute.

Value

log-likelihood value.

References

Liang, Q., de la Torre, J., & Law, N. (2023). Latent transition cognitive diagnosis model with covariates: A three-step approach. *Journal of Educational and Behavioral Statistics*.doi:10.3102/10769986231163320

Q — *Data Set Q*

Description

The Q-matrix empirically validated by Tan et al.(2023).

Usage

```
Q
```

Format

A data frame with 40 rows and 4 columns.

step3.est — *Step 3 estimation for latent logistic regression coefficients*

Description

Step 3 estimation for latent logistic regression coefficients

Usage

```
step3.est(
  cep,
  z_t1,
  z_t2,
  K,
  t,
  beta_in = matrix(0, ncol(z_t1), K),
  ga01_in = matrix(0, ncol(z_t2), K),
  ga10_in = matrix(0, ncol(z_t2))
)
```

Arguments

`cep`	estimated classification error probabilities returned from `CEP_t`. The uncorrected attribute profile (EAP) is also stored in this object.
`z_t1`	covariates at Time 1, which has already had the intercept column (1s).
`z_t2`	covariates at Time 2, which has already had the intercept column (1s).
`K`	the number of attributes.
`t`	the number of time points. This package can only handle two time points can for the time being.

`beta_in` the initial values for the regression coefficients at Time 1 (initial state). Default are 0s.

`ga01_in` the initial values for the regression coefficients of transition from absence (0) to presence (1) at Time 2. Default are 0s.

`ga10_in` the initial values for the regression coefficients of transition from presence (1) to absence (0) at Time 2. Default are 0s.

Value

a list with elements

beta A data frame with 2 rows and 4 columns, representing the estimated regression coefficients at Time 1 (initial state)

gamma_01 A data frame with 4 rows and 4 columns, representing the estimated regression coefficients of transition from absence (0) to presence (1) at Time 2

gamma_10 A data frame with 4 rows and 4 columns, representing the estimated regression coefficients of transition from absence (0) to presence (1) at Time 2

result A data frame with dimensions 40 by 9, containing the results of the estimation, including all regression coefficients and the corresponding odds ratios, Cohen's d, standard errors (SE), 95% confidence intervals, and p-values.

Examples

```
t = 2 # the number of time points
K = ncol(Q) # the number of attributes
z_t1_test = matrix(sample(c(0, 1), size = 20, replace = TRUE), nrow = 10)
z_t2_test = matrix(sample(c(0, 1), size = 40, replace = TRUE), nrow = 10)
# Set appropriate initial values of the coefficients
# Initial values of initial state's regression coefficients
beta_in = matrix(0, ncol(z_t1_test), K)

# Initial values of transition probability's regression coefficients
# These were computed using the raw data.
# When Gender coding is 1 = male, 0 = female:
ga01_in = cbind(c(-2.15, 0.56, 0.09, -0.79),
                c(-1.6, 0.05, -0.01, -0.38),
                c(-1.25, 0.06, -0.25, 0.14),
                c(-1.18, -0.26, 0.04, 0.37))
                #initial values of regression coefficients (for transition from 0 to 1)
ga10_in = cbind(c(-0.84, -0.18, -0.14, 0.23),
                c(-0.18, 0.49, 0.44, -0.35),
                c(-0.22, 0.18, 0.37, -0.45),
                c(-0.49, 0.10, 0.43, 0.20))
cep_test = list()
cep_test[["mp"]][[1]] = matrix(runif(40,min = 0,max=1),nrow = 10)
cep_test[["mp"]][[2]] = matrix(runif(40,min = 0,max=1),nrow = 10)
cep_test[["eap"]][[1]] = matrix(runif(40,min = 0,max=1),nrow = 10)
cep_test[["eap"]][[2]] = matrix(runif(40,min = 0,max=1),nrow = 10)
for (i in 1:4){
cep_test[["cep_matrix"]][[i]]=list()
```

```
cep_test[["w"]][[i]]=list()
for (k in 1:2) {
  cep_test[["cep_matrix"]][[i]][[k]]=matrix(c(1,0.02,0.06,1),nrow = 2)
      cep_test[["w"]][[i]][[k]] = matrix(runif(20,min = 0,max=1),nrow = 10)
        }
       }
step3.output_test <- step3.est(cep = cep_test, z_t1 = z_t1_test, z_t2 = z_t2_test,
 K = K, t = t, beta_in, ga01_in, ga10_in)
## Not run:
# The run is dependent on the output of the CEP_t() function
# And the process time takes more than 5s.
# It is not recommended to run it.
# Covariates
Z = dat1[, c(1,2)] # use intervention and gender as covariates
z_t1 = cbind(1, Z$gender)  # Covariate at time 1
z_t2 = cbind(1, Z$gender, Z$intervention, apply(Z,1,prod)) # Covariates at time 2
colnames(z_t1) <- c("intercept", "gender")
colnames(z_t2) <- c("intercept", "gender", "intervention", "intervention_gender")

t = 2 # the number of time points
K = ncol(Q) # the number of attributes

# Set appropriate initial values of the coefficients
# Initial values of initial state's regression coefficients
beta_in = matrix(0, ncol(z_t1), K)

# Initial values of transition probability's regression coefficients
# These were computed using the raw data.
# When Gender coding is 1 = male, 0 = female:
ga01_in = cbind(c(-2.15, 0.56, 0.09, -0.79),
                c(-1.6, 0.05, -0.01, -0.38),
                c(-1.25, 0.06, -0.25, 0.14),
                c(-1.18, -0.26, 0.04, 0.37))
                #initial values of regression coefficients (for transition from 0 to 1)
ga10_in = cbind(c(-0.84, -0.18, -0.14, 0.23),
                c(-0.18, 0.49, 0.44, -0.35),
                c(-0.22, 0.18, 0.37, -0.45),
                c(-0.49, 0.10, 0.43, 0.20))
                #initial values of regression coefficients (for transition from 1 to 0)
# Step 3 estimation (This will take a few minutes)
step3.output <- step3.est(cep = cep, z_t1 = z_t1, z_t2 = z_t2, K = K,
                         t = t, beta_in = beta_in, ga01_in = ga01_in, ga10_in = ga10_in)

# Obtain estimation results
step3.output$result

# Latent logistic regression coefficients
beta = step3.output$beta
gamma_01 = step3.output$gamma_01
gamma_10 = step3.output$gamma_10

## End(Not run)
```

step3.output   *Data Set step3.output*

Description

The output of step 3 estimation can be obtained in this data example.

Usage

```
step3.output
```

Format

A list containing:

`beta` A data frame with 2 rows and 4 columns.

`gamma_01` A data frame with 4 rows and 4 columns.

`gamma_10` A data frame with 4 rows and 4 columns.

`result` A list containing the results of the estimation, with dimensions 40 by 9.

trans.matrix   *Compute transition matrix*

Description

Function to compute transition matrix using classification results

Usage

```
trans.matrix(X)
```

Arguments

`X`	a matrix containing the initial state (first column) and the transition state (second column).

Value

a 2×2 matrix where rows represent initial states (0 and 1) and the columns represent transition states (0 and 1).

Examples

```
initial_states <- c(1, 2, 1, 2)
final_states <- c(1, 1, 2, 2)
transition_matrix <- trans.matrix(data.frame(initial_states, final_states))
print(transition_matrix)
## Not run:
# transition probabilities (corrected and updated)
t = 2 # the number of time points
K = ncol(Q) # the number of attributes
Z = dat1[, c(1,2)]
z_t1 = cbind(1, Z$gender)  # Covariate at time 1
z_t2 = cbind(1, Z$gender, Z$intervention, apply(Z,1,prod)) # Covariates at time 2
beta = step3.output$beta
gamma_01 = step3.output$gamma_01
gamma_10 = step3.output$gamma_10
updated.class <- update.class(cep = cep, K = K, t = t, z_t1 = z_t1,
                       z_t2 = z_t2, beta = beta, gamma_01 = gamma_01, gamma_10 = gamma_10)
C.eap.t1 = updated.class$cor.profile[[1]]
C.eap.t2 = updated.class$cor.profile[[2]]
C.eap.t1t2 <- cbind(z_t2, C.eap.t1, C.eap.t2)
t.A1.c = trans.matrix(as.matrix(C.eap.t1t2[,c("A1_t1","A1_t2")]))
t.A1.c

## End(Not run)
```

`update_class` *Classification update using the Bayes' Theorem*

Description

Function to update classifications (attribute profiles) using the Bayes' Theorem

Usage

```
update_class(cep, K, t, z_t1, z_t2, beta, gamma_01, gamma_10)
```

Arguments

`cep`	estimated classification error probabilities returned from `CEP_t`. The uncorrected attribute profile (EAP) is also stored in this object.
`K`	the number of attributes.
`t`	the number of time points. This package can only handle two time points can for the time being.
`z_t1`	covariates at Time 1, which has already had the intercept column (1s).
`z_t2`	covariates at Time 2, which has already had the intercept column (1s).
`beta`	the estimated regression coefficients at Time 1 (initial state)

gamma_01 the estimated regression coefficients of transition from absence (0) to presence (1) at Time 2

gamma_10 the estimated regression coefficients of transition from absence (0) to presence (1) at Time 2

Value

a list with elements

post.prob the corrected and updated posterior probability

att.prevalance the corrected and updated attribute prevalance

cor.profile the corrected and updated attribute profiles for different time points

References

Liang, Q., de la Torre, J., & Law, N. (2023). Latent transition cognitive diagnosis model with covariates: A three-step approach. *Journal of Educational and Behavioral Statistics*.doi:10.3102/10769986231163320

Examples

```
## Not run:
#The run is dependent on the output of the step3.est() function and CEP_t() function
#It is not recommended for run it.
t = 2 # the number of time points
K = ncol(Q) # the number of attributes
Z = dat1[, c(1,2)]
z_t1 = cbind(1, Z$gender)  # Covariate at time 1
z_t2 = cbind(1, Z$gender, Z$intervention, apply(Z,1,prod)) # Covariates at time 2
beta = step3.output$beta
gamma_01 = step3.output$gamma_01
gamma_10 = step3.output$gamma_10

# Update classifications using the Bayes' Theorem
updated.class <- update.class(cep = cep, K = K, t = t, z_t1 = z_t1, z_t2 = z_t2,
                              beta = beta, gamma_01 = gamma_01, gamma_10 = gamma_10)

# The corrected and updated attribute prevalance
updated.class$att.prevalance

# The corrected and updated posterior probability
updated.class$post.prob

## End(Not run)
```

Index