Package 'TSQLEM'

September 28, 2026

Title Two Stage Estimation for Generalized Structural Equation Models

Version 0.1.0

Description Provides a framework to estimate high dimensional generalized structural equation models using two stage quasi-likelihood expectation-maximization. The structural model supports binomial (logit and probit), Poisson, negative binomial, and gamma distributions for the outcome variable. Hattab (2026) ``A Two Stage Quasi-Likelihood Estimation Method for High Dimensional Generalized Structural Equation Models'' <doi:10.48550/arXiv.2608.16017>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 8.0.0

Depends R (>= 4.6.0)

Imports lavaan, Matrix, MASS, mvtnorm

Suggests psych

NeedsCompilation no

Author Mohammad Hattab [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-0100-740X>)

Maintainer Mohammad Hattab <mwhattab@unm.edu>

Repository CRAN

Date/Publication 2026-09-28 08:10:08 UTC

Contents

`CondLatent` *Sample the Full Conditional Distribution of the Latent Space*

Description

Extracts the full conditional distribution of the latent variables given the observed indicators and the structural outcome via Markov Chain Monte Carlo (MCMC).

Usage

```
CondLatent(
  df,
  y_name,
  betahat,
  phi = 1,
  family,
  link,
  scores,
  cov_scores,
  nsim = 10000,
  individual_index = NULL,
  method = "MH",
  mh_burnin = 2000,
  mh_tune_c = 1
)
```

Arguments

`df`	A data frame containing the observed indicators and the outcome variable. Must contain complete cases.
`y_name`	A character string specifying the name of the outcome variable.
`betahat`	A numeric vector of the estimated structural parameters (output from TSQLEM).
`phi`	A numeric scalar for the dispersion parameter (output from TSQLEM). Defaults to 1 for binomial/poisson.
`family`	A character string specifying the GLM family.
`link`	A character string specifying the link function.
`scores`	A matrix of the Stage 1 empirical Bayes expected latent scores (output from TSQLEM).
`cov_scores`	A matrix of the Stage 1 conditional covariance of the latent space (output from TSQLEM).
`nsim`	An integer specifying the number of retained MCMC draws. Defaults to 10000.
`individual_index`	A numeric vector specifying which rows (individuals) to sample. Defaults to all rows.

method	A character string specifying the sampling algorithm: "MH" (Metropolis-Hastings) or "SIR" (Sampling Importance Resampling).
mh_burnin	An integer specifying the burn-in period for the MH algorithm.
mh_tune_c	A numeric tuning scalar to adjust the MH acceptance rate (target 0.23 to 0.44).

Value

A list of individual-level data containing the conditional distribution of the latent space given the observed data, MCMC draws, and the MH acceptance rate.

See Also

TSQLEM

Examples

```
# 1. PREP THE DATA
# ========================================
library(psych)
data(bfi)
df <- na.omit(bfi[, c(1:25, 26)]) # Must be complete cases

# 2. DEFINE MEASUREMENT MODEL
# ========================================
model_cfa <-  '
  Agree =~ A1 + A2 + A3 + A4 + A5
  Cons  =~ C1 + C2 + C3 + C4 + C5
  Extra =~ E1 + E2 + E3 + E4 + E5
  Neuro =~ N1 + N2 + N3 + N4 + N5
  Open  =~ O1 + O2 + O3 + O4 + O5'

# 3. RUN TSQLEM ALGORITHM
# ========================================
Fit <- TSQLEM(df = df,
              y_name = "gender",
              lav_syn_cfa = model_cfa,
              B = 5000,
              num.it = 20,
              tol = 10^-5,
              cfa_estimator = "ML",
              family = "binomial",
              link = "probit",
              DebiasBern = TRUE,
              probit_exact = TRUE)

print(Fit$beta_hat)

# 4. SAMPLE FULL CONDITIONAL LATENT DISTRIBUTIONS
# ========================================
# Extract counterfactual/conditional densities for individuals 10, 376, and 1057
mcmc_results <- CondLatent(df = df,
                           y_name = "gender",
```

```
                                betahat = Fit$beta_hat,
                                phi = Fit$dispersion_hat, # Inherits 1 for binomial
                                family = "binomial",
                                link = "probit",
                                scores = Fit$scores,
                                cov_scores = Fit$cov_scores,
                                nsim = 10^4,
                                individual_index = c(10, 376, 1057),
                                method = "MH",
                                mh_burnin =3000,
                                mh_tune_c = 1.25)

# Review acceptance rates and the mean of the conditional distributions
print(mcmc_results[["10"]]$mh_accept_rate)
print(mcmc_results[["10"]]$conditional_means)
```

TSQLEM *Two-Stage Quasi-Likelihood Estimation for Generalized Structural Equation Models*

Description

Fits high-dimensional generalized structural equation models (GSEMs) utilizing a Two-Stage Quasi-Likelihood Expectation-Maximization (TSQLEM) framework.

Usage

```
TSQLEM(
  df,
  y_name,
  lav_syn_cfa,
  B,
  start_beta = NA,
  num.it = 10,
  tol = 10^-4,
  cfa_estimator = "ML",
  family = "binomial",
  link = "logit",
  DebiasBern = TRUE,
  probit_exact = TRUE
)
```

Arguments

df	A data frame containing the observed indicators and the outcome variable. Rows with missing values are omitted internally.
y_name	A character string specifying the name of the outcome variable in the dataset (must be binary for binomial families).

lav_syn_cfa	A character string specifying the measurement model syntax in lavaan format.
B	An integer specifying the number of Monte Carlo draws for the E-step when simulation is required.
start_beta	A numeric vector of initial starting values for the structural parameters. Defaults to NA (auto-initialized).
num.it	An integer specifying the maximum number of iterations. Defaults to 10.
tol	A numeric value specifying the convergence tolerance threshold. Defaults to 10^-4.
cfa_estimator	A character string specifying the lavaan estimator for Stage 1. Defaults to "ML".
family	A character string specifying the generalized linear model family ("binomial", "poisson", "gamma", "negbin"). Defaults to "binomial".
link	A character string specifying the link function ("logit", "probit", "log"). Defaults to "logit".
DebiasBern	A logical indicating whether to apply a second-order Delta Method correction to debias weights for Bernoulli outcomes when MC simulations are utilized. Defaults to TRUE.
probit_exact	A logical indicating whether to use exact analytical integration for the probit link. Defaults to TRUE.

Details

The algorithm operates in two modular stages:

1. **Stage 1:** Fits a measurement model using lavaan to extract latent factor scores and their conditional covariance matrix.
2. **Stage 2:** Employs marginal quasi-likelihood estimating equations via an iteratively reweighted least squares (IRLS) scheme, utilizing either exact analytical integration or Monte Carlo simulations.

For Gamma and Negative Binomial models, extra dispersion parameters are estimated internally via profile working models using standard variance parameterizations:

- **Gamma:** Modeled with Var(y) = dispersion * mean^2.
- **Negative Binomial:** Modeled with Var(y) = mean + (mean^2 / dispersion).

Supported outcome distributions and link functions include:

- family = "binomial": paired with link = "logit" or link = "probit"
- family = "poisson": paired with link = "log"
- family = "negbin": paired with link = "log"
- family = "gamma": paired with link = "log"

Value

A list containing:

- `beta_hat`: The final estimated structural parameters.
- `dispersion_hat`: The estimated dispersion parameter (applicable to Gamma/Negative Binomial).
- `BREAK`: An integer indicating if the algorithm did not coverage (1) or ran normally (0).
- `Num.it`: The number of iterations completed.
- `eb_beta`: The empirical Bayes starting values extracted during initialization.
- `beta_history`: A matrix tracking parameter estimates across iterations.
- `gradient_history`: A vector tracking the gradient norm across iterations.
- `scores`: A matrix containing the Stage 1 empirical Bayes expected latent factor scores for each complete case.
- `cov_scores`: A matrix representing the approximate conditional covariance of the latent space, extracted from the measurement model.

Author(s)

Mohammad W. Hattab

References

Hattab, M. W. (2026). A Two Stage Quasi-Likelihood Estimation Method for High Dimensional Generalized Structural Equation Models. arXiv:2608.16017. doi:10.48550/arXiv.2608.16017

See Also

`CondLatent`

Examples

```
# 1. PREP THE DATA
# ========================================
library(psych)
data(bfi)
df <- bfi[, c(1:25, 26)]
df <- na.omit(df)
model_cfa <- '
   Agree =~ A1 + A2 + A3 + A4 + A5
   Cons  =~ C1 + C2 + C3 + C4 + C5
   Extra =~ E1 + E2 + E3 + E4 + E5
   Neuro =~ N1 + N2 + N3 + N4 + N5
   Open  =~ O1 + O2 + O3 + O4 + O5'

Fit <- TSQLEM(df = df,
              y_name = "gender",
              lav_syn_cfa = model_cfa,
              B = 5000,
```

```
                num.it = 20,
                tol = 10^-5,
                cfa_estimator = "ML",
                family = "binomial",
                link = "probit",
                DebiasBern = TRUE,
                probit_exact = TRUE)

print(Fit$beta_hat)
```

Index