Package ‘TDCM’

May 7, 2026

Title The Transition Diagnostic Classification Model Framework

Version 0.3.0

Description Estimate the transition diagnostic classification model (TDCM) described in Madison & Bradshaw (2018) <doi:10.1007/s11336-018-9638-5>, a longitudinal extension of the log-linear cognitive diagnosis model (LCDM) in Henson, Templin & Willse (2009) <doi:10.1007/s11336-008-9089-5>. As the LCDM subsumes many other diagnostic classification models (DCMs), many other DCMs can be estimated longitudinally via the TDCM. The 'TDCM' package includes functions to estimate the single-group and multigroup TDCM, summarize results of interest including item parameters, growth proportions, transition probabilities, transitional reliability, attribute correlations, model fit, and growth plots.

License MIT + file LICENSE

URL `https://cotterell.github.io/tdcm/`, `https://github.com/cotterell/tdcm`

BugReports `https://github.com/cotterell/tdcm/issues`

Encoding UTF-8

Language en-US

LazyData true

Depends R (>= 4.2.0)

Suggests devtools, knitr, lintr, MASS, rmarkdown, ROI, roxygen2, rsconnect, styler (>= 1.10.2), testthat (>= 3.0.0), tinytex, V8

Config/testthat/edition 3

Config/Needs/website pkgdown

SystemRequirements pandoc

VignetteBuilder knitr

RoxygenNote 7.3.3

Imports CDM, gtools, polycor, stringr

NeedsCompilation no

Author Matthew J. Madison [aut, cph] (ORCID:
<https://orcid.org/0000-0002-2944-7442>),
Sergio Haab [aut],
Minjeong Jeon [aut, cph],
Constanza Mardones-Segovia [aut, cph],
Michael E. Cotterell [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-4302-0891>),
University of Georgia [cph] (The content and opinions expressed in this material do not necessarily reflect the views of nor are they endorsed by the University of Georgia or the University System of Georgia.),
Institute of Education Sciences [fnd] (This work is supported by the U.S. Department of Education Institute of Education Sciences under IES Award Number R305D220020.),
National Science Foundation [fnd] (This work is supported by the National Science Foundation under NSF Award Number 1921373.)

Maintainer Michael E. Cotterell <mepcott@uga.edu>

Repository CRAN

Date/Publication 2026-04-22 21:20:02 UTC

Contents

`data.tdcm` *Several data sets for the* **TDCM** *package.*

Description

Several data sets for the **TDCM** package.

Usage

```
data.tdcm01

data.tdcm02

data.tdcm03

data.tdcm04

data.tdcm05
```

Format

`data.tdcm01` is a simulated dataset with two time points, four attributes, twenty items, one group of size 1000, and a single Q-matrix. The format is a list of two:

- `data`: a data frame of binary item responses
- `q.matrix`: a data frame specifying the Q-matrix

`data.tdcm02` is a simulated dataset with three time points, two attributes, ten items, one group of size 2500, and a single Q-matrix. The format is a list of two:

- `data`: a data frame of binary item responses
- `q.matrix`: a data frame specifying the Q-matrix

`data.tdcm03` is a simulated dataset with three time points, two attributes, one group of size 1500, and three different ten-item Q-matrices for each time point. Anchor items are specified as items 1/11/21 and items 14/24. The format is a list of five:

- `data`: a data frame of binary item responses
- `q.matrix.1`: a data frame specifying the Q-matrix for the first time point
- `q.matrix.2`: a data frame specifying the Q-matrix for the second time point
- `q.matrix.3`: a data frame specifying the Q-matrix for the third time point
- `q.matrix.stacked`: data frame specifying the combined Q-matrix for all time points

`data.tdcm04` is a simulated dataset with two time points, four attributes, twenty items, two group of size 800 and 900, respectively, and a single Q-matrix. The format is a list of three:

- `data`: a data frame of binary item responses
- `q.matrix`: a data frame specifying the Q-matrix
- `groups`: a vector specifying the examinee group membership

`data.tdcm05` is a simulated dataset with one time point, four attributes, and twenty items. For use with the 1-PLCDM. The format is a list of two:

- `data`: a data frame of binary item responses
- `q.matrix`: a data frame specifying the Q-matrix

Examples

```
#############################
## Example 1: T = 2, A = 4 ##
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix
model <- TDCM::tdcm(data, q.matrix, num.time.points = 2)
results <- TDCM::tdcm.summary(model)
results$item.parameters
results$growth.effects

#############################
## Example 3: T = 3, A = 2 ##
data <- data.tdcm03$data
q1 <- data.tdcm03$q.matrix.1
q2 <- data.tdcm03$q.matrix.2
q3 <- data.tdcm03$q.matrix.3
q <- data.tdcm03$q.matrix.stacked

#TDCM with anchor items constrained
m1 <- tdcm(data, q, num.time.points = 3, num.q.matrix = 3,
anchor = c(1,11,1,21,14,24), num.items = c(10,10,10))

#TDCM without anchor items
m2 <- tdcm(data, q, num.time.points = 3, num.q.matrix = 3, num.items = c(10,10,10))

#Compare models to assess measurement invariance
tdcm.compare(m1, m2)

#Summarize model 1
r1 = tdcm.summary(m1, transition.option = 3)
r1$item.parameters
r1$growth
r1$growth.effects
```

`item.influence` *Estimating item influence measures.*

Description

Function to estimate estimate item influence measures. Code adapted from (Jurich & Madison, 2023). This function is not available for longitudinal DCMs.

Usage

```
item.influence(model, data, fullcorrelation = FALSE, progress = TRUE)
```

Arguments

- `model` — a previously calibrated model; an object of class `gdina`.
- `data` — a required $N \times I$ matrix. Binary item responses are in the columns.
- `fullcorrelation` — optional logical argument indicating a full or reduced response-classification correlation matrix.
- `progress` — An optional logical indicating whether the function should print the progress of estimation.

Details

For DCMs, item influence quantifies how much an item impacts classifications. Given an estimated DCM and item response data, this function estimates five item influence measures, including item pull, item override, proportion of attribute information, response-classification correlation (corr1), and response-posterior correlation (corr2).

Value

A list containing several item influence measures.

Note

Currently, this function currently only runs on DCMs estimated at a single time point. It will not run properly for TDCM objects.

References

Jurich, D. & Madison, M. J. (2023). Measuring item influence for diagnostic classification models. *Educational Assessment*.

Examples

```
## Item influence illustration
#load data (simulated based on Jurich and Bradshaw (2014))
qmatrix <- CDM::data.sda6$q.matrix
responses <- CDM::data.sda6$data

#Estimate the full LCDM
model1 <- CDM::gdina(responses, qmatrix, linkfct = "logit", method = "ML")

#Estimate item influence measures
influence <- TDCM::item.influence(model1, responses)

#Summarize influence statistics
influence$Pull #item pull
influence$Override #item override
influence$Information #proportion of attribute information
influence$Correlation1 #correlation of responses and classifications
influence$Correlation2 #correlation of responses and posterior probabilities
```

mg.tdcm *Estimating the multigroup transition diagnostic classification model (TDCM)*

Description

mg.tdcm() estimates the Transition Diagnostic Classification Model for scenarios involving multiple groups (e.g., control and treatment group; Madison & Bradshaw, 2018b). Similar to tdcm(), this function supports the estimation of various DCMs by allowing different rule specifications via the rule option and link functions via the linkfct option,with LCDM as the default rule and link function. The rule can be modified to estimate the DINA model, DINO model, CRUM (i.e., ACDM, or main effects model), or reduced interaction versions of the LCDM. Additionally, the link function can be adjusted to specify the GDINA model.

Usage

```
mg.tdcm(
  data,
  q.matrix,
  num.time.points,
  rule = "LCDM",
  linkfct = "logit",
  groups,
  forget.att = c(),
  group.invariance = TRUE,
  time.invariance = TRUE,
  progress = TRUE
)
```

Arguments

data
: A required $N \times T \times I$ matrix or data.frame where rows correspond to N examinees and columns represent the binary item responses across T time points and I items.

q.matrix
: A required $I \times A$ matrix indicating which items measure which attributes. Currently, the function only accepts a single Q-matrix.

num.time.points
: A required integer ≥ 2 specifying the number of time points (i.e., measurement occasions).

rule
: A string or a vector indicating the specific DCM to be employed. A vector of supported rule values is provided by tdcm.rules. Currently accepted values are: "LCDM", "DINA", "DINO", "CRUM", "RRUM", "LCDM1" for the LCDM with only main effects, "LCDM2" for the LCDM with two-way interactions,

"LCDM3", and so on. If `rule` is supplied as a single string, then that DCM will be assumed for each item. If entered as a vector, a rule can be specified for each item. The vector must have length equal to the total number of items across all time points.

`linkfct` A `string` or a `vector` indicating the LCDM link function. Currently accepts "logit" (default) to estimate the LCDM, "identity" to estimate the GDINA model, and "log" link function to estimate the reduced reparameterized unified model (RRUM). The vector must have length equal to the total number of items across all time points.

`groups` A required `vector` of integer group identifiers for multiple group estimation.

`forget.att` An optional `vector` allowing for constraining of individual attribute proficiency loss, or forgetting. The default allows forgetting for each measured attribute (e.g., $P(1 \to 0) \neq 0$). See tdcm for more detailed information.

`group.invariance`
logical argument. If `TRUE` (default), item parameters are assumed to be equal for all groups. If `FALSE`, item parameters are not assumed to be equal for all groups.

`time.invariance`
logical argument. If `TRUE` (default), item parameters are assumed to be equal for all time points. If `FALSE`, item parameters are not assumed to be equal for all time points.

`progress` logical argument. If `FALSE`, the function will print the progress of estimation. If `TRUE` (default), no progress information is printed.

Details

Multigroup Transition Diagnostic Classification Model (Multigroup TDCM)

Multigroup TDCM is a confirmatory latent transition model that measures examinees' growth or decline in attribute mastery over time among groups (Madison & Bradshaw, 2018b). In this model, the probability of the item response vector X_e is conditioned on observed groups membership G:

$$P(X_e = x_e|G = g) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C}\cdots\sum_{c_T=1}^{C} v_{c_1|g}\tau_{c_2|c_1,g}\tau_{c_3|c_2,g}\cdots\tau_{c_T|c_{T-1},g}\prod_{t=1}^{T}\prod_{i=1}^{I}\pi_{ic_{t,g}}^{x_{eit}}(1-\pi_{ic_{t,g}})^{1-x_{eit}},$$

where:

- $v_{c_1|g}$ represents the probability of belonging to attribute profile c at time 1 given the observed group g.
- $\tau_{c_t|c_{T-1},g}$ represents the probability of transitioning attribute profiles from time point $t-1$ to time point t.
- $\pi_{ic_{t,g}}$ is the item response function, which models the probability of answering item i correctly at time t given attribute profile c and observed group g.

Therefore, if the study purpose is to assess growth between a treatment and control group in a pre- and post-test design, the probability of the item response vector reduces to:

$$P(X_e = x_e|G = g) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C} v_{c_1|g}\tau_{c_2|c_{1,g}} \prod_{t=1}^{2}\prod_{i=1}^{I} \pi_{ic_{t,g}}^{x_{eit}}(1-\pi_{ic_{t,g}})^{1-x_{eit}}.$$

Accounting for Measurement Invariance

Measurement invariance indicates whether the **item response function** remains constant over time points (**time invariance**) or across groups (**group invariance**). Note that regardless of the assumed invariance, attribute mastery transitions can still be compared across time and groups.

Depending on the assumed constrained, one of the four measurement invariance conditions can be applied:

Consider an experiment design with a treatment and control group.

a) No time Invariance across time or group invariance assumed:

If neither time nor group invariance is assumed, each item has a different response function over time and across groups. Thus, the probability of the item response function remains unchanged.

$$P(X_e = x_e|G = g) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C} v_{c_1|g}\tau_{c_2|c_1,g} \prod_{t=1}^{2}\prod_{i=1}^{I} \pi_{ic_{t,g}}^{x_{eit}}(1-\pi_{ic_{t,g}})^{1-x_{eit}}.$$

b) No time Invariance across time assumed but group invariance assumed:

If time invariance is not assumed but group invariance is, each item has a different response function over time. Thus, the probability of the item response function only depends on t.

$$P(X_e = x_e|G = g) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C} v_{c_1|g}\tau_{c_2|c_1,g} \prod_{t=1}^{2}\prod_{i=1}^{I} \pi_{ic_t}^{x_{eit}}(1-\pi_{ic_t})^{1-x_{eit}}.$$

c) Time Invariance across time assumed but not group invariance:

If time invariance is assumed but group invariance is not, each item has a different response function across groups. Thus, the probability of the item response function only depends on g.

$$P(X_e = x_e|G = g) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C} v_{c_1|g}\tau_{c_2|c_1,g} \prod_{t=1}^{2}\prod_{i=1}^{I} \pi_{ic_g}^{x_{eit}}(1-\pi_{ic_g})^{1-x_{eit}}.$$

d) Time Invariance across time and group invariance assumed:

Finally, when time and group invariance are assumed, each item has the same item response function over time and groups, reducing the measurement model to an LCDM.

$$P(X_e = x_e|G = g) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C} v_{c_1|g}\tau_{c_2|c_{1,g}} \prod_{t=1}^{2}\prod_{i=1}^{I} \pi_{ic}^{x_{eit}}(1-\pi_{ic})^{1-x_{eit}}.$$

Value

An object of class `gdina` with entries as indicated in the **CDM** package. For the TDCM-specific results (e.g., growth, transitions), use `TDCM::mg.tdcm.summary()`.

References

de la Torre, J. (2011). The Generalized DINA model framework. *Psychometrika, 76*, 179–199. doi:10.1007/s11336-011-9207-7.

George, A. C., Robitzsch, A., Kiefer, T., Gross, J., & Ünlü , A. (2016). The *R* package **CDM** for cognitive diagnosis models. *Journal of Statistical Software,74*(2), 1-24. doi:10.18637/jss.v074.i02

Henson, R., Templin, J., & Willse, J. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika, 74*, 191-21. doi:10.1007/s11336-008-9089-5.

Johnson, M. S., & Sinharay, S. (2020). The reliability of the posterior probability of skill attainment in diagnostic classification models. *Journal of Educational Measurement, 47*(1), 5–31. doi:10.3102/1076998619864550.

Kaya, Y., & Leite, W. (2017). Assessing change in latent skills across time With longitudinal cognitive diagnosis modeling: An evaluation of model performance. *Educational and Psychological Measurement, 77*(3), 369–388. doi:10.1177/0013164416659314.

Li, F., Cohen, A., Bottge, B., & Templin, J. (2015). A latent transition analysis model for assessing change in cognitive skills. *Educational and Psychological Measurement, 76*(2), 181–204. doi:10.1177/0013164415588946.

Madison, M. J. (2019). Reliably assessing growth with longitudinal diagnostic classification models. *Educational Measurement: Issues and Practice, 38*(2), 68-78. doi:10.1111/emip.12243.

Madison, M. J., & Bradshaw, L. (2018a). Assessing growth in a diagnostic classification model framework. *Psychometrika, 83*(4), 963-990. doi:10.1007/s11336-018-9638-5.

Madison, M. J., & Bradshaw, L. (2018b). Evaluating intervention effects in a diagnostic classification model framework. *Journal of Educational Measurement, 55*(1), 32-51. doi:10.1111/jedm.12162.

Madison, M.J., Chung, S., Kim, J., & Bradshaw, L.P. (2024) Approaches to estimating longitudinal diagnostic classification models. *Behaviormetrika, 51*(7), 7-19. doi:10.1007/s41237-023-00202-5.

Madison, M. J., Jeon, M., Cotterell, M., Haab, S., & Zor, S. (2025). TDCM: An R package for estimating longitudinal diagnostic classification models. *Multivariate Behavioral Research*, 1–10. doi:10.1080/00273171.2025.2453454.

Ravand, H., Effatpanah, F., Kunina-Habenicht, O., & Madison, M. J. (2025). A didactic illustration of writing skill growth through a longitudinal diagnostic classification model. *Frontiers in Psychology*. doi:10.3389/fpsyg.2024.1521808.

Rupp, A. A., Templin, J., & Henson, R. (2010). *Diagnostic Measurement: Theory, Methods, and Applications*. New York: Guilford. ISBN: 9781606235430.

Schellman, M., & Madison, M. J. (2024). Estimating the reliability of skill transition in longitudinal DCMs. *Journal of Educational and Behavioral Statistics*.doi:10.3102/10769986241256032.

Templin, J., & Bradshaw, L. (2013). Measuring the reliability of diagnostic classification model examinee estimates. *Journal of Classification, 30*, 251-275. doi:10.1007/s00357-013-9129-4.

Wang. S., Yang. Y., Culpepper, S. A., & Douglas, J. (2018). Tracking skill acquisition with cognitive diagnosis models: A higher-order, hidden Markov model With covariates. *Journal of Educational and Behavioral Statistics, 43*(1), 57-87. doi:10.3102/1076998617719727.

Examples

```
############################################################################
# Example 1: Multigroup TDCM without assuming time or group invariance
############################################################################

# Load data: G = 2, T = 2, A = 4, I = 20
data(data.tdcm04, package = "TDCM")
data <- data.tdcm04$data
q.matrix <- data.tdcm04$q.matrix
groups <- data.tdcm04$groups

# Estimate model
mg.model1 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           groups = groups, time.invariance = FALSE,
                           group.invariance = FALSE)

# Summarize results
results1 <- TDCM::mg.tdcm.summary(mg.model1)
results1$item.parameters

## In this case, neither time nor group invariance is assumed,
## meaning that item parameters are estimated separately for
## each group and time point. This allows item functioning to vary both
## across groups and over time.

# , , Group 1
#
#            l0     l1,1  l1,2  l1,3  l1,4  l2,12  l2,13 l2,14 l2,23 l2,24 l2,34
#   Item 1  -2.05  2.594   --    --    --    --     --    --    --    --    --
#   Item 2  -2.041 2.47    --    --    --    --     --    --    --    --    --
#   Item 3  -1.849 2.282   --    --    --    --     --    --    --    --    --
#   Item 4  -2.168 2.008 1.783   --    -- -0.068   --    --    --    --    --
#   Item 5  -2.021 1.827   --  1.061   --    --   0.699  --    --    --    --
#   Item 6  -2.118   --  2.515   --    --    --     --    --    --    --    --
#   Item 7  -1.835   --  2.535   --    --    --     --    --    --    --    --
#   Item 8  -1.987   --  2.512   --    --    --     --    --    --    --    --
#   Item 9  -2.219   --  2.032 1.916   --    --     --    --  0.039   --    --
#   Item 10 -2.119   --  1.263   --  1.717   --     --    --    --  1.355   --
#   Item 11 -1.984   --    --  2.422   --    --     --    --    --    --    --
#   Item 12 -2.511   --    --  2.858   --    --     --    --    --    --    --
#   Item 13 -2.108   --    --  2.245   --    --     --    --    --    --    --
#   Item 14 -1.914   --    --  0.346 0.977   --     --    --    --    --  2.097
#   Item 15 -2.148 1.678   --  2.224   --    --   0.583  --    --    --    --
#   Item 16 -2.039   --    --    --  2.416   --     --    --    --    --    --
#   Item 17 -2.439   --    --    --  3.186   --     --    --    --    --    --
#   Item 18 -2.056   --    --    --  2.643   --     --    --    --    --    --
#   Item 19 -1.926 1.293   --    --  1.068   --     --  1.461   --    --    --
#   Item 20 -2.227   --  1.882   --  1.749   --     --    --    --  0.208   --
#   Item 21 -1.797 2.202   --    --    --    --     --    --    --    --    --
#   Item 22 -1.959 2.405   --    --    --    --     --    --    --    --    --
#   Item 23 -2.454 2.804   --    --    --    --     --    --    --    --    --
#   Item 24 -2.353 1.785 1.909   --    --  0.789    --    --    --    --    --
```

```
#   Item 25 -2.313 1.237   --  2.041   --    --   1.354   --    --    --    --
#   Item 26 -1.836   --  2.349   --    --    --      --   --    --    --    --
#   Item 27 -1.951   --  2.555   --    --    --      --   --    --    --    --
#   Item 28 -1.949   --  2.487   --    --    --      --   --    --    --    --
#   Item 29 -1.96    --  1.632 1.775   --    --      --   --  0.71    --    --
#   Item 30 -2.286   --  1.949   --  1.973   --      --   --    --  0.361   --
#   Item 31 -1.794   --    --  2.466   --    --      --   --    --    --    --
#   Item 32 -1.886   --    --  2.574   --    --      --   --    --    --    --
#   Item 33 -1.516   --    --  1.969   --    --      --   --    --    --    --
#   Item 34 -2.066   --    --  1.307 1.667   --      --   --    --    --  1.191
#   Item 35 -2.329 2.013   --  1.643   --    --   0.891   --    --    --    --
#   Item 36 -2.577   --    --    --  3.058   --      --   --    --    --    --
#   Item 37 -2.028   --    --    --  2.627   --      --   --    --    --    --
#   Item 38 -1.889   --    --    --  2.206   --      --   --    --    --    --
#   Item 39 -1.982 1.678   --    --  1.374   --      --  0.825  --    --    --
#   Item 40 -2.19    --  2.483   --  1.572   --      --   --    --  0.602   --
#
#   , , Group 2
#
#            l0    l1,1  l1,2  l1,3  l1,4  l2,12 l2,13 l2,14 l2,23 l2,24 l2,34
#   Item 1  -1.932 2.385   --    --    --    --    --    --    --    --    --
#   Item 2  -1.921 2.347   --    --    --    --    --    --    --    --    --
#   Item 3  -2.055 2.388   --    --    --    --    --    --    --    --    --
#   Item 4  -2.112 2.102 1.31    --    --  0.889   --    --    --    --    --
#   Item 5  -2.027 1.556   --  0.995   --    --  1.578   --    --    --    --
#   Item 6  -1.893   --  2.523   --    --    --    --    --    --    --    --
#   Item 7  -1.871   --  2.496   --    --    --    --    --    --    --    --
#   Item 8  -2.053   --  2.392   --    --    --    --    --    --    --    --
#   Item 9  -2.062   --  1.69  1.373   --    --    --    --  1.149   --    --
#   Item 10 -1.997   --  1.478   --  1.222   --    --    --    --  1.271   --
#   Item 11 -2.053   --    --  2.467   --    --    --    --    --    --    --
#   Item 12 -2.006   --    --  2.547   --    --    --    --    --    --    --
#   Item 13 -2.003   --    --  2.658   --    --    --    --    --    --    --
#   Item 14 -2.056   --    --  3.463 1.336   --    --    --    --    --  -1.045
#   Item 15 -2.051 1.423   --  1.399   --    --  1.317   --    --    --    --
#   Item 16 -2.182   --    --    --  2.729   --    --    --    --    --    --
#   Item 17 -2.289   --    --    --  2.932   --    --    --    --    --    --
#   Item 18 -2.266   --    --    --  2.768   --    --    --    --    --    --
#   Item 19 -2.227 1.408   --    --  1.542   --    --  1.128   --    --    --
#   Item 20 -1.898   --  1.182   --  1.453   --    --    --    --  1.319   --
#   Item 21 -1.757 2.365   --    --    --    --    --    --    --    --    --
#   Item 22 -2.455 3.151   --    --    --    --    --    --    --    --    --
#   Item 23 -2.393 2.99    --    --    --    --    --    --    --    --    --
#   Item 24 -1.626 1.061 1.029   --    --  1.447   --    --    --    --    --
#   Item 25 -1.96  1.027   --  1.274   --    --  1.765   --    --    --    --
#   Item 26 -1.723   --  2.266   --    --    --    --    --    --    --    --
#   Item 27 -1.766   --  2.177   --    --    --    --    --    --    --    --
#   Item 28 -1.908   --  2.338   --    --    --    --    --    --    --    --
#   Item 29 -2.22    --  1.947 1.341   --    --    --    --  1.014   --    --
#   Item 30 -1.967   --  1.124   --  1.527   --    --    --    --  1.036   --
#   Item 31 -2.011   --    --  2.503   --    --    --    --    --    --    --
#   Item 32 -2.6     --    --  3.072   --    --    --    --    --    --    --
#   Item 33 -2.027   --    --  2.652   --    --    --    --    --    --    --
```

```
#   Item 34 -1.663   --    --  0.832 0.805   --    --    --    --    --  1.816
#   Item 35 -2.037 1.656   --  1.554   --    --  0.601   --    --    --    --
#   Item 36 -2.022   --    --    --  2.431   --    --    --    --    --    --
#   Item 37 -2.866   --    --    --  3.443   --    --    --    --    --    --
#   Item 38 -1.935   --    --    --  2.235   --    --    --    --    --    --
#   Item 39 -1.947 1.525   --    --  1.439   --    --  0.635   --    --    --
#   Item 40 -2.809   --  2.121   --  2.788   --    --    --    --  -0.19   --

results1$growth
results1$growth.effects
results1$transition.probabilities

# plot results
TDCM::tdcm.plot(results1)

############################################################################
# Example 2: Multigroup TDCM assuming group invariance
############################################################################

# Estimate model
mg.model2 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           groups = groups, time.invariance = FALSE,
                           group.invariance = TRUE)

# summarize results
results2 <- TDCM::mg.tdcm.summary(mg.model2)
results2$item.parameters

## In this case, since group invariance is assumed,
## the item parameters are the same across groups.
## However, items parameters can still vary across time points.

#           l0   l1,1  l1,2  l1,3  l1,4  l2,12 l2,13 l2,14 l2,23 l2,24 l2,34
#  Item 1  -1.983 2.496   --    --    --    --    --    --    --    --    --
#  Item 2  -1.965 2.401   --    --    --    --    --    --    --    --    --
#  Item 3  -1.943 2.33    --    --    --    --    --    --    --    --    --
#  Item 4  -2.133 2.073 1.586   --    --  0.319   --    --    --    --    --
#  Item 5  -1.999 1.676   --  0.948   --    --  1.222   --    --    --    --
#  Item 6  -1.976   --  2.495   --    --    --    --    --    --    --    --
#  Item 7  -1.855   --  2.532   --    --    --    --    --    --    --    --
#  Item 8  -2.011   --  2.445   --    --    --    --    --    --    --    --
#  Item 9  -2.096   --  1.822 1.751   --    --    --    --  0.498   --    --
#  Item 10 -2.044   --  1.357   --  1.456   --    --    --    --  1.327   --
#  Item 11 -1.989   --    --  2.433   --    --    --    --    --    --    --
#  Item 12 -2.172   --    --  2.628   --    --    --    --    --    --    --
#  Item 13 -2.035   --    --  2.461   --    --    --    --    --    --    --
#  Item 14 -2.008   --    --  1.764 1.221   --    --    --    --    --  0.665
#  Item 15 -2.071 1.551   --  1.942   --    --  0.765   --    --    --    --
#  Item 16 -2.108   --    --    --  2.587   --    --    --    --    --    --
#  Item 17 -2.327   --    --    --  3.016   --    --    --    --    --    --
#  Item 18 -2.17    --    --    --  2.727   --    --    --    --    --    --
#  Item 19 -2.073 1.469   --    --  1.316   --    --  1.139   --    --    --
#  Item 20 -2.04    --  1.537   --  1.608   --    --    --    --  0.709   --
```

```
#  Item 21 -1.787 2.316   --    --    --    --    --    --    --    --    --
#  Item 22 -2.14  2.731   --    --    --    --    --    --    --    --    --
#  Item 23 -2.435 2.934   --    --    --    --    --    --    --    --    --
#  Item 24 -2.104 1.485 1.586   --    -- 1.004   --    --    --    --    --
#  Item 25 -2.197 1.263   --  1.734   --    --  1.394   --    --    --    --
#  Item 26 -1.847   --  2.374   --    --    --    --    --    --    --    --
#  Item 27 -1.902   --  2.369   --    --    --    --    --    --    --    --
#  Item 28 -1.961   --  2.418   --    --    --    --    --    --    --    --
#  Item 29 -2.066   --  1.705 1.603   --    --    --    --  0.865   --    --
#  Item 30 -2.137   --  1.437   --  1.76    --    --    --    --  0.753   --
#  Item 31 -1.933   --    --  2.502   --    --    --    --    --    --    --
#  Item 32 -2.222   --    --  2.786   --    --    --    --    --    --    --
#  Item 33 -1.725   --    --  2.264   --    --    --    --    --    --    --
#  Item 34 -1.844   --    --  1.088 1.253   --    --    --    --    --  1.404
#  Item 35 -2.236 1.919   --  1.623   --    --  0.629   --    --    --    --
#  Item 36 -2.211   --    --    --  2.652   --    --    --    --    --    --
#  Item 37 -2.429   --    --    --  3.025   --    --    --    --    --    --
#  Item 38 -1.906   --    --    --  2.216   --    --    --    --    --    --
#  Item 39 -2.026 1.658   --    --  1.48    --    --  0.636   --    --    --
#  Item 40 -2.405   --  1.843   --  1.921   --    --    --    --  0.741   --

results2$growth
results2$growth.effects
results2$transition.probabilities

# plot results
TDCM::tdcm.plot(results2)

############################################################################
# Example 3: Multigroup TDCM assuming time invariance
############################################################################

# Estimate model
mg.model3 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           groups = groups, time.invariance = TRUE,
                           group.invariance = FALSE)

# summarize results
results3 <- TDCM::mg.tdcm.summary(mg.model3)
results3$item.parameters

## Since time invariance is assumed, the item parameters are the same across time points.
## However, items parameters can still vary across groups.

# , , Group 1
#
#            l0   l1,1  l1,2  l1,3  l1,4  l2,12 l2,13 l2,14 l2,23 l2,24 l2,34
#   Item 1  -1.945 2.38    --    --    --    --    --    --    --    --    --
#   Item 2  -2.029 2.44    --    --    --    --    --    --    --    --    --
#   Item 3  -2.174 2.551   --    --    --    --    --    --    --    --    --
#   Item 4  -2.281 1.887 1.859   --    --  0.364   --    --    --    --    --
#   Item 5  -2.155 1.415   --  1.519   --    --  1.16    --    --    --    --
#   Item 6  -1.96    --  2.414   --    --    --    --    --    --    --    --
```

```
#   Item 7  -1.884   --  2.542   --    --    --    --    --    --    --    --
#   Item 8  -1.964   --  2.501   --    --    --    --    --    --    --    --
#   Item 9  -2.11    --  1.771 1.931   --    --    --    -- 0.404   --    --
#   Item 10 -2.198   --  1.535   --  1.847   --    --    --    -- 0.945   --
#   Item 11 -1.966   --    --  2.543   --    --    --    --    --    --    --
#   Item 12 -2.247   --    --  2.749   --    --    --    --    --    --    --
#   Item 13 -1.834   --    --  2.118   --    --    --    --    --    --    --
#   Item 14 -1.952   --    --  0.731 1.23    --    --    --    --    -- 1.817
#   Item 15 -2.291 1.877   --  1.923   --    -- 0.688   --    --    --    --
#   Item 16 -2.262   --    --    --  2.706   --    --    --    --    --    --
#   Item 17 -2.231   --    --    --  2.899   --    --    --    --    --    --
#   Item 18 -1.985   --    --    --  2.439   --    --    --    --    --    --
#   Item 19 -1.961 1.425   --    --  1.195   --    -- 1.245   --    --    --
#   Item 20 -2.244   --  2.284   --  1.701   --    --    --    -- 0.268   --
#
#   , , Group 2
#
#             l0   l1,1  l1,2  l1,3  l1,4  l2,12 l2,13 l2,14 l2,23 l2,24 l2,34
#   Item 1  -1.833 2.364   --    --    --    --    --    --    --    --    --
#   Item 2  -2.146 2.718   --    --    --    --    --    --    --    --    --
#   Item 3  -2.201 2.68    --    --    --    --    --    --    --    --    --
#   Item 4  -1.865 1.597 1.149   --    -- 1.167   --    --    --    --    --
#   Item 5  -1.969 1.309   --  1.118   --    -- 1.665   --    --    --    --
#   Item 6  -1.824   --  2.413   --    --    --    --    --    --    --    --
#   Item 7  -1.832   --  2.349   --    --    --    --    --    --    --    --
#   Item 8  -1.973   --  2.361   --    --    --    --    --    --    --    --
#   Item 9  -2.088   --  1.758 1.25    --    --    --    -- 1.201   --    --
#   Item 10 -1.985   --  1.354   --  1.401   --    --    --    -- 1.106   --
#   Item 11 -2.006   --    --  2.452   --    --    --    --    --    --    --
#   Item 12 -2.192   --    --  2.69    --    --    --    --    --    --    --
#   Item 13 -1.984   --    --  2.624   --    --    --    --    --    --    --
#   Item 14 -1.946   --    --  2.269 1.185   --    --    --    --    -- 0.27
#   Item 15 -2.025 1.539   --  1.453   --    -- 0.957   --    --    --    --
#   Item 16 -2.057   --    --    --  2.549   --    --    --    --    --    --
#   Item 17 -2.435   --    --    --  3.068   --    --    --    --    --    --
#   Item 18 -2.094   --    --    --  2.514   --    --    --    --    --    --
#   Item 19 -2.076 1.589   --    --  1.492   --    -- 0.738   --    --    --
#   Item 20 -2.288   --  1.674   --  2.12    --    --    --    -- 0.495   --

results3$growth
results3$growth.effects
results3$transition.probabilities

# plot results
TDCM::tdcm.plot(results3)

############################################################################
# Example 4: Multigroup TDCM assuming time and group invariance
############################################################################

# Estimate model
mg.model4 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           groups = groups)
```

```
# summarize results
results4 <- TDCM::mg.tdcm.summary(mg.model4)
results4$item.parameters

## Since both time and group invariance are assumed, the item parameters remain the same
## across time and groups.

#            l0  l1,1  l1,2  l1,3  l1,4  l2,12 l2,13 l2,14 l2,23 l2,24 l2,34
#  Item 1  -1.888 2.393   --    --    --    --    --    --    --    --    --
#  Item 2  -2.06  2.572   --    --    --    --    --    --    --    --    --
#  Item 3  -2.185 2.633   --    --    --    --    --    --    --    --    --
#  Item 4  -2.126 1.778 1.575   --    --  0.688   --    --    --    --    --
#  Item 5  -2.072 1.431   --  1.353   --    --  1.328   --    --    --    --
#  Item 6  -1.918   --  2.432   --    --    --    --    --    --    --    --
#  Item 7  -1.888   --  2.451   --    --    --    --    --    --    --    --
#  Item 8  -2.001   --  2.443   --    --    --    --    --    --    --    --
#  Item 9  -2.096   --  1.76  1.694   --    --    --    --  0.699   --    --
#  Item 10 -2.093   --  1.38    --  1.614   --    --    --    --  1.04    --
#  Item 11 -1.973   --    --  2.485   --    --    --    --    --    --    --
#  Item 12 -2.191   --    --  2.697   --    --    --    --    --    --    --
#  Item 13 -1.893   --    --  2.375   --    --    --    --    --    --    --
#  Item 14 -1.941   --    --  1.437 1.236   --    --    --    --    --  1.06
#  Item 15 -2.172 1.743   --  1.784   --    --  0.683   --    --    --    --
#  Item 16 -2.153   --    --    --  2.613   --    --    --    --    --    --
#  Item 17 -2.366   --    --    --  3.007   --    --    --    --    --    --
#  Item 18 -2.046   --    --    --  2.47    --    --    --    --    --    --
#  Item 19 -2.036 1.537   --    --  1.367   --    --  0.922   --    --    --
#  Item 20 -2.225   --  1.721   --  1.774   --    --    --    --  0.694   --
results4$growth
results4$growth.effects
results4$transition.probabilities

############################################################################
# Example 5: Assess measurement invariance
############################################################################

# Compare model 1 (no group invariance) with model 2 (group invariance)
TDCM::tdcm.compare(mg.model1, mg.model2)

# Compare model 1 (no time invariance) with model 3 (time invariance)
TDCM::tdcm.compare(mg.model1, mg.model3)

#############################################################################
# Example 6: DINA multigroup TDCM with time and group invariance assumed
############################################################################

# Estimate model
mg.model6 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           rule = "DINA",
                           groups = groups, time.invariance = TRUE,
                           group.invariance = TRUE)
```

```
# summarize results
results6 <- TDCM::mg.tdcm.summary(mg.model6)
results6$item.parameters
results6$growth
results6$growth.effects
results6$transition.probabilities

#############################################################################
# Example 7: DINO multigroup with time and group invariance assumed
############################################################################

# Estimate model
mg.model7 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           rule = "DINO",
                           groups = groups, time.invariance = TRUE,
                           group.invariance = TRUE)

# summarize results
results7 <- TDCM::mg.tdcm.summary(mg.model7)
results7$item.parameters
results7$growth
results7$growth.effects
results7$transition.probabilities

#############################################################################
# Example 8: CRUM multigroup with time and group invariance assumed
############################################################################

# Estimate model
mg.model8 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           rule = "CRUM",
                           groups = groups, time.invariance = TRUE,
                           group.invariance = TRUE)

# summarize results
results8 <- TDCM::mg.tdcm.summary(mg.model8)
results8$item.parameters
results8$growth
results8$growth.effects
results8$transition.probabilities

#############################################################################
# Example 9: RRUM multigroup with time and group invariance assumed
############################################################################

# Estimate model
mg.model9 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           rule = "RRUM",
                           groups = groups, time.invariance = TRUE,
                           group.invariance = TRUE)

# summarize results
results9 <- TDCM::mg.tdcm.summary(mg.model9)
```

```
results9$item.parameters
results9$growth
results9$growth.effects
results9$transition.probabilities

##############################################################################
# Example 10: Multigroup TDCM with and without forgetting
#############################################################################

##----------------------------------------------------------------------------
# With forgetting
#----------------------------------------------------------------------------
## Consider a default model in which students can retain or lose their mastery status
## from one time point to another

# Estimate model
mg.model10_forgetting <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           rule = "LCDM",
                           groups = groups, time.invariance = TRUE,
                           group.invariance = TRUE)

# Summarize results with mg.tdcm.summary().
results_forgetting <- TDCM::mg.tdcm.summary(mg.model10_forgetting,transition.option = 1)
results_forgetting$transition.probabilities
# , , Attribute 1: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.634  0.366
# T1 [1]  0.399  0.601
#
# , , Attribute 2: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.571  0.429
# T1 [1]  0.393  0.607
#
# , , Attribute 3: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.438  0.562
# T1 [1]  0.185  0.815
#
# , , Attribute 4: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.334  0.666
# T1 [1]  0.166  0.834
#
# , , Attribute 1: Time 1 to Time 2, Group 2
#
#         T2 [0] T2 [1]
# T1 [0]  0.435  0.565
# T1 [1]  0.231  0.769
```

```
#
# , , Attribute 2: Time 1 to Time 2, Group 2
#
#         T2 [0] T2 [1]
# T1 [0]  0.362  0.638
# T1 [1]  0.104  0.896
#
# , , Attribute 3: Time 1 to Time 2, Group 2
#
#         T2 [0] T2 [1]
# T1 [0]  0.361  0.639
# T1 [1]  0.073  0.927
#
# , , Attribute 4: Time 1 to Time 2, Group 2
#
#         T2 [0] T2 [1]
# T1 [0]  0.353  0.647
# T1 [1]  0.208  0.792

##-------------------------------------------------------------------------------
# Without forgetting
#-------------------------------------------------------------------------------
## Consider a model in which students cannot lose their mastery status for attribute 4
## from one time point to another.

# Estimate  model
mg.model10_noforgetting <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                                       rule = "LCDM",
                                       groups = groups, time.invariance = TRUE,
                                       group.invariance = TRUE,
                                       forget.at=c(4))

# Summarize results with mg.tdcm.summary().
results_noforgetting <- TDCM::mg.tdcm.summary(mg.model10_noforgetting,transition.option = 1 )
results_noforgetting$transition.probabilities
# , , Attribute 1: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.635  0.365
# T1 [1]  0.396  0.604
#
# , , Attribute 2: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.570  0.430
# T1 [1]  0.406  0.594
#
# , , Attribute 3: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.435  0.565
# T1 [1]  0.199  0.801
#
```

```
# , , Attribute 4: Time 1 to Time 2, Group 1
#
#         T2 [0] T2 [1]
# T1 [0]  0.376  0.624
# T1 [1]  0.000  1.000
#
# , , Attribute 1: Time 1 to Time 2, Group 2
#
#         T2 [0] T2 [1]
# T1 [0]  0.435  0.565
# T1 [1]  0.241  0.759
#
# , , Attribute 2: Time 1 to Time 2, Group 2
#
#         T2 [0] T2 [1]
# T1 [0]  0.365  0.635
# T1 [1]  0.122  0.878
#
# , , Attribute 3: Time 1 to Time 2, Group 2
#
#        T2 [0] T2 [1]
# T1 [0]  0.361  0.639
# T1 [1]  0.075  0.925
#
# , , Attribute 4: Time 1 to Time 2, Group 2
#
#         T2 [0] T2 [1]
# T1 [0]  0.415  0.585
# T1 [1]  0.000  1.000
```

mg.tdcm.summary — *Multigroup TDCM results compiler and summarizer*

Description

Function to summarize results obtained with the mg.tdcm function. It includes information regarding the item parameters, attribute posterior probabilities, transition posterior probabilities, attribute mastery classifications, growth, growth effects,transition probabilities, attribute correlations, model fit statistics, and several transition reliability metrics developed by Templin and Bradshaw (2013) and Johnson and Sinharay (2020).

Usage

```
mg.tdcm.summary(
  model,
  transition.option = 1,
  classthreshold = 0.5,
```

```
  attribute.names = c(),
  group.names = c()
)
```

Arguments

`model` A `tdcm` object returned from the `mg.tdcm` function.

`transition.option`
An optional argument to specify how transition probabilities should be reported for each attribute in a Q-matrix across time points.

- `transition.option = 1` (default): Summarizes the transition probabilities by comparing the first and last time point.
- `transition.option = 2`: Summarizes the transition by comparing the first time point to every subsequent time point.
- `transition.option = 3`: summarizes the transition probabilities by comparing each consecutive time point sequentially.

`classthreshold` A numeric value between 0 and 1 specifying the probability threshold for determining examinees' proficiency based on the posterior probabilities.

- The default value is `.50`, which optimizes overall classification accuracy.
- Lower values reduce the probability of false negatives, such that fewer mastery examinees are misclassified as non-proficient.
- Higher values reduce the probability of false positives, such that fewer non-master examinees are misclassified as proficient.

`attribute.names`
An optional character `vector` specifying the attribute names to be included in the plots. By default, `attribute.names=NULL`, which uses the generic attribute labels from the Q-matrix.

`group.names` An optional character `vector` specifying the group names to be included in the plots. By default, `group.names=NULL`, which uses a generic group label based on the number of groups in the model (e.g., Group 1, Group 2, etc).

Value

A list with the following items:

- `$item.parameters`: Item parameter estimates (logit) from the specified DCM.
- `$growth`: Proficiency proportions for each time point and each attribute.
- `$growth.effects`: It includes three growth effect size metrics for each attribute and specified transitions:
  1. **Growth**: Difference in proficiency proportions between the later and earlier time point.
  2. **Odds Ratio**: Ratio between the proficiency odds at the later time point and the proficiency odds at the earlier time point.
  3. **Cohen's h** (Cohen, 1988): Arcsine-transformed difference in proficiency proportions.

  Note that the `growth.effect` output directly depend on the option specified in `transition.option`.

  Example:

  Suppose a test measures two attributes at three time points. Because there are more than two time points, the growth effect output is calculated based on the option specified in `transition.option`.

  - If `transition.option=1`, the growth effect for Attribute 1 and 2 is computed between time point 1 (earlier) and time point 3 (latter).
  - If `transition.option=2`, the growth effect for Attribute 1 and 2 is computed between:
    * Time point 1 (earlier) and time point 2 (latter).
    * Time point 1 (earlier) and time point 3 (latter).
  - If `transition.option=3`, the growth effect for Attribute 1 and 2 is obtained between:
    * Time point 1 (earlier) and time point 2 (latter).
    * Time point 2 (earlier), and time point 3 (latter).

- `$transition.probabilities`: Conditional attribute proficiency transition probability matrices.
- `$posterior.probabilities`: Examinee marginal attribute posterior probabilities of proficiency.
- `$transition.posteriors`: Examinee marginal attribute transition posterior probabilities.
- `$most.likely.transitions`: Examinee most likely transitions for each attribute and transition.
- `$classifications`: Examinee classifications determined by the specified threshold applied to the posterior probabilities.
- `$reliability`: Estimated transition reliability metrics for each attribute for the specified transitions option specified. It includes seven metrics:
  - **pt bis**: Longitudinal point biserial metric, which reflects the ratio between the estimated attribute proficiency base rates with the attribute proficiency posterior probabilities.
  - **info gain**: Longitudinal information gain metric. It quantifies how much additional information is gained regarding an attribute's transition over time.
  - **polychor**: Longitudinal tetrachoric metric. It quantifies how consistently an examinee transitions between mastery states between two time points.
  - **ave max tr**: Average maximum transition posterior metric. It quantities how likely an examinee is classified into a specific transition state over time.
  - **P(t > k)**: Proportion of examinees whose marginal attribute transition posteriors exceed a threshold k. The thresholds used are $k = 0.6, 0.7, 0.8$, and 0.9, representing the proportion of examinees with attribute transition posterior probabilities greater than these values. For example, if P(t>.6) = 0.90, 90% of examinees have a posterior probability greater than 0.6.
  - **wt pt bis**: Weighted longitudinal point biserial. A variation of the longitudinal point biserial metric that computes the correlation between true attribute transition classification and observed marginal transition probabilities. It assigns greater weight to more prevalent attribute transitions based on each attributes' transition base rate, ensuring that transitions occurring more frequently in the data contribute more significantly to the computed reliability value.
  - **wt info gain**: Weighted longitudinal information gain. A variation of the longitudinal information gain that quantifies the additional information provided by the attribute transition posterior probabilities in predicting examinees' true transition status. It assigns greater weight to more prevalent attribute transitions, ensuring that transitions occurring more frequently in the data contribute more significantly to the computed reliability value.
- `$att.corr`: Estimated attribute correlation matrix.

- `$model.fit`: Several model fit indices and tests are output including:
  - Item root mean square error of approximation (RMSEA; von Davier, 2005).
  - Mean RMSEA.
  - Bivariate item fit statistics (Chen et al., 2013).
  - Absolute fit statistics such as mean absolute deviation for observed.
  - Expected item correlations (MADcor; DiBello, Roussos, & Stout, 2007).
  - Standardized root mean square root of squared residuals (SRMSR; Maydeu-Olivares, 2013).

References

Chen, J., de la Torre, J. ,& Zhang, Z. (2013). Relative and absolute fit evaluation in cognitive diagnosis modeling. *Journal of Educational Measurement, 50*, 123-140.

DiBello, L. V., Roussos, L. A., & Stout, W. F. (2007). *Review of cognitively diagnostic assessment and a summary of psychometric models*. In C. R. Rao and S. Sinharay (Eds.), Handbook of Statistics, Vol. 26 (pp.979–1030). Amsterdam: Elsevier.

Johnson, M. S., & Sinharay, S. (2020). The reliability of the posterior probability of skill attainment in diagnostic classification models. *Journal of Educational Measurement, 47*(1), 5 – 31.

Madison, M. J. (2019). Reliably assessing growth with longitudinal diagnostic classification models. *Educational Measurement: Issues and Practice, 38*(2), 68-78.

Madison, M. J., & Bradshaw, L. (2018). Evaluating intervention effects in a diagnostic classification model framework. *Journal of Educational Measurement, 55*(1), 32-51.

Maydeu-Olivares, A. (2013). Goodness-of-fit assessment of item response theory models (with discussion). *Measurement: Interdisciplinary Research and Perspectives, 11*, 71-137.

Schellman, M., & Madison, M. J. (2024). Estimating the reliability of skill transition in longitudinal DCMs. *Journal of Educational and Behavioral Statistics*.

Templin, J., & Bradshaw, L. (2013). Measuring the reliability of diagnostic classification model examinee estimates. *Journal of Classification, 30*, 251-275.

von Davier M. (2008). A general diagnostic model applied to language testing data. *The British journal of mathematical and statistical psychology, 61*(2), 287–307.

Examples

```
### ADD EXAMPLE WITH DIFFERENT TRANSITION OPTION --> there is no dataset for this

############################################################################
# Example 1: Multigroup TDCM assuming time and group invariance
############################################################################

# Load data: G = 2, T = 2, A = 4, I = 20
data(data.tdcm04, package = "TDCM")
data <- data.tdcm04$data
q.matrix <- data.tdcm04$q.matrix
groups <- data.tdcm04$groups
```

```
# Estimate model
mg.model1 <- TDCM::mg.tdcm(data, q.matrix, num.time.points = 2,
                           groups = groups, time.invariance = TRUE,
                           group.invariance = TRUE)

#----------------------------------------------------------------------------
# With different thresholds
#----------------------------------------------------------------------------
## a) If classthreshold = 0.5 (default)

# Summarize results
results1 <- TDCM::mg.tdcm.summary(mg.model1, transition.option = 1)
head(results1$posterior.probabilities)
#       T1A1  T1A2  T1A3  T1A4  T2A1  T2A2  T2A3  T2A4
# [1,] 0.000 0.000 0.000 0.002 0.343 0.009 0.961 0.908
# [2,] 0.000 0.013 0.000 0.013 0.009 0.006 0.001 0.006
# [3,] 0.029 0.013 0.000 0.000 0.006 0.004 0.848 0.004
# [4,] 0.020 0.015 0.006 0.651 0.999 0.998 0.973 1.000
# [5,] 0.660 0.978 0.002 0.006 0.137 0.779 0.025 1.000
# [6,] 0.999 0.998 0.998 1.000 0.001 1.000 0.990 0.894

head(results1$classifications)
#      T1A1 T1A2 T1A3 T1A4 T2A1 T2A2 T2A3 T2A4
#    1    0    0    0    0    0    0    1    1
#    2    0    0    0    0    0    0    0    0
#    3    0    0    0    0    0    0    1    0
#    4    0    0    0    1    1    1    1    1
#    5    1    1    0    0    0    1    0    1
#    6    1    1    1    1    0    1    1    1

## b) If classthreshold = 0.7

# Summarize results
results2 <- TDCM::mg.tdcm.summary(mg.model1, transition.option = 1, classthreshold = 0.7)
head(results2$posterior.probabilities)
#       T1A1  T1A2  T1A3  T1A4  T2A1  T2A2  T2A3  T2A4
# [1,] 0.000 0.000 0.000 0.002 0.343 0.009 0.961 0.908
# [2,] 0.000 0.013 0.000 0.013 0.009 0.006 0.001 0.006
# [3,] 0.029 0.013 0.000 0.000 0.006 0.004 0.848 0.004
# [4,] 0.020 0.015 0.006 0.651 0.999 0.998 0.973 1.000
# [5,] 0.660 0.978 0.002 0.006 0.137 0.779 0.025 1.000
# [6,] 0.999 0.998 0.998 1.000 0.001 1.000 0.990 0.894

head(results2$classifications)
#       T1A1 T1A2 T1A3 T1A4 T2A1 T2A2 T2A3 T2A4
#    1    0    0    0    0    0    0    1    1
#    2    0    0    0    0    0    0    0    0
#    3    0    0    0    0    0    0    1    0
#    4    0    0    0    0    1    1    1    1
#    5    0    1    0    0    0    1    0    1
#    6    1    1    1    1    0    1    1    1

## c) If classthreshold = 0.3
```

```
# Summarize results
results3 <- TDCM::mg.tdcm.summary(mg.model1, transition.option = 1, classthreshold = 0.3)
head(results3$posterior.probabilities)
#       T1A1  T1A2  T1A3  T1A4  T2A1  T2A2  T2A3  T2A4
# [1,] 0.000 0.000 0.000 0.002 0.343 0.009 0.961 0.908
# [2,] 0.000 0.013 0.000 0.013 0.009 0.006 0.001 0.006
# [3,] 0.029 0.013 0.000 0.000 0.006 0.004 0.848 0.004
# [4,] 0.020 0.015 0.006 0.651 0.999 0.998 0.973 1.000
# [5,] 0.660 0.978 0.002 0.006 0.137 0.779 0.025 1.000
# [6,] 0.999 0.998 0.998 1.000 0.001 1.000 0.990 0.894

head(results3$classifications)
#      T1A1 T1A2 T1A3 T1A4 T2A1 T2A2 T2A3 T2A4
#    1    0    0    0    0    1    0    1    1
#    2    0    0    0    0    0    0    0    0
#    3    0    0    0    0    0    0    1    0
#    4    0    0    0    1    1    1    1    1
#    5    1    1    0    0    0    1    0    1
#    6    1    1    1    1    0    1    1    1
```

`oneplcdm` *One-parameter log-linear cognitive diagnosis model.*

Description

Function to estimate the 1-PLCDM (Madison et al., 2023; Maas et al., 2023).

Usage

```
oneplcdm(data, q.matrix, progress = TRUE)
```

Arguments

`data`	a required $N \times I$ matrix. Binary item responses are in the columns.
`q.matrix`	a required $I \times A$ matrix indicating which items measure which attributes.
`progress`	An optional logical indicating whether the function should print the progress of estimation.

Details

Estimates the single-attribute and multi-attribute 1-PLCDM described in Madison et al. (2024). Example shows that attribute subscores are sufficient statistics for classifications.

Value

An object of class `gdina` with entries as indicated in the CDM package.

Note

Currently, this model cannot be embedded within the TDCM via the `rule` argument.

References

George, A. C., Robitzsch, A., Kiefer, T., Gross, J., & Ünlü , A. (2016). The R package CDM for cognitive diagnosis models. *Journal of Statistical Software, 74*(2), 1-24.

Henson, R., Templin, J., & Willse, J. (2009). Defining a family of cognitive diagnosis models using log linear models with latent variables. *Psychometrika, 74*, 191-21.

Madison, M.J., Wind, S., Maas, L., Yamaguchi, K. & Haab, S. (2024). A one-parameter diagnostic classification model with familiar measurement properties. *Journal of Educational Measurement*.

Maas, L., Madison, M. J., & Brinkhuis, M. J. (2024). Properties and performance of the one-parameter log-linear cognitive diagnosis model. *Frontiers*.

Examples

```
## Example 1: A = 4
data(data.tdcm05)
dat5 <- data.tdcm05$data
qmat5 <- data.tdcm05$q.matrix

# calibrate LCDM
m1 <- CDM::gdina(dat5, qmat5, linkfct = "logit", method = "ML")

# calibrate 1-PLCDM
m2 <- TDCM::oneplcdm(dat5, qmat5)
summary(m2)
#demonstrate 1-PLCDM sum score sufficiency for each attribute
subscores <- cbind(rowSums(dat5[, 1:5]), rowSums(dat5[, 6:10]),
rowSums(dat5[, 11:15]), rowSums(dat5[, 16:20]))
colnames(subscores) <- c("Att1", "Att2", "Att3", "Att4")
proficiency <- cbind(m2$pattern[, 6] > .50, m2$pattern[, 7] > .50,
m2$pattern[, 8] > .50, m2$pattern[, 9] > .5) * 1
table(subscores[, 1], proficiency[, 1])
table(subscores[, 2], proficiency[, 2])
table(subscores[, 3], proficiency[, 3])
table(subscores[, 4], proficiency[, 4])

#plot sum score sufficiency for each attribute
posterior1pl <- m2$pattern[, 6:9]
posteriorlcdm <- m1$pattern[, 6:9]
oldpar <- par(mfrow = c(2, 2))
for (i in 1:4) {
 plot(subscores[, i], posteriorlcdm[, i], pch = 19,las = 1, cex.lab = 1.5,
 xlab = "Sum Scores", ylab = "P(proficiency)",
 cex.main = 1.5, col = "grey", xaxt = "n", yaxt = "n", cex = 1.2,
 main = paste("Attribute ", i, sep = ""))
 graphics::axis(side = 1, at = c(0, 1, 2, 3, 4, 5), )
 graphics::axis(side = 2, at = c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1.0), las = 1)
 graphics::points(subscores[, i], posterior1pl[, i], col = "black", pch = 18, cex = 1.5)
```

```
 graphics::abline(a = .50, b = 0, col = "red")
 graphics::legend("bottomright", c("1-PLCDM", "LCDM"), col = c("black", "grey"),
 pch = c(18 ,19), box.lwd = 0, box.col = "white", bty = 'n')
}
par(oldpar)
```

---

tdcm — *Estimating the Transition Diagnostic Classification Model (TDCM)*

Description

`tdcm()` estimates the transition diagnostic classification model (TDCM; Madison & Bradshaw, 2018a), which is a longitudinal extension of the log-linear cognitive diagnosis model (LCDM; Henson, Templin, & Willse, 2009). For the multigroup TDCM, see `mg.tdcm()`. This function supports the estimation of various longitudinal DCMs by allowing different rule specifications via the `rule` option and link functions via the `linkfct` option, with LCDM as the default rule and link function. The rule can be modified to estimate the DINA model, DINO model, CRUM (i.e., ACDM, or main effects model), or reduced interaction versions of the LCDM. Additionally, the link function can be adjusted to specify the GDINA model.

Usage

```
tdcm(
  data,
  q.matrix,
  num.time.points,
  invariance = TRUE,
  rule = "LCDM",
  linkfct = "logit",
  num.q.matrix = 1,
  num.items = c(),
  anchor = c(),
  forget.att = c(),
  progress = TRUE
)
```

Arguments

`data`	A required $N \times T \times I$ `matrix` or `data.frame` where rows correspond to `N` examinees and columns represent the binary item responses across `T` time points and `I` items.
`q.matrix`	A required $I \times A$ `matrix` indicating which items measure which attributes. If there are multiple Q-matrices, then they must have the same number of attributes and must be stacked on top of each other for estimation (to specify multiple Q-matrices, see `num.q.matrix`, `num.items`, and `anchor`).

`num.time.points`
: A required integer ≥ 2 specifying the number of time points (i.e., measurement occasions).

`invariance`
: logical. If `TRUE` (the default), then item parameters will be constrained to be equal at each time point. If `FALSE`, item parameters are not assumed to be equal over time.

`rule`
: A `string` or a `vector` indicating the specific DCM to be employed. A vector of supported `rule` values is provided by tdcm.rules. Currently accepted values are: "LCDM", "DINA", "DINO", "CRUM", "RRUM", "LCDM1" for the LCDM with only main effects, "LCDM2" for the LCDM with two-way interactions, "LCDM3", and so on. If `rule` is supplied as a single string, then that DCM will be assumed for each item. If entered as a vector, a rule can be specified for each item. The rule vector must have length equal to the total number of items across all time points.

`linkfct`
: A `string` or a `vector` indicating the LCDM link function. Currently accepts "logit" (default) to estimate the LCDM, "identity" to estimate the GDINA model, and "log" link function to estimate the reduced reparameterized unified model (RRUM). The link function vector must have length equal to the total number of items across all time points.

`num.q.matrix`
: An optional integer specifying the number of Q-matrices. For many applications, the same assessment is administered at each time point and this number is 1 (the default). If there are different Q-matrices for each time point, then this argument must be specified and should be equal to the number of time points. For example, if there are three time points, and the Q-matrices for each time point are different, then `num.q.matrix = 3`. If there are three time points, and the Q-matrix is different only for time point 3, then `num.q.matrix` is still specified as `3`.

`num.items`
: An integer specifying the number of items. When there are multiple Q-matrices, the number of items in each Q-matrix is specified as a vector. For example, if there are three time points, and the Q-matrices for each time point have 8, 10, and 12 items, respectively. Then `num.items = c(8, 10, 12)`.

`anchor`
: An optional `vector` specifying how items are linked across time points to maintain item invariance when different tests are administered. By default, `anchor` is an empty `vector`, indicating the absence of anchor items. **Note:** When `anchor` is specified, invariance is automatically set to `FALSE` for non-anchor items. Each pair in the `anchor` vector consists of a **reference item** and a **linked item**, where the linked item is mapped to its corresponding reference item. The reference item does not necessarily appear in the first test; it can be from any time point.

  Example: Suppose we have three different 10-item tests with their corresponding Q-matrices. However, some items remain the same across time points:

  - **Item 1** (from the first test), **Item 11** (from the second test), and **Item 21** (from the third test) correspond to the same item. Since **Item 1** serves as the reference, **Items 11** and **21** can be linked to it using: `anchor = c(1, 11, 1, 21)`
  - If we additionally assume that **Item 14** (from the second test) and **Item 24** (from the third test) correspond to the same item, **Item 14** serves as

the reference, and **Item 24** is linked to it. Thus, the final anchor vector is specified as: `anchor = c(1, 11, 1, 21, 14, 24)`

`forget.att` An optional vector allowing for constraining of individual attribute proficiency loss, or forgetting.

- By default, forgetting is allowed for all measured attributes, meaning that probability of transitioning from mastery to non-mastery can be different than zero ($P(1 \to 0) \neq 0$).
- If a vector of attributes is provided, $P(1 \to 0) = 0$ for those specific attributes, meaning that forgetting is not permitted. For example, if `forget.att= c(2,4)`, then forgetting for Attributes 2 and 4 is not allowed, while other attributes can exhibit forgetting.

`progress` logical. If `FALSE`, the function will print the progress of estimation. If `TRUE` (default), no progress information is printed.

Details

Transition Diagnostic Classification Model (TDCM)

TDCM is a confirmatory and constrained latent transition model that measures examinees' growth or decline in attribute mastery over time (Madison & Bradshaw, 2018a). Assume that X_{eit} corresponds to the binary response of examinee $e \in \{1, \ldots, N\}$ to item $i \in \{1, \ldots, I\}$ across time points $t \in \{1, \ldots, T\}$, and A_t denotes the number of attributes measured at time t. The probability of the item response vector $X_e = (x_{e11}, x_{e12}, \ldots, x_{e1I}, x_{e21}, \ldots, x_{eTI})$ is given by:

$$P(X_e = x_e) = \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} \cdots \sum_{c_T=1}^{C} v_{c_1} \tau_{c_2|c_1} \tau_{c_3|c_2} \cdots \tau_{c_T|c_{T-1}} \prod_{t=1}^{T} \prod_{i=1}^{I} \pi_{ic_t}^{x_{eit}} (1 - \pi_{ic_t})^{1-x_{eit}},$$

where:

- v_{c_1} represents the probability of belonging to attribute profile c at time 1.
- $\tau_{c_t|c_{t-1}}$ represents the probability of transitioning attribute profiles from time point $t-1$ to time point t.
- π_{ic_t} is the item response function, which models the probability of answering item i correctly at time t given attribute profile c.

Model Assumptions and Variations

1. Accounting for Measurement Invariance

Measurement invariance indicates whether the **item response function** remains **consistent over time** or changes across time points. Depending on the testing conditions, different measurement invariance assumptions can be assumed:

a) No Measurement Invariance:

- If measurement invariance is **not** assumed, each item has a **different** response function over time: $\pi_{ic_1} \neq \pi_{ic_2} \neq \ldots \neq \pi_{ic_T}$. Thus, the probability of the item response vector is:

$$P(X_e = x_e) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C}\cdots\sum_{c_T=1}^{C} v_{c_1}\tau_{c_2|c_1}\tau_{c_3|c_2}\cdots\tau_{c_T|c_{T-1}}\prod_{t=1}^{T}\prod_{i=1}^{I}\pi_{ic_t}^{x_{eit}}(1-\pi_{ic_t})^{1-x_{eit}},$$

and

$$\pi_{ic_t} = P(X_{ic_t} = 1|\alpha_{c_t}) = \frac{exp(\lambda_{i,0} + \boldsymbol{\lambda_i^{(t)T} h(\alpha_{c_t}, q_i^{(t)})})}{1 + exp(\lambda_{i,0} + \boldsymbol{\lambda_i^{(t)T} h(\alpha_{c_t}, q_i^{(t)})})},$$

where:

- $q_i^{(t)}$ is the q-matrix for item i at time point t.
- $\lambda_{i,0}$ is the intercept parameter for item i and corresponds to the logit of a correct response when none of the attributes in the Q-matrix are mastered.
- $\boldsymbol{\lambda_i}^{(t)}$ is a column vector of main and interaction effects for item i at time point t.
- $\boldsymbol{h(\alpha_{c_t}, q_i^{(t)})}$ is a function mapping the attribute profile α_{c_t} and the Q-matrix for item i at time point t.

b) Full Measurement Invariance:

If measurement invariance **is** assumed (default option), items maintain a **constant response function across time**: $\forall i \in I, \pi_{ic_1} = \pi_{ic_2} = \cdots = \pi_{ic_T}, \forall t \in T$. Therefore, the probability of the item response vector simplifies the to:

$$P(X_e = x_e) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C}\cdots\sum_{c_T=1}^{C} v_{c_1}\tau_{c_3|c_2}\cdots\tau_{c_T|c_{T-1}}\prod_{t=1}^{T}\prod_{i=1}^{I}\pi_{ic}^{x_{eit}}(1-\pi_{ic})^{1-x_{eit}},$$

and

$$\pi_{ic} = P(X_{ic} = 1|\alpha_c) = \frac{exp(\lambda_{i,0} + \boldsymbol{\lambda_i^T h(\alpha_c, q_i)})}{1 + exp(\lambda_{i,0} + \boldsymbol{\lambda_i^T h(\alpha_c, q_i)})},$$

where:

- q_i is the Q-matrix for item i. Recall that as item are the same across time points and measurement invariance is assumed, the Q-matrix should also remain the same across time.
- $\lambda_{i,0}$ is the intercept parameter for item i.
- $\boldsymbol{h(\alpha_{c_t}, q_i)}$ is a function mapping the attribute profile α_c and the item Q-matrix.

c) Partial Measurement Invariance:

When measurement invariance is **partially** assumed, some items (anchor items) maintain the same item response function across time points, while others (non-anchor items) vary over time.

Assume that $i \in B$ are **anchor items**, such that $\forall i \in B, \forall t \in T, \pi_{ic_1} = \pi_{ic_2} = \cdots = \pi_{ic_T}$. This implies that anchor items measure the same attributes across time and their corresponding Q-matrix entries remain unchanged.

Assume also that $i \in Z$ are **non-anchor items**, such that $\forall i \in Z, \forall t \in \{2, \ldots, T\}, \pi_{ic_t} \neq \pi_{ic_{t-1}}$. This means that non-anchor items may change across time or measure different attributes, leading to changes in their corresponding Q-matrix entries. Then, the probability of the item response vector is:

$$P(X_e = x_e) = \sum_{c_1=1}^{C}\sum_{c_2=1}^{C}\cdots\sum_{c_T=1}^{C} v_{c_1}\tau_{c_2|c_1}\tau_{c_3|c_2}\cdots\tau_{c_T|c_{T-1}}\prod_{t=1}^{T}\prod_{i\in B}\pi_{ic}^{x_{eit}}(1-\pi_{ic})^{1-x_{eit}}\prod_{t=1}^{T}\prod_{i\in Z}\pi_{ic_t}^{x_{eit}}(1-\pi_{ic_t})^{1-x_{eit}}.$$

***2. Modeling Forgetting in Attribute Transitions*:**

Unlike standard latent transition models that assume monotonic learning,TDCM allows for **both mastery acquisition and forgetting**. By default, TDCM does not impose that mastery must always increase over time. Instead, the transition probabilities $\tau_{c_t|c_{t-1}}$ for examinee e can represent a transition from:

- A transition from non-mastery status to master attribute status (learning).
- A transition from master attribute status to a non-mastery status (forgetting).

However, TDCM also allows for attribute-specific constrains, enabling to restrict transition probabilities for certain attributes.

3. Special Cases

In TDCM, the item response function π_{ic_t} is parameterized using the LCDM. LCDM is a general and flexible model that allows special models to be derived by constraining specific parameters.

DINA Model:

The DINA model is a non-compensatory DCM, meaning that examinees can correctly answer to an item only if they have mastered all attributes required by that item. Given this characteristic, the DINA model is derived by constraining the main effects of the LCDM to zero, such that only the highest-order interaction term influences the item response probability.

Example*:*

Suppose item 1 measures Attributes 1 and 2, and item invariance is assumed across time points. The item response function for item 1 following the LCDM can be expressed as:

$$\pi_{1c} = P(X_{1c} = 1|\alpha_c) = \frac{exp(\lambda_{1,0} + \lambda_{1,1(1)}\alpha_{c1} + \lambda_{1,1(2)}\alpha_{c2} + \lambda_{1,2(1,2)}\alpha_{c1}\alpha_{c2})}{1 + exp(\lambda_{1,0} + \lambda_{1,1(1)}\alpha_{c1} + \lambda_{1,1(2)}\alpha_{c2} + \lambda_{1,2(1,2)}\alpha_{c1}\alpha_{c2})},$$

Then, the DINA model is obtained by constraining the LCDM main effects to zero, resulting in:

$$\pi_{1c} = P(X_{1c} = 1|\alpha_c) = \frac{exp(\lambda_{1,0} + \lambda_{1,2(1,2)}\alpha_{c1}\alpha_{c2})}{1 + exp(\lambda_{1,0} + \lambda_{1,2(1,2)}\alpha_{c1}\alpha_{c2})}.$$

DINO Model:

The DINO model is a compensatory DCM, meaning that examinees can correctly answer an item if they have mastered at least one of the attributes required by that item. Consequently, the main and interaction terms in the LCDM are constrained to be equal, and we subtract the interaction term to ensure the item response probability remains unchanged when multiple attributes are mastered. Following the previous example, the DINO model can be expressed as:

$$\pi_{1c} = P(X_{1c} = 1|\alpha_c) = \frac{exp(\lambda_{1,0} + (\lambda_{1,1(1)}\alpha_{c1} + \lambda_{1,1(2)}\alpha_{c2} - \lambda_{1,2(1,2)}\alpha_{c1}\alpha_{c2}))}{1 + exp(\lambda_{1,0} + (\lambda_{1,1(1)}\alpha_{c1} + \lambda_{1,1(2)}\alpha_{c2} - \lambda_{1,2(1,2)}\alpha_{c1}\alpha_{c2}))}.$$

CRUM Model:

The CRUM is a compensatory DCM where each attribute independently contributes to the probability of a correct response. Unlike the DINO model, mastering multiple attributes neither penalizes nor provides an additional advantage. Thus, the probability of a correct response is determined solely by the sum of individual main effects, constraining the interaction term to zero.

Following the previous example, the CRUM model can be expressed as:

$$\pi_{1c} = P(X_{1c} = 1|\alpha_c) = \frac{exp(\lambda_{1,0} + \lambda_{1,1(1)}\alpha_{c1} + \lambda_{1,1(2)}\alpha_{c2})}{1 + exp(\lambda_{1,0} + \lambda_{1,1(1)}\alpha_{c1} + \lambda_{1,1(2)}\alpha_{c2})}.$$

Estimation methods

Estimation of the TDCM via the **CDM** package (George, et al., 2016), which is based on an EM algorithm as described in de la Torre (2011). The estimation approach is further detailed in Madison et al. (2023).

Value

An object of class `gdina` with entries as described in `CDM::gdina()`. To see a TDCM-specific summary of the object (e.g.,growth, transitions), use `tdcm.summary()`.

References

de la Torre, J. (2011). The Generalized DINA model framework. *Psychometrika, 76*, 179–199. doi:10.1007/s11336-011-9207-7.

George, A. C., Robitzsch, A., Kiefer, T., Gross, J., & Ünlü , A. (2016). The *R* package **CDM** for cognitive diagnosis models. *Journal of Statistical Software,74*(2), 1-24. doi:10.18637/jss.v074.i02

Henson, R., Templin, J., & Willse, J. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika, 74*, 191-21. doi:10.1007/s11336-008-9089-5.

Johnson, M. S., & Sinharay, S. (2020). The reliability of the posterior probability of skill attainment in diagnostic classification models. *Journal of Educational Measurement, 47*(1), 5–31. doi:10.3102/1076998619864550.

Kaya, Y., & Leite, W. (2017). Assessing change in latent skills across time With longitudinal cognitive diagnosis modeling: An evaluation of model performance. *Educational and Psychological Measurement, 77*(3), 369–388. doi:10.1177/0013164416659314.

Li, F., Cohen, A., Bottge, B., & Templin, J. (2015). A latent transition analysis model for assessing change in cognitive skills. *Educational and Psychological Measurement, 76*(2), 181–204. doi:10.1177/0013164415588946.

Madison, M. J. (2019). Reliably assessing growth with longitudinal diagnostic classification models. *Educational Measurement: Issues and Practice, 38*(2), 68-78. doi:10.1111/emip.12243.

Madison, M. J., & Bradshaw, L. (2018a). Assessing growth in a diagnostic classification model framework. *Psychometrika, 83*(4), 963-990. doi:10.1007/s11336-018-9638-5.

Madison, M. J., & Bradshaw, L. (2018b). Evaluating intervention effects in a diagnostic classification model framework. *Journal of Educational Measurement, 55*(1), 32-51. doi:10.1111/jedm.12162.

Madison, M.J., Chung, S., Kim, J., & Bradshaw, L.P. (2024) Approaches to estimating longitudinal diagnostic classification models. *Behaviormetrika, 51*(7), 7-19. doi:10.1007/s41237-023-00202-5.

Madison, M. J., Jeon, M., Cotterell, M., Haab, S., & Zor, S. (2025). TDCM: An R package for estimating longitudinal diagnostic classification models. *Multivariate Behavioral Research*, 1–10. doi:10.1080/00273171.2025.2453454.

Ravand, H., Effatpanah, F., Kunina-Habenicht, O., & Madison, M. J. (2025). A didactic illustration of writing skill growth through a longitudinal diagnostic classification model. *Frontiers in Psychology*. doi:10.3389/fpsyg.2024.1521808.

Rupp, A. A., Templin, J., & Henson, R. (2010). *Diagnostic Measurement: Theory, Methods, and Applications*. New York: Guilford. ISBN: 9781606235430.

Schellman, M., & Madison, M. J. (2024). Estimating the reliability of skill transition in longitudinal DCMs. *Journal of Educational and Behavioral Statistics*.doi:10.3102/10769986241256032.

Templin, J., & Bradshaw, L. (2013). Measuring the reliability of diagnostic classification model examinee estimates. *Journal of Classification, 30*, 251-275. doi:10.1007/s00357-013-9129-4.

Wang. S., Yang. Y., Culpepper, S. A., & Douglas, J. (2018). Tracking skill acquisition with cognitive diagnosis models: A higher-order, hidden Markov model With covariates. *Journal of Educational and Behavioral Statistics, 43*(1), 57-87. doi:10.3102/1076998617719727.

Examples

```
############################################################################
# Example 1: TDCM with full measurement invariance
############################################################################

# Load dataset: T=2, A=4
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix
# Estimate model
model1 <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                     rule = "LCDM", num.q.matrix = 1)
# Summarize results with tdcm.summary().
results <- TDCM::tdcm.summary(model1)
results$item.parameters
results$growth
results$growth.effects
results$transition.probabilities

#############################################################################
# Example 2: TDCM with no measurement invariance
############################################################################

# Load dataset: T=2, A=4
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix
# Estimate model
model2 <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = FALSE,
                     rule = "LCDM", num.q.matrix = 1)
# Summarize results with tdcm.summary().
results2 <- TDCM::tdcm.summary(model2)
results2$item.parameters
results2$growth
results2$growth.effects
```

```
results2$transition.probabilities


##############################################################################
# Example 3: TDCM with different Q-matrices for each time point and no
# anchor items
#############################################################################

# Load dataset: T=3, A=2
data(data.tdcm03, package = "TDCM")
data <- data.tdcm03$data
q1 <- data.tdcm03$q.matrix.1
q2 <- data.tdcm03$q.matrix.2
q3 <- data.tdcm03$q.matrix.3
q <- data.tdcm03$q.matrix.stacked

# Estimate model
model3 <- TDCM::tdcm(data, q, num.time.points = 3, rule = "LCDM",
                    num.q.matrix = 3, num.items = c(10, 10, 10))

#----------------------------------------------------------------------------
# Summarize results with tdcm.summary() for more than 2 time points.
#----------------------------------------------------------------------------

## There are three post hoc approaches to summarize the transition probabilities
## for each attribute across time using the tdcm.summary() function.
## Each of them is illustrated below.

## 1. When the transition.option argument in the tdcm.summary() is not specified,
## the function assumes by default that transition.option = 1.
## Thus, when summarizing the transition probabilities
## you will compare the results for the first and last time point.

### Summary with default option

results3_def_transition <- TDCM::tdcm.summary(model3)
results3_def_transition$transition.probabilities
#, , Attribute 1: Time 1 to Time 3
#
#       T3 [0] T3 [1]
#T1 [0]  0.202  0.798
#T1 [1]  0.146  0.854
#
#, , Attribute 2: Time 1 to Time 3
#
#       T3 [0] T3 [1]
#T1 [0]  0.325  0.675
#T1 [1]  0.257  0.743

## 2. When the transition.option = 2, you can compare the transition probabilities
## from the first time point to every other time point. In this case, you can
## compare the transition probabilities between Time Point 1 and Time Point 2,
## and Time Point 1 with Time Point 3.
```

```
### Summary with transition.option = 2
results3_2transition <- TDCM::tdcm.summary(model3, transition.option = 2)
results3_2transition$transition.probabilities
#, , Attribute 1: Time 1 to Time 2
#
#.    [0]   [1]
#[0] 0.510 0.490
#[1] 0.424 0.576
#
#, , Attribute 2: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.456 0.544
#[1] 0.334 0.666
#
#, , Attribute 1: Time 1 to Time 3
#
#     [0]   [1]
#[0] 0.202 0.798
#[1] 0.146 0.854
#
#, , Attribute 2: Time 1 to Time 3
#
#     [0]   [1]
#[0] 0.325 0.675
#[1] 0.257 0.743

## 3. When the transition.option = 3, you can compare the transition probabilities
## sequentially, such that for each attribute, you can compare the transition
## probabilities between Time Point 1 and Time Point 2, Time Point 2 and Time Point 3

### Summary with transition.option = 3
results3_3transition <- TDCM::tdcm.summary(model3, transition.option = 3)
results3_3transition$transition.probabilities
#, , Attribute 1: Time 1 to Time 2
#
#    [0]   [1]
#[0] 0.510 0.490
#[1] 0.424 0.576
#
#, , Attribute 2: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.456 0.544
#[1] 0.334 0.666
#
#, , Attribute 1: Time 2 to Time 3
#
#     [0]   [1]
#[0] 0.183 0.817
#[1] 0.188 0.812
#
```

```
#, , Attribute 2: Time 2 to Time 3
#
#     [0]   [1]
#[0] 0.361 0.639
#[1] 0.262 0.738

#############################################################################
# Example 4: Full TDCM with different Q-matrices for each time point and
# anchor items
############################################################################

# Load dataset: T=3, A=2
data <- data.tdcm03$data
q1 <- data.tdcm03$q.matrix.1
q2 <- data.tdcm03$q.matrix.2
q3 <- data.tdcm03$q.matrix.3
q <- data.tdcm03$q.matrix.stacked
## Estimate model
## Anchor items:
## - item 1, item 11, and item 21 are the same
## - item 14 and item 24 are the same.

model4 <- TDCM::tdcm(data, q, num.time.points = 3, rule = "LCDM",
                    num.q.matrix = 3, anchor = c(1,11,
                                                 1,21,
                                                 14,24),
                    num.items = c(10, 10, 10))

# Summarize results with tdcm.summary().
results4 <- TDCM::tdcm.summary(model4)
results4$item.parameters
results4$growth
results4$growth.effects
results4$transition.probabilities

#----------------------------------------------------------------------------
#Compare models from example 3 and 4 to assess measurement invariance
#----------------------------------------------------------------------------

## Additionally, we can measure the measurement invariance between a TDCM model
## that assumes full measurement invariance (model3) and a model that assumes partial
## measurement invariance (model4)

model_comparison <- tdcm.compare(model3, model4)


#############################################################################
# Example 5: DINA TDCM with full measurement invariance
############################################################################

# Load dataset: T=2, A=4
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
```

```
q.matrix <- data.tdcm01$q.matrix

# Estimate model
model5 <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                    rule = "DINA", num.q.matrix = 1)
# Summarize results with tdcm.summary().
results5 <- TDCM::tdcm.summary(model5)
results5$item.parameters
results5$growth
results5$growth.effects
results5$transition.probabilities

#############################################################################
# Example 6: DINO TDCM with full measurement invariance
############################################################################

# Load dataset: T=2, A=4
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix

# Estimate model
model6 <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                    rule = "DINO", num.q.matrix = 1)
# Summarize results with tdcm.summary().
results6 <- TDCM::tdcm.summary(model6)
results6$item.parameters
results6$growth
results6$growth.effects
results6$transition.probabilities

#############################################################################
# Example 7: CRUM TDCM with full measurement invariance
############################################################################

# Load dataset: T=2, A=4
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix

# Estimate model
model7 <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                    rule = "CRUM", num.q.matrix = 1)
# Summarize results with tdcm.summary().
results7 <- TDCM::tdcm.summary(model7)
results7$item.parameters
results7$growth
results7$growth.effects
results7$transition.probabilities

#############################################################################
# Example 8: RRUM TDCM with full measurement invariance
############################################################################
```

```
# Load dataset: T=2, A=4
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix

# Estimate model
model8 <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                    rule = "RRUM", num.q.matrix = 1)
# Summarize results with tdcm.summary().
results8 <- TDCM::tdcm.summary(model8)
results8$item.parameters
results8$growth
results8$growth.effects
results8$transition.probabilities

############################################################################
# Example 9: TDCM with and without forgetting
###########################################################################

# Load dataset: T=2, A=4,
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix

##---------------------------------------------------------------------------
# With forgetting
#---------------------------------------------------------------------------
## Consider a default model in which students can retain or lose their mastery status
## from one time point to another

# Estimate the model
model11_forgetting <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                                rule = "LCDM", num.q.matrix = 1)

# Summarize results with tdcm.summary().
results_forgetting <- TDCM::tdcm.summary(model11_forgetting, transition.option = 3)
results_forgetting$transition.probabilities

#, , Attribute 1: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.680 0.320
#[1] 0.417 0.583
#
#, , Attribute 2: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.581 0.419
#[1] 0.353 0.647
#
#, , Attribute 3: Time 1 to Time 2
#
```

```
#     [0]   [1]
#[0] 0.549 0.451
#[1] 0.221 0.779
#
#, , Attribute 4: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.371 0.629
#[1] 0.104 0.896

##-----------------------------------------------------------------------------
# Without forgetting
#-----------------------------------------------------------------------------
## Consider a model in which students cannot lose their mastery status for Attribute 4
## from one time point to another.

# Estimate the model
model11_noforgetting <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                                  rule = "LCDM", num.q.matrix = 1, forget.att = c(4))

# Summarize results with tdcm.summary().
results_noforgetting <- TDCM::tdcm.summary(model11_noforgetting, transition.option = 3)
results_noforgetting$transition.probabilities

#, , Attribute 1: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.678 0.322
#[1] 0.416 0.584
#
#, , Attribute 2: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.578 0.422
#[1] 0.359 0.641
#
#, , Attribute 3: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.546 0.454
#[1] 0.226 0.774
#
#, , Attribute 4: Time 1 to Time 2
#
#      [0]   [1]
#[0] 0.382 0.618
#[1] 0.000 1.000
```

---

tdcm.compare *Comparing the fit of two TDCMs*

Description

Provides a comparison of two TDCMs. Can be used to compare different measurement models or assess measurement invariance over time or over groups in the multigroup TDCM case. Only accepts two models.

Usage

```
tdcm.compare(model1, model2)
```

Arguments

`model1`	a `gdina` object returned from the `tdcm` or `mg.tdcm` function.
`model2`	a second `gdina` object returned from the `tdcm` or `mg.tdcm` function

Value

This function returns a data frame with model fit statistics (AIC/BIC) and results from a likelihood ratio or deviance test.

Note

- Currently, this function currently accepts two models for comparison.
- Both models must be fit to the same item responses and Q-matrix.
- The function will provide results for two non-nested models. Please ensure that models are nested before interpreting the likelihood ratio test for nested models.
- The likelihood ratio test is not valid for some model comparisons (e.g., LCDM vs DINA) because of model constraints.

Examples

```
## Example 1: T = 2, A = 4
data(data.tdcm01, package = "TDCM")
dat1 <- data.tdcm01$data
qmat1 <- data.tdcm01$q.matrix

# estimate TDCM with invariance assumed and full LCDM
m1 <- TDCM::tdcm(dat1, qmat1, num.time.points = 2, invariance = TRUE, rule = "LCDM")

# estimate TDCM with invariance not assumed
m2 <- TDCM::tdcm(dat1, qmat1, num.time.points = 2, invariance = FALSE, rule = "LCDM")

# compare models to assess measurement invariance.
TDCM::tdcm.compare(m1, m2)
```

`tdcm.ipd` *Assessing item parameter drift (IPD) in the Transition Diagnostic Classification Model (TDCM)*

Description

The `tdcm.ipd()` function assesses item parameter drift (IPD) in the TDCM (e.g., Madison & Bradshaw, 2018) by applying the Wald test for differential item functioning (de la Torre, 2011; Hou, de la Torre & Nandakumar, 2014). The *p*-values are also calculated by a Holm adjustment for multiple comparisons. In the case of two time points, an effect size of item parameter drift (labeled as UA in the ipd.stats value) is defined as the weighted absolute difference of item response functions.

Usage

```
tdcm.ipd(model)
```

Arguments

`model`	A `tdcm` object returned from the `tdcm` function.

Value

A list with the following items:

- `$ipd.stats`: Data frame containing results of item-wise Wald tests.
- `$coef`: Data frame containing item parameter estimates for each time point.
- `$estimates`: List of λ vectors containing all item parameter estimates.
- `$item.probs.time`: List with predicted item response probabilities for each time point.

References

de la Torre, J. (2011). The Generalized DINA model framework. *Psychometrika, 76*, 179–199. doi:10.1007/s11336-011-9207-7.

Hou, L., de la Torre, J., & Nandakumar, R. (2014). Differential item functioning assessment in cognitive diagnostic modeling: Application of the Wald test to investigate DIF in the DINA model. *Journal of Educational Measurement, 51*, 98-125. doi:10.1111/jedm.12036.

Madison, M. J., & Bradshaw, L. (2018a). Assessing growth in a diagnostic classification model framework. *Psychometrika*, **83**(4), 963-990. doi:10.1007/s11336-018-9638-5.

Examples

```
################################################################################
# Example 1: TDCM with full measurement invariance
################################################################################

# Load dataset: T=2, A=4
```

```
data(data.tdcm01, package = "TDCM")
data <- data.tdcm01$data
q.matrix <- data.tdcm01$q.matrix
# Estimate model
model1 <- TDCM::tdcm(data, q.matrix, num.time.points = 2, invariance = TRUE,
                    rule = "LCDM", num.q.matrix = 1)

# Run IPD analysis
ipd = tdcm.ipd(model1)
ipd$ipd.stats
ipd$coef
ipd$parameters
```

---

tdcm.plot *Plotting TDCM Results*

Description

tdcm.plot() visualizes the results from TDCM analyses.

Usage

```
tdcm.plot(results, attribute.names = c(), group.names = c(), type = "both")
```

Arguments

results	results from tdcm.summary or mg.tdcm.summary
attribute.names	an optional vector of attribute names to include in plots.
group.names	an optional vector of group names to include in plots.
type	an option to specify the type of plot in single group cases; "both" is default and will produce a line plot and a bar chart; "line" will produce a line plot; and "bar" will produce a bar chart.

Value

No return value, called for side effects.

Examples

```
## Example 1: T = 2, A = 4
data(data.tdcm01, package = "TDCM")
dat1 = data.tdcm01$data
qmat1 = data.tdcm01$q.matrix

#estimate TDCM with invariance assumed and full LCDM
```

```
m1 = TDCM::tdcm(dat1, qmat1, num.time.points = 2, invariance = TRUE, rule = "LCDM")

#summarize results with tdcm.summary function
results1 = TDCM::tdcm.summary(m1)

#plot results
TDCM::tdcm.plot(results1, attribute.names = c("Addition", "Subtraction",
"Multiplication", "Division"))
```

`tdcm.rules` *TDCM Condensation Rules*

Description

A *condensation rule* is a formula that states how different attributes combine to form an observed or latent response (Rupp, Templin, & Henson, 2010). The **TDCM** package includes support for `"LCDM"`, `"DINA"`, `"DINO"`, `"CRUM"`, `"RRUM"`, `"LCDM1"` for the LCDM with only main effects, `"LCDM2"` for the LCDM with two-way interactions, `"LCDM3"`, and so on. Evaluate `TDCM::tdcm.rules$TDCM` for a complete list of condensation rules supported by the **TDCM** package.

Usage

```
tdcm.rules
```

Format

An object of class `data.frame` with 15 rows and 2 columns.

References

Rupp, A. A., Templin, J., & Henson, R. (2010). *Diagnostic Measurement: Theory, Methods, and Applications*. New York: Guilford. ISBN: 9781606235430.

Examples

```
TDCM::tdcm.rules$TDCM
```

tdcm.score *DCM scoring function.*

Description

Function to score responses with fixed item parameters from a previously calibrated LCDM.

Usage

```
tdcm.score(
  calibration.model,
  newdata,
  q.matrix,
  attr.prob.fixed = NULL,
  progress = TRUE
)
```

Arguments

`calibration.model`
: the previously calibrated model; an object of class `gdina`.

`newdata`
: a required $N \times I$ matrix. Binary item responses are in the columns.

`q.matrix`
: a required $I \times A$ matrix indicating which items measure which attributes.

`attr.prob.fixed`
: optional argument for attribute profile proportions. Default is uniform distribution of profiles.

`progress`
: An optional logical indicating whether the function should print the progress of estimation.

Details

Obtain classifications for new responses to items that were previously calibrated. The calibrate-and-score approach is further detailed in Madison et al. (2023).

Value

An object of class `gdina` with entries as indicated in the CDM package.

References

George, A. C., Robitzsch, A., Kiefer, T., Gross, J., & Ünlü , A. (2016). The R package CDM for cognitive diagnosis models. *Journal of Statistical Software, 74*(2), 1-24.

Henson, R., Templin, J., & Willse, J. (2009). Defining a family of cognitive diagnosis models using log linear models with latent variables. *Psychometrika, 74*, 191-21.

Madison, M.J., Chung, S., Kim, J., & Bradshaw, L. (2023). Approaches to estimating longitudinal diagnostic classification models. *Behaviormetrika*.

Examples

```
## Example 1: T = 2, A = 4
data(data.tdcm01, package = "TDCM")
dat1 <- data.tdcm01$data
qmat1 <- data.tdcm01$q.matrix
pre <- dat1[, 1:20]
post <- dat1[, 21:40]

# calibrate LCDM with post-test data
m1 <- CDM::gdina(data = pre, q.matrix = qmat1, linkfct = "logit", method = "ML")

# score pre-test responses
m2 <- TDCM::tdcm.score(m1, newdata = post, q.matrix = qmat1)
summary(m2)
m2$pattern
```

`tdcm.summary` *TDCM results compiler and summarizer.*

Description

Function to summarize results obtained with the `tdcm` function. It includes information regarding the item parameters, attribute posterior probabilities, transition posterior probabilities, attribute mastery classifications, growth, growth effects,transition probabilities, attribute correlations, model fit statistics, and several transition reliability metrics developed by Schellman and Madison (2024).

Usage

```
tdcm.summary(
  model,
  transition.option = 1,
  classthreshold = 0.5,
  attribute.names = c()
)
```

Arguments

`model` A `tdcm` object returned from the `tdcm` function.

`transition.option`
An optional argument to specify how growth and transition probabilities should be reported for each attribute across time points.

- `transition.option` = 1 (default): Summarizes the transition probabilities by comparing the first and last time point.
- `transition.option` = 2: Summarizes the transition by comparing the first time point to every subsequent time point.

- `transition.option = 3`: Summarizes the transition probabilities by comparing each consecutive time point sequentially.

`classthreshold` A numeric value between 0 and 1 specifying the probability threshold for determining examinees' proficiency based on the posterior probabilities.

- The default value is `.50`, which optimizes overall classification accuracy.
- Lower values reduce the probability of false negatives, such that fewer proficient examinees are misclassified as non-proficient.
- Higher values reduce the probability of false positives, such that fewer non-proficient examinees are misclassified as proficient.

`attribute.names`
An optional character `vector` specifying the attribute names to be included in the results outputs. By default, `attribute.names=NULL`, which uses the generic attribute labels from the Q-matrix.

Value

A list with the following items:

- `$item.parameters`: Item parameter estimates from the specified DCM.
- `$growth`: Proficiency proportions for each time point and each attribute.
- `$growth.effects`: It includes three growth effect size metrics for each attribute and specified transitions:
  1. **Growth**: Difference in proficiency proportions between the later and earlier time point.
  2. **Odds Ratio**: Ratio between the proficiency odds at the later time point and the proficiency odds at the earlier time point.
  3. **Cohen's h** (Cohen, 1988): Arcsine-transformed difference in proficiency proportions.

  Note that the `growth.effect` output directly depend on the option specified in `transition.option`.

  Example:

  Suppose a test measures two attributes at three time points. Because there are more than two time points, the growth effect output is calculated based on the option specified in `transition.option`.

  – If `transition.option=1`, the growth effect for Attribute 1 and 2 is computed between Time Point 1 (first) and Time Point 3 (last).
  – If `transition.option=2`, the growth effect for Attribute 1 and 2 is computed between:
    * Time Point 1 (first) and Time Point 2 (latter).
    * Time Point 1 (first) and Time Point 3 (latter).
  – If `transition.option=3`, the growth effect for Attribute 1 and 2 is obtained between:
    * Time Point 1 (earlier) and Time Point 2 (next).
    * Time Point 2 (earlier), and Time Point 3 (next).
- `$transition.probabilities`: Conditional attribute proficiency transition probability matrices.
- `$posterior.probabilities`: Examinee marginal attribute posterior probabilities of proficiency.
- `$transition.posteriors`: Examinee marginal attribute transition posterior probabilities.

- `$most.likely.transitions`: Examinee most likely transitions for each attribute and transition.
- `$classifications`: Examinee classifications determined by the specified threshold applied to the posterior probabilities.
- `$reliability`: Estimated transition reliability metrics for each attribute for the specified transitions option specified (Madison, 2019; Schellman & Madison, 2024). It includes seven metrics:
  - **pt bis**: Longitudinal point biserial metric, which reflects the ratio between the estimated attribute proficiency base rates with the attribute proficiency posterior probabilities.
  - **info gain**: Longitudinal information gain metric. It quantifies how much additional information is gained regarding an attribute's transition over time.
  - **polychor**: Longitudinal tetrachoric metric. It quantifies how consistently an examinee transitions between mastery states between two time points.
  - **ave max tr**: Average maximum transition posterior metric. It quantities how likely an examinee is classified into a specific transition state over time.
  - **P(t > k)**: Proportion of examinees whose marginal attribute transition posteriors exceed a threshold k. The thresholds used are $k = 0.6$, 0.7, 0.8, and 0.9, representing the proportion of examinees with attribute transition posterior probabilities greater than these values. For example, if P(t>.6) = 0.90, 90% of examinees have a posterior probability greater than 0.6.
  - **wt pt bis**: Weighted longitudinal point biserial. A variation of the longitudinal point biserial metric that computes the correlation between true attribute transition classification and observed marginal transition probabilities. It assigns greater weight to more prevalent attribute transitions based on each attributes' transition base rate, ensuring that transitions occurring more frequently in the data contribute more significantly to the computed reliability value.
  - **wt info gain**: Weighted longitudinal information gain. A variation of the longitudinal information gain that quantifies the additional information provided by the attribute transition posterior probabilities in predicting examinees' true transition status. It assigns greater weight to more prevalent attribute transitions, ensuring that transitions occurring more frequently in the data contribute more significantly to the computed reliability value.
- `$att.corr`: Estimated attribute correlation matrix.
- `$model.fit`: Several model fit indices and tests are output including:
  - Item root mean square error of approximation (RMSEA; von Davier, 2005).
  - Mean item RMSEA.
  - Bivariate item fit statistics (Chen et al., 2013).
  - Absolute fit statistics such as mean absolute deviation for observed.
  - Expected item correlations (MADcor; DiBello, Roussos, & Stout, 2007).
  - Standardized root mean square root of squared residuals (SRMSR; Maydeu-Olivares, 2013).

References

Chen, J., de la Torre, J., & Zhang, Z. (2013). Relative and absolute fit evaluation in cognitive diagnosis modeling. *Journal of Educational Measurement, 50*, 123-140.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

DiBello, L. V., Roussos, L. A., & Stout, W. F. (2007). *Review of cognitively diagnostic assessment and a summary of psychometric models*. In C. R. Rao and S. Sinharay (Eds.), Handbook of Statistics, Vol. 26 (pp.979–1030). Amsterdam: Elsevier.

Johnson, M. S., & Sinharay, S. (2020). The reliability of the posterior probability of skill attainment in diagnostic classification models. *Journal of Educational Measurement, 47*(1), 5 – 31.

Madison, M. J. (2019). Reliably assessing growth with longitudinal diagnostic classification models. *Educational Measurement: Issues and Practice, 38*(2), 68-78.

Maydeu-Olivares, A. (2013). Goodness-of-fit assessment of item response theory models (with discussion). *Measurement: Interdisciplinary Research and Perspectives, 11*, 71-137.

Schellman, M., & Madison, M. J. (2024). Estimating the reliability of skill transition in longitudinal DCMs. *Journal of Educational and Behavioral Statistics*.

Templin, J., & Bradshaw, L. (2013). Measuring the reliability of diagnostic classification model examinee estimates. *Journal of Classification, 30*, 251-275.

von Davier M. (2008). A general diagnostic model applied to language testing data. *The British journal of mathematical and statistical psychology, 61*(2), 287–307.

Examples

```
############################################################################
# Example 1: TDCM with full measurement invariance and equal Q-matrix
############################################################################

# Load data: T = 2, A = 4
data(data.tdcm01, package = "TDCM")
dat1 <- data.tdcm01$data
qmat1 <- data.tdcm01$q.matrix

# Estimate model

model1 <- TDCM::tdcm(dat1, qmat1, num.time.points = 2, invariance = TRUE, rule = "LCDM")

# summarize results with tdcm.summary function
results1 <- TDCM::tdcm.summary(model1, transition.option = 1)
results1$item.parameters
results1$growth
results1$growth.effects
results1$transition.probabilities
results1$reliability
head(results1$most.likely.transitions)
results1$model.fit$Item.RMSEA

############################################################################
# Example 2: TDCM with full measurement invariance and different Q-matrices
############################################################################

# Load dataset: T=3, A=2
data(data.tdcm03, package = "TDCM")
```

```
data <- data.tdcm03$data
q1 <- data.tdcm03$q.matrix.1
q2 <- data.tdcm03$q.matrix.2
q3 <- data.tdcm03$q.matrix.3
q <- data.tdcm03$q.matrix.stacked

# Estimate model
model2 <- TDCM::tdcm(data, q, num.time.points = 3,
                    rule = "LCDM",
                    num.q.matrix = 3,
                    num.items = c(10,10,10))

#----------------------------------------------------------------------------
# With different transition options
#----------------------------------------------------------------------------

## a) If transition.option = 1

results2_option1 <- TDCM::tdcm.summary(model2)
results2_option1$transition.probabilities
#, , Attribute 1: Time 1 to Time 3
#
#       T3 [0] T3 [1]
#T1 [0]  0.202  0.798
#T1 [1]  0.146  0.854
#
#, , Attribute 2: Time 1 to Time 3
#
#       T3 [0] T3 [1]
#T1 [0]  0.325  0.675
#T1 [1]  0.257  0.743

results2_option1$reliability
#            pt bis info gain polychor ave max tr P(t>.6) P(t>.7) P(t>.8)
#Attribute 1  0.550     0.387    0.737      0.830   0.888   0.780   0.643
#Attribute 2  0.665     0.474    0.808      0.851   0.899   0.801   0.694

# b) If transition.option = 2

# Summary with transition.option = 2
results2_option2 <- TDCM::tdcm.summary(model2, transition.option = 2)
results2_option2$transition.probabilities
#, , Attribute 1: Time 1 to Time 2
#
#.    [0]   [1]
#[0] 0.510 0.490
#[1] 0.424 0.576
#
#, , Attribute 2: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.456 0.544
#[1] 0.334 0.666
```

```
#
#, , Attribute 1: Time 1 to Time 3
#
#     [0]   [1]
#[0] 0.202 0.798
#[1] 0.146 0.854
#
#, , Attribute 2: Time 1 to Time 3
#
#     [0]   [1]
#[0] 0.325 0.675
#[1] 0.257 0.743

results2_option2$reliability
#, , T1 to T2
#
#.           pt bis info gain polychor ave max tr P(t>.6) P(t>.7) P(t>.8)
#Attribute 1  0.586     0.444    0.770      0.796   0.828   0.710   0.581
#Attribute 2  0.692     0.503    0.838      0.853   0.885   0.799   0.713
#
#, , T1 to T3
#
#.           pt bis info gain polychor ave max tr P(t>.6) P(t>.7) P(t>.8)
#Attribute 1  0.550     0.387    0.737      0.830   0.888   0.780   0.643
#Attribute 2  0.665     0.474    0.808      0.851   0.899   0.801   0.694

## c) If transition.option = 3

results2_option3 <- TDCM::tdcm.summary(model2, transition.option = 3)
results2_option3$transition.probabilities
#, , Attribute 1: Time 1 to Time 2
#
#    [0]   [1]
#[0] 0.510 0.490
#[1] 0.424 0.576
#
#, , Attribute 2: Time 1 to Time 2
#
#     [0]   [1]
#[0] 0.456 0.544
#[1] 0.334 0.666
#
#, , Attribute 1: Time 2 to Time 3
#
#     [0]   [1]
#[0] 0.183 0.817
#[1] 0.188 0.812
#
#, , Attribute 2: Time 2 to Time 3
#
#     [0]   [1]
#[0] 0.361 0.639
#[1] 0.262 0.738
```

```
results2_option3$reliability
#, , T1 to T2
#
#.          pt bis info gain polychor ave max tr P(t>.6) P(t>.7) P(t>.8)
#Attribute 1  0.586     0.444    0.770      0.796   0.828   0.710   0.581
#Attribute 2  0.692     0.503    0.838      0.853   0.885   0.799   0.713
#
#, , T2 to T3
#
#.           pt bis info gain polychor ave max tr P(t>.6) P(t>.7) P(t>.8)
#Attribute 1  0.537     0.396    0.724      0.801   0.841   0.724   0.578
#Attribute 2  0.691     0.502    0.861      0.853   0.880   0.799   0.714

#----------------------------------------------------------------------------
# With different thresholds
#----------------------------------------------------------------------------

## a) If classthreshold = 0.5 (default)

results2_1 <- TDCM::tdcm.summary(model2)
head(results2_1$posterior.probabilities)
#    T1A1  T1A2  T2A1  T2A2  T3A1  T3A2
# 1 0.068 0.882 0.961 0.967 1.000 1.000
# 2 0.001 0.010 0.845 0.749 0.070 0.402
# 3 0.005 0.683 0.816 0.395 0.987 0.133
# 4 0.007 0.988 0.996 0.998 0.993 0.997
# 5 0.001 0.000 0.205 0.019 0.999 0.814
# 6 0.011 0.001 0.630 0.004 0.900 0.077

head(results2_1$classifications)
#   T1A1 T1A2 T2A1 T2A2 T3A1 T3A2
# 1    0    1    1    1    1    1
# 2    0    0    1    1    0    0
# 3    0    1    1    0    1    0
# 4    0    1    1    1    1    1
# 5    0    0    0    0    1    1
# 6    0    0    1    0    1    0

## b) If classthreshold = 0.7

results2_2 <- TDCM::tdcm.summary(model2, classthreshold = 0.7)
head(results2_2$posterior.probabilities)
#    T1A1  T1A2  T2A1  T2A2  T3A1  T3A2
# 1 0.068 0.882 0.961 0.967 1.000 1.000
# 2 0.001 0.010 0.845 0.749 0.070 0.402
# 3 0.005 0.683 0.816 0.395 0.987 0.133
# 4 0.007 0.988 0.996 0.998 0.993 0.997
# 5 0.001 0.000 0.205 0.019 0.999 0.814
# 6 0.011 0.001 0.630 0.004 0.900 0.077

head(results2_2$classifications)
#   T1A1 T1A2 T2A1 T2A2 T3A1 T3A2
```

```
# 1    0    1    1    1    1    1
# 2    0    0    1    1    0    0
# 3    0    0    1    0    1    0
# 4    0    1    1    1    1    1
# 5    0    0    0    0    1    1
# 6    0    0    0    0    1    0

## c) If classthreshold = 0.3

results2_3 <- TDCM::tdcm.summary(model2, classthreshold = 0.3)
head(results2_3$posterior.probabilities)
#    T1A1  T1A2  T2A1  T2A2  T3A1  T3A2
# 1 0.068 0.882 0.961 0.967 1.000 1.000
# 2 0.001 0.010 0.845 0.749 0.070 0.402
# 3 0.005 0.683 0.816 0.395 0.987 0.133
# 4 0.007 0.988 0.996 0.998 0.993 0.997
# 5 0.001 0.000 0.205 0.019 0.999 0.814
# 6 0.011 0.001 0.630 0.004 0.900 0.077

head(results2_3$classifications)
#   T1A1 T1A2 T2A1 T2A2 T3A1 T3A2
# 1    0    1    1    1    1    1
# 2    0    0    1    1    0    1
# 3    0    1    1    1    1    0
# 4    0    1    1    1    1    1
# 5    0    0    0    0    1    1
# 6    0    0    1    0    1    0
```

Index