

Package: hdmed (via r-universe)

September 13, 2024

Title Methods for Mediation Analysis with High-Dimensional Mediators

Version 1.0.1

Maintainer Dylan Clark-Boucher <dc.larkboucher@fas.harvard.edu>

Description A suite of functions for performing mediation analysis with high-dimensional mediators. In addition to centralizing code from several existing packages for high-dimensional mediation analysis, we provide organized, well-documented functions for a handle of methods which, though programmed their original authors, have not previously been formalized into R packages or been made presentable for public use. The methods we include cover a broad array of approaches and objectives, and are described in detail by both our companion manuscript---`Methods for Mediation Analysis with High-Dimensional DNA Methylation Data: Possible Choices and Comparison"---and the original publications that proposed them. The specific methods offered by our package include the Bayesian sparse linear mixed model (BSLMM) by Song et al. (2019); high-dimensional mediation analysis (HDMA) by Gao et al. (2019); high-dimensional multivariate mediation (HDMM) by Chén et al. (2018); high-dimensional linear mediation analysis (HILMA) by Zhou et al. (2020); high-dimensional mediation analysis (HIMA) by Zhang et al. (2016); latent variable mediation analysis (LVMA) by Derkach et al. (2019); mediation by fixed-effect model (MedFix) by Zhang (2021); pathway LASSO by Zhao & Luo (2022); principal component mediation analysis (PCMA) by Huang & Pan (2016); and sparse principal component mediation analysis (SPCMA) by Zhao et al. (2020). Citations for the corresponding papers can be found in their respective functions.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 2.10)

LazyData true

Imports bama ($\geq 1.3.0$), foreach ($\geq 1.5.2$), freebird (≥ 1.0), gcdnet ($\geq 1.0.6$), genlasso ($\geq 1.6.1$), hdi ($\geq 0.1.9$), iterators ($\geq 1.0.14$), MASS, mediation ($\geq 4.5.0$), ncvreg ($\geq 3.13.0$), stats, utils

NeedsCompilation no

Author Dylan Clark-Boucher [aut, cre], Mike Kleinsasser [aut]

Repository CRAN

Date/Publication 2024-05-15 02:40:07 UTC

Contents

mediate_bslmm	2
mediate_hdma	4
mediate_hdmm	6
mediate_hilma	8
mediate_hima	9
mediate_lvma	11
mediate_medfix	13
mediate_pcma	15
mediate_plasso	17
mediate_spcma	19
med_dat	21

Index 22

mediate_bslmm	<i>Bayesian Sparse Linear Mixed Model</i>
---------------	---

Description

mediate_bslmm fits the Bayesian sparse linear mixed model proposed by Song et al. (2020) for high-dimensional mediation analysis, estimating the mediation contributions of potential mediators.

Usage

```
mediate_bslmm(
  A,
  M,
  Y,
  C1 = NULL,
  C2 = C1,
  burnin = 30000,
  ndraws = 5000,
  ci_level = 0.95,
  weights = NULL,
  k = 2,
```

```

lm0 = 1e-04,
lm1 = 1,
lma1 = 1,
l = 1
)

```

Arguments

A	length n numeric vector containing exposure variable
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector containing continuous outcome variable.
C1	optional numeric matrix of covariates to include in the outcome model.
C2	optional numeric matrix of covariates to include in the mediator model. Default is C1.
burnin	number of MCMC draws prior to sampling.
ndraws	number of MCMC draws after burn-in.
ci_level	the desired credible interval level. Default is 0.95.
weights	optional numeric vector of observation weights.
k	shape parameter for the inverse gamma priors. Default is 2.
lm0	scale parameter for the inverse gamma prior on the variance of the smaller-variance normal components. Default is 1e-4. If k=2, this parameter equals the prior mean on the smaller normal variance.
lm1	scale parameter for the inverse gamma prior on the variance of the larger-variance components of beta_m. Default is 1. If k=2, this parameter equals the prior mean on the larger normal variance of the mediator-outcome associations.
lma1	scale parameter for the inverse gamma prior on the variance of the larger-variance components of alpha_a. Default is 1. If k=2, this parameter equals the prior mean on the larger normal variance of the exposure-mediator associations.
l	scale parameter for the other inverse gamma priors.

Details

mediate_bslmm is a wrapper function for the "BSLMM" option from `bama::bama()`, which fits a Bayesian sparse linear mixed model for performing mediation analysis with high-dimensional mediators. The model assumes that the mediator-outcome associations (β_m) and the exposure-mediator associations (α_a) independently follow a mixture of small-variance and high-variance normal distributions, and that if a mediator M_j has both $(\beta_m)_j$ and $(\alpha_a)_j$ belonging to the larger-variance distribution, it has a notably large mediation contribution compared to the others. The posterior inclusion probability (PIP) of belonging to both larger-variance distributions is reported for each mediator as `ab_pip`.

Value

A list containing:

- `contributions`: a data frame containing the estimates, Bayesian credible intervals, and posterior inclusion probabilities of the mediation contributions
- `effects`: a data frame containing the estimated direct, global mediation, and total effects.

References

Song, Y. et al. Bayesian shrinkage estimation of high dimensional causal mediation effects in omics studies. *Biometrics* 76, 700-710 (2020).

Examples

```
A <- med_dat$A
M <- med_dat$M
Y <- med_dat$Y

# Toy example with small burnin and ndraws
out <- mediate_bslmm(A, M, Y, burnin = 100, ndraws = 10)
out$effects
head(out$contributions)
```

mediate_hdma

High-Dimensional Mediation Analysis

Description

mediate_hdma fits a high-dimensional mediation model with the de-biased LASSO approach as proposed by Gao et al. (2022), estimating the mediation contributions of potential mediators.

Usage

```
mediate_hdma(
  A,
  M,
  Y,
  C1 = NULL,
  C2 = NULL,
  binary_y = FALSE,
  n_include = NULL,
  ...
)
```

Arguments

A	length n numeric vector containing exposure variable
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector containing continuous or binary outcome variable.

C1	optional numeric matrix of covariates to include in the outcome model.
C2	optional numeric matrix of covariates to include in the mediator model.
binary_y	logical flag for whether Y should be interpreted as a binary variable with 1/0 coding rather than as continuous. Default is FALSE.
n_include	integer specifying the number of top markers from sure independent screening to be included. Default is NULL, in which case n_include will be either $\text{ceiling}(n/\log(n))$ if <code>binary_Y = F</code> , or $\text{ceiling}(n/(2*\log(n)))$ if <code>binary_Y = T</code> . If <code>n_include >= p</code> , all mediators are included with no screening. Note that if <code>binary_y = F</code> , screening is performed based on the single-mediator outcome model p-values, and if <code>binary_y = T</code> , screening is based on the the mediator model p-values.
...	other arguments passed to <code>hdi::hdi()</code> .

Details

The first step in HDMA is to perform sure independence screening (SIS) to choose the `n_include` mediators that are most associated with the outcome (when Y is continuous) or the exposure (when Y is binary), based on p-values from linear regression. The second step is to fit the outcome model for the remaining mediators using de-sparsified (A.K.A de-biased) LASSO, which has asymptotic properties allowing for computation of p-values by the `hdi` package. HDMA then fits the mediator models using linear regression among those mediators that have both survived SIS (in step 1) and been identified by the LASSO (in step 2), obtaining p-values for the mediation contributions by taking the maximum of the α_a and β_m p-values. The global indirect effect is estimated by summing the mediation contributions, and the direct effect is estimated by subtracting the global indirect effect from an estimate of the total effect. See References for more detail.

Value

A list containing:

- `contributions`: a data frame containing the estimates and p-values of the mediation contributions.
- `effects`: a data frame containing the estimated direct, global mediation, and total effects.

Source

<https://github.com/YuzhaoGao/High-dimensional-mediation-analysis-R>

References

- Gao, Y. et al. Testing Mediation Effects in High-Dimensional Epigenetic Studies. *Front. Genet.* 10, 1195 (2019).
- Fan, J. & Lv, J. Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc.* 70, 849-911 (2008)

Examples

```

A <- med_dat$A
M <- med_dat$M
Y <- med_dat$Y

# Fit hdma with continuous outcomes
out <- mediate_hdma(A, M, Y)
head(out$contributions)
out$effects

```

mediate_hdmm

High-dimensional Multivariate Mediation Analysis with Principal Directions of Mediation

Description

mediate_hdmm estimates the first "direction of mediation" in the causal mediation mechanism of an exposure A, an outcome Y, and high-dimensional mediators M as proposed by Chén et al. (2018).

Usage

```

mediate_hdmm(
  A,
  M,
  Y,
  sims = 1000,
  boot_ci_type = "bca",
  ci_level = 0.95,
  tol = 10^-5,
  theta = rep(1, 5),
  w1 = rep(1, ncol(M)),
  interval = 10^6,
  step = 10^4,
  imax = 100
)

```

Arguments

A	numeric vector containing exposure variable.
M	numeric matrix of high-dimensional mediators. It is not recommended to supply a matrix with more mediators than observations.
Y	numeric vector containing continuous outcome variable.
sims	number of Monte Carlo draws for nonparametric bootstrap or quasi-Bayesian approximation. See mediate . Default is 1000.

boot_ci_type	a character string indicating the type of bootstrap confidence intervals for when boot = TRUE. If "bca", bias-corrected and accelerated (BCa) confidence intervals will be estimated. If "perc", percentile confidence intervals will be estimated (see <code>mediation::mediate()</code>). Default is "bca".
ci_level	the designated confidence level. Default 0.95.
tol	tolerance. Default 10^{-5} .
theta	numeric vector of length 5 describing starting value of pathway coefficients. Default is a vector of 1's.
w1	numeric vector of the same length of A specifying PDM starting values. Default is a vector of 1's.
interval	numeric vector proportional to the intervals from where the smoothing parameter is searched. Default is 10^6 .
step	numerical number specifying step width for smoothing parameter search. Default is 10^4 .
imax	integer specifying the maximum number of iterations allowed. Default is 100.

Details

HDMM provides latent variable approach to high-dimensional mediation analysis. The function `mediate_hdmm` uses a likelihood-based approach to compute principal directions of mediation (PDMs), which are loading weights used to linearly combine the inputted mediators to form a single, latent variable that replaces the original mediators in the analysis. Though HDMM cannot be used to estimate the global mediation effect or the contributions of specific mediators, it can still be useful for inferring whether there is mediation occurring through the set of mediators as a joint system. See the provided reference for more details.

Value

A list containing:

- `pdm`: the first direction of mediation by which mediators are weighted.
- `mediator`: the latent mediator corresponding to the first direction of mediation.
- `effects`: a data frame containing the estimates, confidence intervals, and p-values of the mediation effects.

Source

<https://github.com/oliverychen/PDM>

References

Chén, O. Y. et al. High-dimensional multivariate mediation with application to neuroimaging data. *Biostatistics* 19, 121-136 (2018).

Examples

```
A <- as.numeric(scale(med_dat$A)) # can help to standardize
M <- scale(med_dat$M[,1:8])
Y <- as.numeric(scale(med_dat$Y))

out <- mediate_hdmm(A, M, Y, sims = 5, tol = 10^-3, imax = 50)
out$effects
```

mediate_hilma

High-Dimensional Linear Mediation Analysis

Description

mediate_hilma applies high-dimensional linear mediation analysis (HILMA) as proposed by Zhou et al. (2020).

Usage

```
mediate_hilma(
  A,
  M,
  Y,
  aic_tuning = FALSE,
  nlambda = 5,
  lambda_minmax_ratio = 0.1,
  center = TRUE
)
```

Arguments

A	length n numeric vector containing a single exposure variable or size n x q numeric matrix containing multiple exposures. If a matrix, should be low-dimensional ($q \ll n$).
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector containing continuous outcome variable.
aic_tuning	logical flag for whether to select the tuning parameter using AIC. Default is FALSE as this was not the preferred approach by the HILMA authors (see references for more detail).
nlambda	number of candidate lambdas for AIC tuning. Default is 5. If aic_tuning=F this parameter is ignored.
lambda_minmax_ratio	ratio of the minimum lambda attempted in AIC tuning to the maximum. If aic_tuning=F, ignored.
center	logical flag for whether the variables should be centered. Default is TRUE.

Details

mediate_hilma is a wrapper function for the `freebird::hilma()` function, which fits the "high-dimensional linear mediation analysis" model proposed by Zhou et al. (2020) for mediation settings when there are high-dimensional mediators and one or several exposures. The function returns estimates of the direct effect, total effect, and global mediation effect, the last of which is tested for statistical significance with a reported p-value. For additional detail, see the attached reference as well as the `freebird::hilma()` documentation.

Value

A list containing, for each exposure, a data frame of the estimated direct, total, and global mediation effects. A p-value is provided for the global mediation effect.

References

Zhou, R. R., Wang, L. & Zhao, S. D. Estimation and inference for the indirect effect in high-dimensional linear mediation models. *Biometrika* 107, 573-589 (2020)

Examples

```
A <- med_dat$A
M <- med_dat$M
Y <- med_dat$Y

# Implement HILMA with one exposure
out <- mediate_hilma(A, M, Y)
out$a1
```

mediate_hima

High-Dimensional Mediation Analysis

Description

mediate_hima fits a high-dimensional mediation model with the minimax concave penalty as proposed by Zhang et al. (2016), estimating the mediation contributions of potential mediators.

Usage

```
mediate_hima(
  A,
  M,
  Y,
  C1 = NULL,
  C2 = NULL,
  binary_y = FALSE,
```

```

    n_include = NULL,
    ...
  )

```

Arguments

A	length n numeric vector containing exposure variable
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector containing continuous or binary outcome variable.
C1	optional numeric matrix of covariates to include in the outcome model.
C2	optional numeric matrix of covariates to include in the mediator model.
binary_y	logical flag for whether Y should be interpreted as a binary variable with 1/0 coding rather than as continuous. Default is FALSE.
n_include	integer specifying the number of top markers from sure independent screening to be included. Default is NULL, in which case n_include will be either $\text{ceiling}(n/\log(n))$ if binary_y = F, or $\text{ceiling}(n/(2*\log(n)))$ if binary_y = T. If n_include >= p, all mediators are included with no screening. Note that if binary_y = F, screening is performed based on the single-mediator outcome model p-values, and if binary_y = F, screening is based on the the mediator model p-values.
...	other arguments passed to hdi .

Details

The first step in HIMA is to perform sure independence screening (SIS) to choose the n_include mediators that are most associated with the outcome (when Y is continuous) or the exposure (when Y is binary), based on p-values from linear regression. The second step is to fit the outcome model for the remaining mediators with the minimax concave penalty. HIMA then fits the mediator models using linear regression among those mediators that have both survived SIS (in step 1) and been selected by the MCP (in step 2), which enables estimation of the mediation contributions. The global indirect effect is estimated by summing these contributions, and the direct effect is estimated by subtracting the global indirect effect from an estimate of the total effect. We compute p-values for the mediation contributions by taking the maximum of the α_a and β_m p-values, where the beta p-values are obtained via a second, unpenalized generalized linear model containing only the mediators selected by the MCP. We include this p-value computation so that our function replicates the behavior of the HIMA function from [HIMA](#) package, the function on which ours is based, but we caution that the beta p-values may be over-optimistic due to double-dipping, since the mediators tested in the unpenalized model are only those chosen by the penalized model. Note also that the HIMA authors apply Bonferroni correction to the final, maxed p-values to account for multiple testing, which we choose to leave up to the user. For more information, see the "HIMA" R package along with the provided reference.

Value

A list containing:

- `contributions`: a data frame containing the estimates and p-values of the mediation contributions.

- effects: a data frame containing the estimated direct, global mediation, and total effects.

References

Zhang, H. et al. Estimating and testing high-dimensional mediation effects in epigenetic studies. *Bioinformatics* 32, 3150-3154 (2016).

Examples

```
A <- med_dat$A
M <- med_dat$M
Y <- med_dat$Y

# Fit hima with continuous outcome
out <- mediate_hima(A, M, Y)
head(out$contributions)
out$effects

# Fit hima with binary outcome
Y1 <- as.numeric(Y > mean(Y))
out1 <- mediate_hima(A, M, Y1, binary_y = TRUE)
head(out1$contributions)
out1$effects
```

mediate_lvma

Latent Variable Mediation Analysis

Description

mediate_lvma fits a high-dimensional mediation model described by Derkach et al. (2019), in which a small number of latent, unmeasured mediators replace the original mediators in the model.

Usage

```
mediate_lvma(A, M, Y, q, rhoLM, rhoEL, rhoLY, scale = TRUE, imax = 5000)
```

Arguments

A	length n numeric vector representing the exposure variable
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector representing the continuous outcome variable.
q	number of latent mediators
rhoLM	numeric vector of candidate penalty parameters for the latent mediator-mediator associations in the joint likelihood. Default is a short toy sequence.
rhoEL	numeric vector of candidate penalty parameters for the exposure-latent mediator associations in the joint likelihood. Default is a short toy sequence.

rhoLY	numeric vector of candidate penalty parameters for the latent mediator-outcome associations in the joint likelihood. Default is a short toy sequence.
scale	logical flag for whether the inputted mediators should be standardized prior to the analysis. Default is FALSE, but TRUE may be worth attempting in case of errors.
imax	integer specifying the maximum number of iterations allowed. Default is 5000.

Details

LVMA is a latent variable mediation model which assumes, contrary to standard assumptions, that the inputted set of candidate mediators do not affect the outcome through the exposure on their own, but rather, occur as result of latent, unmeasured mediators which themselves transmit effects from the exposure to outcome. The required parameters for fitting this model are rhoLE, a regularization parameter for effects of the latent mediators on the inputted mediators; rhoEL, a regularization parameter for the effects of the exposure on the latent mediators; and rhoLY, a regularization parameter for the effects of the latent mediators on the exposure. These parameters should ideally be supplied by the user as vectors, so that each combination of the three parameters can be attempted in the estimation. However, this can be intensely computation costly, and for simplicity our default values are vectors of length 4, corresponding to a 64 by 64 parameter grid. In practice, Derkach et al. use a much larger grid with 5 values of rhoLM (ranging from 6 to 8.5), 40 values of rhoEY (ranging from 0 to 40), and 40 values of rhoLY (ranging from 0 to 75). Supplying longer parameter vectors makes the fit more flexible, but more computationally costly, and to reliably implement LVMA on real data one should use a larger parameter grid with parallel computation on a remote computing cluster, as did the authors. For more information on the likelihood, parameters, and mediation model, see the referenced article and/or its supplement files.

Value

A list containing the selected models based on AIC, BIC, and EBIC (recommended) as three sub-lists. The sub-lists include objects indicating the penalty set that was used (`penalty`), the values of the chosen parameters (e.g., EBIC), the exposure-latent mediator effects (`AL_effects`), the latent mediator-mediator effects (`LM_effects`, a data frame), the direct effect of the exposure on the outcome (`AY_direct_effect`), the the latent mediator-outcome effects (`LY_effects`), and binary vector indicating whether each mediator was determined to be active. Here, active mediators are those which are associated with a latent mediator that itself is associated with both A and Y.

Source

<https://pubmed.ncbi.nlm.nih.gov/30859548/>

References

Derkach, A., Pfeiffer, R. M., Chen, T.-H. & Sampson, J. N. High dimensional mediation analysis with latent variables. *Biometrics* 75, 745-756 (2019).

Examples

```
A <- med_dat$A
M <- med_dat$M
```

```

Y <- med_dat$Y

# Perform latent variable mediation analysis with 4 latent mediators and print
# whether the original 20 mediators are "actively" related to mediation
out <- mediate_lvma(A, M, Y, q = 4, rhoLM = 2, rhoEL = 2, rhoLY = 2, imax = 50)
table(out$EBIC_out$mediator_active)

```

mediate_medfix

Mediation Analysis via Fixed Effects Model

Description

mediate_medfix fits a high-dimensional mediation model with the adaptive LASSO approach as proposed by Zhang (2021) for the special case of MedFix that there is only one exposure variable.

Usage

```

mediate_medfix(
  A,
  M,
  Y,
  C1 = NULL,
  C2 = C1,
  nlambda = 100,
  nlambda2 = 50,
  nfold = 10,
  seed = 1
)

```

Arguments

A	length n numeric vector representing the exposure variable
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector representing the continuous outcome variable.
C1	optional numeric matrix of covariates to include in the outcome model. Default is NULL.
C2	optional numeric matrix of covariates to include in the mediator model. Default is C1.
nlambda	number of lambdas attempted in the adaptive LASSO. See <code>gcdnet::cv.gcdnet()</code> . The specific sequence of lambdas is chosen by the <code>cv.gcdnet()</code> function.
nlambda2	number of lambda2s attempted in the initial elastic net used for computing adaptive weights prior to adaptive LASSO. Default is 50. See <code>gcdnet::cv.gcdnet()</code> for details on elastic net regression. If 0, lambda2=0 is fed to <code>cv.gcdnet()</code> and the initial fit is based on standard LASSO, not elastic net. If not 0, the specific sequence of lambda2s is given by <code>exp(seq(1e-4, 0.02, length.out=nlambda2))</code> , and the lambda2 with the least cross-validated error is chosen.

nfolds	number of folds for cross-validation. See <code>gcdnet::cv.gcdnet()</code> . Default is 10.
seed	numeric random seed.

Details

MedFix performs mediation analysis when there are multiple mediators by applying adaptive LASSO to the outcome model. In order to fit the adaptive LASSO, we first obtain an initial model fit using either LASSO (which deploys the L1 penalty) or elastic net (which deploys both the L1 and L2 penalties) depending on the provided `nlambda2`. Estimates from this fit are then used to compute the adaptive weights used in the adaptive LASSO. Once the final adaptive LASSO estimates (β_m) are obtained for the outcome model, estimates for the p mediator models (α_a) are obtained by linear regression. The mediation contributions are computed as α_a times β_m , and the p-value is taken as the maximum of the α_a and β_m p-values. Last, the global indirect effect is estimated by summing the mediation contributions, and the direct effect is estimated by subtracting the global indirect effect from an estimate of the total effect. This function is specific to applying MedFix to the special case that there is only one exposure; for details on how to apply MedFix when the exposures are high-dimensional, as proposed by the authors, see the supplemental files of the referenced manuscript.

Value

A list containing:

- `contributions`: a data frame containing the estimates and p-values of the mediation contributions.
- `effects`: a data frame containing the estimated direct, global mediation, and total effects.

References

Zhang, Q. High-Dimensional Mediation Analysis with Applications to Causal Gene Identification. *Stat. Biosci.* 14, 432-451 (2021).

Examples

```
A <- med_dat$Y
M <- med_dat$M
Y <- med_dat$Y

out <- mediate_medfix(A, M, Y, nlambda = 10, nlambda2 = 5, seed = 1)
out$effects
head(out$contributions)
```

mediate_pcma	<i>Principal Component Mediation Analysis for High-dimensional Mediators</i>
--------------	--

Description

mediate_pcma applies principal component mediation analysis (Huang and Pan, 2013) to mediation settings in which the mediators are high-dimensional.

Usage

```
mediate_pcma(
  A,
  M,
  Y,
  var_per = 0.8,
  n_pc = NULL,
  sims = 1000,
  boot_ci_type = "bca",
  ci_level = 0.95,
  seed = 1
)
```

Arguments

A	length n numeric vector containing exposure variable
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector containing continuous outcome variable.
var_per	a numeric variable with the desired proportion of variance explained. Default is 0.8.
n_pc	optional numeric variable with the desired number of PCs, in which case var_per is ignored. Default is NULL and the number of PCs is determined based on the desired proportion of variance explained.
sims	number of Monte Carlo draws for nonparametric bootstrap or quasi-Bayesian approximation (see mediation::mediate()). Default is 1000.
boot_ci_type	a character string indicating the type of bootstrap confidence intervals for when boot = TRUE. If "bca", bias-corrected and accelerated (BCa) confidence intervals will be estimated. If "perc", percentile confidence intervals will be estimated (see mediation::mediate()). Default is "bca".
ci_level	the desired confidence level. Default is 0.95.
seed	seed used for fitting single-mediator models after PCA

Details

Principal component mediation analysis (PCMA) is a method for estimating mediation effects when the mediators are high-dimensional. The first step is to compute the residuals of mediator models ($M|A$), then perform PCA on those residuals to reduce them to a smaller number of mediators that efficiently explain the residual variance. Then, since those mediators are linearly independent conditional on A , one can trivially perform single-mediator mediation analysis for each PC on its own, in this case by using the `mediation::mediate()` function. The global mediation effect is estimated by summing the mediation effects of the individual PCs.

Value

A list containing:

- `loadings`: a matrix of the PC loadings.
- `pcs`: a matrix of the PCs.
- `var_explained`: the cumulative proportion of variance explained by the PCs.
- `contributions`: a data frame containing the estimates, confidence intervals, and p-values of the mediation contributions.
- `effects`: a data frame containing the estimated direct, global mediation, and total effects.

Source

<https://rdr.io/github/zhaoyi1026/spcma>

References

Huang, Y.-T. & Pan, W.-C. Hypothesis test of mediation effect in causal mediation model with high-dimensional continuous mediators. *Biometrics* 72, 402-413 (2016).

Examples

```
A <- med_dat$A
M <- med_dat$M
Y <- med_dat$Y
# Fit PCMA with 3 principal components and print the effects. In practice one
# should choose n_pc (or var_per) and the number sims to be larger
out <- mediate_pcma(A, M, Y, n_pc = 3, sims = 10)
out$effects
```

mediate_lasso	<i>Pathway LASSO for Mediation Analysis with High-Dimensional Mediators</i>
---------------	---

Description

mediate_lasso fits a high-dimensional mediation model with the penalized likelihood described by Zhao and Luo (2022), estimating the mediation contributions of each site.

Usage

```
mediate_lasso(
  A,
  M,
  Y,
  lambdas = NULL,
  select_lambda = FALSE,
  vss_rep = 5,
  vss_cutoff = 0.1,
  omega_ratio = 1,
  phi = 2,
  maxit = 5000,
  tol = 1e-06
)
```

Arguments

A	numeric vector containing exposure variable.
M	numeric matrix of high-dimensional mediators.
Y	numeric vector containing continuous outcome variable.
lambdas	numeric vector of tuning parameters λ for which model is fitted. Default is a vector of length 50 ranging from 10^{-5} to 10^4 , with more density in the lower range.
select_lambda	logical flag indicating whether to conduct a tuning parameter selection using the Variable Selection Stability Criterion described by Sun et al. (2013). Default is FALSE.
vss_rep	if select_lambda is TRUE, number of VSSC replications.
vss_cutoff	if select_lambda is TRUE, cutoff used for VSSC. Default is 0.1.
omega_ratio	ratio of the omega parameter to lambda parameter in the likelihood penalty. Default is 1.
phi	value of the phi parameter in the likelihood penalty. Default is 2. Cannot be less than 1/2.
maxit	the maximum number of iterations. Default is 5000.
tol	convergence tolerance. Default is 10^{-6} .

Details

Pathway LASSO fits a high-dimensional mediation model with a likelihood that directly penalizes the mediator-outcome effects, exposure-mediator effects, and mediation contributions (i.e., the mediation "pathways"). The shrinkage of the model is determined by three parameters— ϕ , ω , and λ —with higher values tending to result in more sparsity. Mediation results are returned for every unique value in the inputted `lambdas` argument, in increasing order, excluding potentially any for which the estimation was unsuccessful. For details on the exact likelihood, see the first reference.

When implementing multiple lambdas (i.e., using either the default argument to `lambdas` or specifying one's own vector), there is an option to perform a tuning parameter selection using the Variable Selection Stability Criterion proposed by Sun et al. (2013), which chooses the parameter for which the variable selection is most stable in repeated data-splitting. However, implementing this can be computationally costly, especially with a long list of tuning parameters, since it involves re-fitting pathway LASSO many times.

Value

A list containing:

- `lambdas`: the lambdas attempted in the same order as the objects in `all_fits`.
- `all_fits`: a list containing, for each lambda, a data.frame of the estimated mediation effects.
- `chosen_lambda`: if `select_lambda` is TRUE, the lambda chosen by VSSC.
- `chosen_fit`: if `select_lambda` is TRUE, the fit corresponding to the chosen lambda.
- `vss`: if `select_lambda` is TRUE, a data.frame containing the variable selection stabilities.

Source

<https://github.com/zhaoyi1026/PathwayLasso>

References

- Zhao, Y. & Luo, X. Pathway LASSO: pathway estimation and selection with high-dimensional mediators. *Stat. Interface* 15, 39-50 (2022).
- Sun, W., Wang, J. & Fang, Y. Consistent selection of tuning parameters via variable selection stability. *J. Mach. Learn. Res.* 14, 3419-3440 (2013).

Examples

```
A <- med_dat$A
M <- med_dat$M[,1:8]
Y <- med_dat$Y
# fit pathway LASSO for two tuning parameters and retrieve their fits
out <- mediate_plasso(A, M, Y, lambdas = c(10^-3, 10^-2), tol = 1e-4)
head(out$all_fits$lambda1)
head(out$all_fits$lambda2)
```

mediate_spcma	<i>Sparse Principal Component Mediation Analysis for High-Dimensional Mediators</i>
---------------	---

Description

mediate_spcma applies sparse principal component mediation analysis to mediation settings in which the mediators are high-dimensional.

Usage

```
mediate_spcma(
  A,
  M,
  Y,
  var_per = 0.8,
  n_pc = NULL,
  sims = 1000,
  boot_ci_type = "bca",
  ci_level = 0.95,
  fused = FALSE,
  gamma = 0,
  per_jump = 0.7,
  eps = 1e-04,
  maxsteps = 2000,
  seed = 1
)
```

Arguments

A	length n numeric vector containing exposure variable
M	n x p numeric matrix of high-dimensional mediators.
Y	length n numeric vector containing continuous outcome variable.
var_per	a numeric variable with the desired proportion of variance explained. Default is 0.8.
n_pc	optional numeric variable with the desired number of PCs, in which case var_per is ignored. Default is NULL and the number of PCs is determined based on the desired proportion of variance explained.
sims	number of Monte Carlo draws for nonparametric bootstrap or quasi-Bayesian approximation (see mediation::mediate()). Default is 1000.
boot_ci_type	character string indicating the type of bootstrap confidence intervals for when boot = TRUE. If "bca", bias-corrected and accelerated (BCa) confidence intervals will be estimated. If "perc", percentile confidence intervals will be estimated (see mediation::mediate()). Default is "bca".
ci_level	the designated confidence level. Default 0.95.

fused	logical variable for whether the fused LASSO should be used instead of the ordinary LASSO. Default is FALSE.
gamma	numeric variable ≥ 0 indicating the ratio of the standard LASSO penalty to the fusion penalty (see <code>genlasso::genlasso()</code>). Ignored if fused = FALSE. Default is 0, meaning there is no standard penalty. Larger values result in more shrinkage and sparser PCs.
per_jump	numeric value used for tuning parameter selection - the quantile cut-off for total variance change under different lambda values in the LASSO. Default is 0.7. Larger values result in more shrinkage and sparser PCs
eps	numeric variable indicating the multiplier for the ridge penalty in case X is rank deficient (see <code>genlasso::genlasso()</code>). Default is $1e-4$.
maxsteps	an integer specifying the maximum number of steps for the algorithm before termination (see <code>genlasso::genlasso()</code>). Default is 2000.
seed	seed used for fitting single-mediator models after PCA

Details

mediate_spcma performs principal component mediation analysis, comparable to mediate_pcma, with the modification that the PC loadings are sparsified by a flexible LASSO penalty. This has the potential make the PCs more interpretable, since, unlike in PCA, they are only linear combinations of a subset of mediators rather than all of them. The choice of LASSO penalties is determined by the fused argument - which, when set to TRUE, deploys a fused LASSO penalty that encourages the model to give consecutive mediators similar loadings. The default is fused = FALSE, and the standard LASSO penalty is used instead of the fusion penalty. Once the sparse PCs are computed, inference proceeds exactly like in PCMA, and the PC-mediators are evaluated with methods from the mediate package.

Value

A list containing:

- loadings: a matrix of the PC loadings.
- pcs: a matrix of the PCs.
- var_explained: the cumulative proportion of variance explained by the PCs.
- contributions: a data frame containing the estimates, confidence intervals, and p-values of the mediation contributions.
- effects: a data frame containing the estimated direct, global mediation, and total effects.

Source

<https://rdrr.io/github/zhaoyi1026/spcma>

References

Zhao, Y., Lindquist, M. A. & Caffo, B. S. Sparse principal component based high-dimensional mediation analysis. *Comput. Stat. Data Anal.* 142, 106835 (2020).

Examples

```
A <- med_dat$A
M <- med_dat$M
Y <- med_dat$Y

# Fit SPCMA with the fused LASSO penalty while choosing the number of PCs based
# on the variance they explain. In practice, var_per and sims should be higher.
out <- mediate_spcma(A, M, Y, var_per = 0.25, fused = TRUE, gamma = 2, sims = 10)
out$effects
```

med_dat

Mediation Example Dataset

Description

A toy dataset for demonstrating mediation analysis in settings where there are multiple mediators

Usage

med_dat

Format

med_dat:

A list with three items:

Y numeric vector containing the outcome variable

M numeric matrix of 20 continuous mediators

A numeric vector containing the exposure variable ...

Index

* datasets

med_dat, [21](#)

bama::bama(), [3](#)

freebird::hilma(), [9](#)

gcdnet::cv.gcdnet(), [13](#), [14](#)

genlasso::genlasso(), [20](#)

hdi, [10](#)

hdi::hdi(), [5](#)

med_dat, [21](#)

mediate, [6](#)

mediate_bslmm, [2](#)

mediate_hdma, [4](#)

mediate_hdmm, [6](#)

mediate_hilma, [8](#)

mediate_hima, [9](#)

mediate_lvma, [11](#)

mediate_medfix, [13](#)

mediate_pcma, [15](#)

mediate_plasso, [17](#)

mediate_spcma, [19](#)

mediation::mediate(), [7](#), [15](#), [16](#), [19](#)