Package 'MetaIntegration'

March 17, 2021

Type Package

Title Ensemble Meta-Prediction Framework

Version 0.1.2

Description An ensemble meta-prediction framework to integrate multiple regression models into a current study. Gu, T., Taylor, J.M.G. and Mukherjee, B. (2020) <arXiv:2010.09971>.
A meta-analysis framework along with two weighted estimators as the ensemble of empirical Bayes estimators, which combines the estimates from the different external models. The proposed framework is flexible and robust in the ways that (i) it is capable of incorporating external models that use a slightly different set of covariates; (ii) it is able to identify the most relevant external information and diminish the influence of information that is less compatible with the internal data; and (iii) it nicely balances the bias-variance trade-off while preserving the most efficiency gain. The proposed estimators are more efficient than the naive analysis of the internal data and other naive combinations of external estimators.

Maintainer Michael Kleinsasser <mkleinsa@umich.edu>

License GPL-2

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.5.0)

Imports Rsolnp, corpcor, MASS, knitr

URL https://github.com/umich-biostatistics/MetaIntegration

BugReports https://github.com/umich-biostatistics/MetaIntegration/issues

RoxygenNote 7.1.1

NeedsCompilation no

Author Tian Gu [aut],
Bhramar Mukherjee [aut],
Michael Kleinsasser [cre]

Repository CRAN

Date/Publication 2021-03-17 17:20:06 UTC

R topics documented:

`asympVar_LinReg` *Asymptotic variance-covariance matrix for gamma_Int and gamma_CML for linear regression (continuous outcome Y)*

Description

Asymptotic variance-covariance matrix for gamma_Int and gamma_CML for linear regression (continuous outcome Y)

Usage

```
asympVar_LinReg(
  k,
  p,
  q,
  YInt,
  XInt,
  BInt,
  gammaHatInt,
  betaHatExt_list,
  CovExt_list,
  rho,
  ExUncertainty
)
```

Arguments

`k`	number of external models
`p`	total number of X covariates including the intercept (i.e. p=ncol(X)+1)
`q`	total number of covariates including the intercept (i.e. q=ncol(X)+ncol(B)+1)
`YInt`	Outcome vector
`XInt`	X covariates that are used in the external models - Do not include intercept
`BInt`	Newly added B covariates that are not included in the external models

gammaHatInt	Internal parameter estimates of the full model using the internal data
betaHatExt_list	a list of k items, each item is a vector of the external parameter estimates (beta). Vector name is required for each covariate, and has to be as consistent as the full model
CovExt_list	a list of k items, each item is the variance-covariance matrix of the external parameter estimates (beta) of the reduced model
rho	a list of k items, each item is the sample size ratio, n/m (the internal sampel size n over the external sample size m)
ExUncertainty	logic indicator, if TRUE then considering the external model uncertainty in the algorithm; if FALSE then ignoring the external model uncertainty

Value

a list containing:

- "asyV.I" Variance of gamma_I (the direct regression parameter estimates using the internal data only)
- "asyV.CML" Variance of gamma_CML (the CML estiamtes (Chatterjee et al. 2016))
- "asyCov.CML" Covariance between two different CML estimates, gamma_CMLi and gamma_CMLj
- "asyCov.CML.I" Covariance between gamma_I and gamma_CML
- "ExtraTerm" the extra variance when ExUncertainty == TRUE (i.e. the external uncertainty is considered in the algorithm)

References

Chatterjee, N., Chen, Y.-H., P.Maas and Carroll, R. J. (2016). Constrained maximum likelihood estimation for model calibration using summary-level information from external big data sources. Journal of the American Statistical Association 111, 107-117.

Gu, T., Taylor, J.M.G. and Mukherjee, B. (2020). An ensemble meta-prediction framework to integrate multiple regression models into a current study. Manuscript in preparation.

Examples

```
# Full model: Y|X1, X2, B
# Reduced model 1: Y|X1 of sample size m1
# Reduced model 2: Y|X2 of sample size m2
# (X1, X2, B) follows normal distribution with mean zero, variance one and correlation 0.3
# Y|X1, X2, B follows N(-1-0.5*X1-0.5*X2+0.5*B, 1)
set.seed(2333)
n = 1000
data.n = data.frame(matrix(ncol = 4, nrow = n))
colnames(data.n) = c('Y', 'X1', 'X2', 'B')
data.n[,c('X1', 'X2', 'B')] = MASS::mvrnorm(n, rep(0,3), diag(0.7,3)+0.3)
data.n$Y = rnorm(n, -1 - 0.5*data.n$X1 - 0.5*data.n$X2 + 0.5*data.n$B, 1)

# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
```

```
# to obtain the beta estiamtes and the corresponsing estimated variance
m = m1 = m2 = 30000
data.m = data.frame(matrix(ncol = 4, nrow = m))
names(data.m) = c('Y', 'X1', 'X2', 'B')
data.m[,c('X1', 'X2', 'B')] = MASS::mvrnorm(m, rep(0,3), diag(0.7,3)+0.3)
data.m$Y = rnorm(m, -1 - 0.5*data.m$X1 - 0.5*data.m$X2 + 0.5*data.m$B, 1)

#fit Y|X to obtain the external beta estimates, save the beta estiamtes and the
# corresponding estimated variance
fit.E1 = lm(Y ~ X1, data = data.m)
fit.E2 = lm(Y ~ X2, data = data.m)
beta.E1 = coef(fit.E1)
beta.E2 = coef(fit.E2)
names(beta.E1) = c('int', 'X1')
names(beta.E2) = c('int', 'X2')
V.E1 = vcov(fit.E1)
V.E2 = vcov(fit.E2)

#Save all the external model information into lists for later use
betaHatExt_list = list(Ext1 = beta.E1, Ext2 = beta.E2)
CovExt_list = list(Ext1 = V.E1, Ext2 = V.E2)
rho = list(Ext1 = n/m1, Ext2 = n/m2)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = lm(Y ~ X1 + X2 + B, data = data.n)
gamma.I = coef(fit.gamma.I)

#Get CML estimates using internal data and the beta estimates from the external
# model 1 and 2, respectively
gamma.CML1 = fxnCC_LinReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X1,
                          BInt=cbind(data.n$X2, data.n$B), betaHatExt=beta.E1,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400,factor=1)[["gammaHat"]]
gamma.CML2 = fxnCC_LinReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X2,
                          BInt=cbind(data.n$X1, data.n$B), betaHatExt=beta.E2,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400, factor=1)[["gammaHat"]]
#It's important to reorder gamma.CML2 so that it follows the order (X1, X2, X3, B)
# as gamma.I and gamma.CML1
gamma.CML2 = c(gamma.CML2[1], gamma.CML2[3], gamma.CML2[2], gamma.CML2[4])

#Get Variance-covariance matricx of c(gamma.I, gamma.CML1, gamma.CML2)
asy.CML = asympVar_LinReg(k=2,
                          p=2,
                          q=4,
                          YInt=data.n$Y,
                          XInt=data.n[,c('X1','X2')],
                          #covariates that appeared in at least one external model
                      BInt=data.n$B,  #covariates that not used in any of the external models
                          gammaHatInt=gamma.I,
                          betaHatExt_list=betaHatExt_list,
                          CovExt_list=CovExt_list,
                          rho=rho,
```

```
                          ExUncertainty=TRUE)
asyV.I = asy.CML[["asyV.I"]]                       #variance of gamma.I
asyV.CML1 = asy.CML[["asyV.CML"]][[1]]             #variance of gamma.CML1
asyV.CML2 = asy.CML[["asyV.CML"]][[2]]             #variance of gamma.CML2
asyCov.CML1.I = asy.CML[["asyCov.CML.I"]][[1]]     #covariance of gamma.CML1 and gamma.I
asyCov.CML2.I = asy.CML[["asyCov.CML.I"]][[2]]     #covariance of gamma.CML2 and gamma.I
asyCov.CML12 = asy.CML[["asyCov.CML"]][["12"]]   #covariance of gamma.CML1 and gamma.CML2
```

`asympVar_LogReg` *Asymptotic variance-covariance matrix for gamma_Int and gamma_CML for logistic regression (binary outcome Y)*

Description

Asymptotic variance-covariance matrix for gamma_Int and gamma_CML for logistic regression (binary outcome Y)

Usage

```
asympVar_LogReg(
  k,
  p,
  q,
  YInt,
  XInt,
  BInt,
  gammaHatInt,
  betaHatExt_list,
  CovExt_list,
  rho,
  ExUncertainty
)
```

Arguments

`k`	number of external models
`p`	total number of all X covariates that is used at least once in the external model, including the intercept (i.e. p=ncol(X)+1)
`q`	total number of covariates including the intercept (i.e. q=ncol(X)+ncol(B)+1)
`YInt`	Outcome vector
`XInt`	X covariates that are used in the external models - Do not include intercept
`BInt`	Newly added B covariates that are not included in the external models
`gammaHatInt`	Internal parameter estimates of the full model using the internal data

`betaHatExt_list`
: a list of k items, each item is a vector of the external parameter estimates (beta). Vector name is required for each covariate, and has to be as consistent as the full model

`CovExt_list`
: a list of k items, each item is the variance-covariance matrix of the external parameter estimates (beta) of the reduced model

`rho`
: a list of k items, each item is the sample size ratio, n/m (the internal sampel size n over the external sample size m)

`ExUncertainty`
: logic indicator, if TRUE then considering the external model uncertainty in the algorithm; if FALSE then ignoring the external model uncertainty

Value

a list containing:

- "asyV.I" Variance of gamma_I (the direct regression parameter estimates using the internal data only)
- "asyV.CML" Variance of gamma_CML (the CML estiamtes (Chatterjee et al. 2016))
- "asyCov.CML" Covariance between two different CML estimates, gamma_CMLi and gamma_CMLj
- "asyCov.CML.I" Covariance between gamma_I and gamma_CML
- "ExtraTerm" the extra variance when ExUncertainty == TRUE (i.e. the external uncertainty is considered in the algorithm)

References

Chatterjee, N., Chen, Y.-H., P.Maas and Carroll, R. J. (2016). Constrained maximum likelihood estimation for model calibration using summary-level information from external big data sources. Journal of the American Statistical Association 111, 107-117.

Gu, T., Taylor, J.M.G. and Mukherjee, B. (2020). An ensemble meta-prediction framework to integrate multiple regression models into a current study. Manuscript in preparation.

Examples

```
# Full model: Y|X1, X2, B
# Reduced model 1: Y|X1 of sample size m1
# Reduced model 2: Y|X2 of sample size m2
# (X1, X2, B) follows normal distribution with mean zero, variance one and correlation 0.3
# Y|X1, X2, B follows Bernoulli[expit(-1-0.5*X1-0.5*X2+0.5*B)], where expit(x)=exp(x)/[1+exp(x)]
set.seed(2333)
n = 1000
data.n = data.frame(matrix(ncol = 4, nrow = n))
colnames(data.n) = c('Y', 'X1', 'X2', 'B')
data.n[,c('X1', 'X2', 'B')] = MASS::mvrnorm(n, rep(0,3), diag(0.7,3)+0.3)
data.n$Y = rbinom(n, 1, expit(-1 - 0.5*data.n$X1 - 0.5*data.n$X2 + 0.5*data.n$B))

# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
# to obtain the beta estiamtes and the corresponsing estimated variance
```

```
m = m1 = m2 = 30000
data.m = data.frame(matrix(ncol = 4, nrow = m))
names(data.m) = c('Y', 'X1', 'X2', 'B')
data.m[,c('X1', 'X2', 'B')] = MASS::mvrnorm(m, rep(0,3), diag(0.7,3)+0.3)
data.m$Y = rbinom(m, 1, expit(-1 - 0.5*data.m$X1 - 0.5*data.m$X2 + 0.5*data.m$B))

#fit Y|X to obtain the external beta estimates, save the beta estiamtes and
# the corresponsing estimated variance
fit.E1 = glm(Y ~ X1, data = data.m, family = binomial(link='logit'))
fit.E2 = glm(Y ~ X2, data = data.m, family = binomial(link='logit'))
beta.E1 = coef(fit.E1)
beta.E2 = coef(fit.E2)
names(beta.E1) = c('int', 'X1')
names(beta.E2) = c('int', 'X2')
V.E1 = vcov(fit.E1)
V.E2 = vcov(fit.E2)

#Save all the external model information into lists for later use
betaHatExt_list = list(Ext1 = beta.E1, Ext2 = beta.E2)
CovExt_list = list(Ext1 = V.E1, Ext2 = V.E2)
rho = list(Ext1 = n/m1, Ext2 = n/m2)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = glm(Y ~ X1 + X2 + B, data = data.n, family = binomial(link='logit'))
gamma.I = coef(fit.gamma.I)

#Get CML estimates using internal data and the beta estimates from the external
# model 1 and 2, respectively
gamma.CML1 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X1,
                          BInt=cbind(data.n$X2, data.n$B), betaHatExt=beta.E1,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400,factor=1)[["gammaHat"]]
gamma.CML2 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X2,
                          BInt=cbind(data.n$X1, data.n$B), betaHatExt=beta.E2,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400, factor=1)[["gammaHat"]]
#It's important to reorder gamma.CML2 so that it follows the order (X1, X2, X3, B)
# as gamma.I and gamma.CML1
gamma.CML2 = c(gamma.CML2[1], gamma.CML2[3], gamma.CML2[2], gamma.CML2[4])

#Get Variance-covariance matricx of c(gamma.I, gamma.CML1, gamma.CML2)
asy.CML = asympVar_LogReg(k=2,
                          p=2,
                          q=4,
                          YInt=data.n$Y,
                          XInt=data.n[,c('X1','X2')], #covariates that appeared
                          # in at least one external model
                      BInt=data.n$B,  #covariates that not used in any of the external models
                          gammaHatInt=gamma.I,
                          betaHatExt_list=betaHatExt_list,
                          CovExt_list=CovExt_list,
                          rho=rho,
                          ExUncertainty=TRUE)
```

```
asyV.I = asy.CML[["asyV.I"]]                        #variance of gamma.I
asyV.CML1 = asy.CML[["asyV.CML"]][[1]]              #variance of gamma.CML1
asyV.CML2 = asy.CML[["asyV.CML"]][[2]]              #variance of gamma.CML2
asyCov.CML1.I = asy.CML[["asyCov.CML.I"]][[1]]    #covariance of gamma.CML1 and gamma.I
asyCov.CML2.I = asy.CML[["asyCov.CML.I"]][[2]]    #covariance of gamma.CML2 and gamma.I
asyCov.CML12 = asy.CML[["asyCov.CML"]][["12"]]   #covariance of gamma.CML1 and gamma.CML2
```

`expit` *Expit*

Description

Standard expit function

Usage

```
expit(x)
```

Arguments

`x`	data to transform

Details

y = exp(x) / (1 + exp(x))

Value

vector of transformed data

`fxnCC_LinReg` *Constraint maximum likelihood (CML) method for linear regression (continuous outcome Y)*

Description

Constraint maximum likelihood (CML) method for linear regression (continuous outcome Y)

Usage

```
fxnCC_LinReg(
  p,
  q,
  YInt,
  XInt,
  BInt,
  betaHatExt,
  gammaHatInt,
  n,
  tol,
  maxIter,
  factor
)
```

Arguments

p	total number of X covariates including the intercept (i.e. p=ncol(X)+1)
q	total number of covariates including the intercept (i.e. q=ncol(X)+ncol(B)+1)
YInt	Outcome vector
XInt	X covariates that are used in the external models - Do not include intercept
BInt	Newly added B covariates that are not included in the external models
betaHatExt	External parameter estimates of the reduced model
gammaHatInt	Full model parameter estimates using the internal data only
n	internal data sample size
tol	convergence criteria e.g. 1e-6
maxIter	Maximum number of iterations to reach convergence e.g. 400
factor	the step-halving factor between 0 and 1, if factor=1 then newton-raphson method; decrease if algorithm cannot converge given the maximum iterations

Value

gammaHat, in the order (intercept, XInt, BInt)

References

Chatterjee, N., Chen, Y.-H., P.Maas and Carroll, R. J. (2016). Constrained maximum likelihood estimation for model calibration using summary-level information from external big data sources. Journal of the American Statistical Association 111, 107-117.

Examples

```
# Full model: Y|X, B
# Reduced model: Y|X
# X,B follows normal distribution with mean zero, variance one and correlation 0.3
# Y|X, B follows N(-1-0.5X+0.5B, 1)
```

```
set.seed(2333)
n = 800
data.n = data.frame(matrix(ncol = 3, nrow = n))
colnames(data.n) = c('Y', 'X', 'B')
data.n[,c('X', 'B')] = MASS::mvrnorm(n, rep(0,2), diag(0.7,2)+0.3)
data.n$Y = rnorm(n, -1 - 0.5*data.n$X + 0.5*data.n$B, 1)

# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
# to obtain the beta estiamtes and the corresponsing estimated variance
m = 30000
data.m = data.frame(matrix(ncol = 3, nrow = m))
names(data.m) = c('Y', 'X', 'B')
data.m[,c('X', 'B')] = MASS::mvrnorm(m, rep(0,2), diag(0.7,2)+0.3)
data.m$Y = rnorm(m, -1 - 0.5*data.m$X + 0.5*data.m$B, 1)

#fit Y|X to obtain the external beta estimates, save the beta estiamtes and the
# corresponsing estimated variance
fit.E = lm(Y ~ X, data = data.m)
beta.E = coef(fit.E)
names(beta.E) = c('int', 'X')
V.E = vcov(fit.E)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = lm(Y ~ X + B, data = data.n)
gamma.I = coef(fit.gamma.I)

#Get CML estimates
gamma.CML = fxnCC_LinReg(p=2,
                         q=3,
                         YInt=data.n$Y,
                         XInt=data.n['X'],
                         BInt=data.n['B'],
                         betaHatExt=beta.E,
                         gammaHatInt=gamma.I,
                         n=nrow(data.n),
                         tol=1e-8,
                         maxIter=400,
                         factor=1)[["gammaHat"]]
```

`fxnCC_LogReg` *Constraint maximum likelihood (CML) method for logistic regression (binary outcome Y)*

Description

Constraint maximum likelihood (CML) method for logistic regression (binary outcome Y)

Usage

```
fxnCC_LogReg(
  p,
  q,
  YInt,
  XInt,
  BInt,
  betaHatExt,
  gammaHatInt,
  n,
  tol,
  maxIter,
  factor
)
```

Arguments

p	total number of X covariates including the intercept (i.e. p=ncol(X)+1)
q	total number of covariates including the intercept (i.e. q=ncol(X)+ncol(B)+1)
YInt	Outcome vector
XInt	X covariates that are used in the external models - Do not include intercept
BInt	Newly added B covariates that are not included in the external models
betaHatExt	External parameter estimates of the reduced model
gammaHatInt	Full model parameter estimates using the internal data only
n	internal data sample size
tol	convergence criteria e.g. 1e-6
maxIter	Maximum number of iterations to reach convergence e.g. 400
factor	the step-halving factor between 0 and 1, if factor=1 then newton-raphson method; decrease if algorithm cannot converge given the maximum iterations

Value

gammaHat, in the order (intercept, XInt, BInt)

References

Chatterjee, N., Chen, Y.-H., P.Maas and Carroll, R. J. (2016). Constrained maximum likelihood estimation for model calibration using summary-level information from external big data sources. Journal of the American Statistical Association 111, 107-117.

Examples

```
# Full model: Y|X, B
# Reduced model: Y|X
# X,B follows normal distribution with mean zero, variance one and correlation 0.3
# Y|X, B follows N(-1-0.5X+0.5B, 1)
```

```
set.seed(2333)
n = 1000
data.n = data.frame(matrix(ncol = 3, nrow = n))
colnames(data.n) = c('Y', 'X', 'B')
data.n[,c('X', 'B')] = MASS::mvrnorm(n, rep(0,2), diag(0.7,2)+0.3)
data.n$Y = rbinom(n, 1, expit(-1 - 0.5*data.n$X + 0.5*data.n$B))

# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
# to obtain the beta estiamtes and the corresponsing estimated variance
m = 30000
data.m = data.frame(matrix(ncol = 3, nrow = m))
names(data.m) = c('Y', 'X', 'B')
data.m[,c('X', 'B')] = MASS::mvrnorm(m, rep(0,2), diag(0.7,2)+0.3)
data.m$Y = rbinom(m, 1, expit(-1 - 0.5*data.m$X + 0.5*data.m$B))

#fit Y|X to obtain the external beta estimates, save the beta estiamtes and the
# corresponsing estimated variance
fit.E = glm(Y ~ X, data = data.m, family = binomial(link='logit'))
beta.E = coef(fit.E)
names(beta.E) = c('int', 'X')
V.E = vcov(fit.E)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = glm(Y ~ X + B, data = data.n, family = binomial(link='logit'))
gamma.I = coef(fit.gamma.I)

#Get CML estimates
gamma.CML = fxnCC_LogReg(p=2,
                         q=3,
                         YInt=data.n$Y,
                         XInt=data.n$X,
                         BInt=data.n['B'],
                         betaHatExt=beta.E,
                         gammaHatInt=gamma.I,
                         n=nrow(data.n),
                         tol=1e-8,
                         maxIter=400,
                         factor=1)[["gammaHat"]]
```

`get_gamma_EB` *Calculate the empirical Bayes (EB) estimates*

Description

Calculate the empirical Bayes (EB) estimates

Usage

```
get_gamma_EB(gamma_I, gamma_CML, asyV.I)
```

Arguments

gamma_I	Full model parameter estimates using the internal data only (MLE from direct regression)
gamma_CML	Full model parameter estimates using the internal data and the external reduced model parameters (Chatterjee et al. 2016)
asyV.I	Variance-covariance matrix of gamma_I from function asympVar_LinReg() or asympVar_LogReg()

Value

a list with:

- "gamma_I" Full model parameter estimates using the internal data only (MLE from direct regression)
- "gamma_CML" Full model parameter estimates using the internal data and the external reduced model parameters (Chatterjee et al. 2016)
- "gamma_EB" The empirical Bayes estimate of the full model (i.e. a weighted average of gamma_I and gamma_CML) (Estes et al. 2017)

References

#' Chatterjee, N., Chen, Y.-H., P.Maas and Carroll, R. J. (2016). Constrained maximum likelihood estimation for model calibration using summary-level information from external big data sources. Journal of the American Statistical Association 111, 107-117.

Gu, T., Taylor, J.M.G. and Mukherjee, B. (2020). An ensemble meta-prediction framework to integrate multiple regression models into a current study. Manuscript in preparation.

Examples

```
# Full model: Y|X1, X2, B
# Reduced model 1: Y|X1 of sample size m1
# Reduced model 2: Y|X2 of sample size m2
# (X1, X2, B) follows normal distribution with mean zero, variance one and
#  correlation 0.3
# Y|X1, X2, B follows Bernoulli[expit(-1-0.5*X1-0.5*X2+0.5*B)],
#  where expit(x)=exp(x)/[1+exp(x)]
set.seed(2333)
n = 1000
data.n = data.frame(matrix(ncol = 4, nrow = n))
colnames(data.n) = c('Y', 'X1', 'X2', 'B')
data.n[,c('X1', 'X2', 'B')] = MASS::mvrnorm(n, rep(0,3), diag(0.7,3)+0.3)
data.n$Y = rbinom(n, 1, expit(-1 - 0.5*data.n$X1 - 0.5*data.n$X2 + 0.5*data.n$B))

# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
# to obtain the beta estiamtes and the corresponsing estimated variance
m = m1 = m2 = 30000
data.m = data.frame(matrix(ncol = 4, nrow = m))
```

```
names(data.m) = c('Y', 'X1', 'X2', 'B')
data.m[,c('X1', 'X2', 'B')] = MASS::mvrnorm(m, rep(0,3), diag(0.7,3)+0.3)
data.m$Y = rbinom(m, 1, expit(-1 - 0.5*data.m$X1 - 0.5*data.m$X2 + 0.5*data.m$B))

#fit Y|X to obtain the external beta estimates, save the beta estimates and the
# corresponding estimated variance
fit.E1 = glm(Y ~ X1, data = data.m, family = binomial(link='logit'))
fit.E2 = glm(Y ~ X2, data = data.m, family = binomial(link='logit'))
beta.E1 = coef(fit.E1)
beta.E2 = coef(fit.E2)
names(beta.E1) = c('int', 'X1')
names(beta.E2) = c('int', 'X2')
V.E1 = vcov(fit.E1)
V.E2 = vcov(fit.E2)

#Save all the external model information into lists for later use
betaHatExt_list = list(Ext1 = beta.E1, Ext2 = beta.E2)
CovExt_list = list(Ext1 = V.E1, Ext2 = V.E2)
rho = list(Ext1 = n/m1, Ext2 = n/m2)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = glm(Y ~ X1 + X2 + B, data = data.n, family = binomial(link='logit'))
gamma.I = coef(fit.gamma.I)

#Get CML estimates using internal data and the beta estimates from the external
# model 1 and 2, respectively
gamma.CML1 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X1,
                          BInt=cbind(data.n$X2, data.n$B), betaHatExt=beta.E1,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400,factor=1)[["gammaHat"]]
gamma.CML2 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X2,
                          BInt=cbind(data.n$X1, data.n$B), betaHatExt=beta.E2,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400, factor=1)[["gammaHat"]]
#It's important to reorder gamma.CML2 so that it follows the order (X1, X2, X3, B)
# as gamma.I and gamma.CML1
gamma.CML2 = c(gamma.CML2[1], gamma.CML2[3], gamma.CML2[2], gamma.CML2[4])

#Get Variance-covariance matricx of c(gamma.I, gamma.CML1, gamma.CML2)
asy.CML = asympVar_LogReg(k=2, p=2,q=4, YInt=data.n$Y, XInt=data.n[,c('X1','X2')],
                          BInt=data.n$B,  gammaHatInt=gamma.I,
                          betaHatExt_list=betaHatExt_list, CovExt_list=CovExt_list,
                          rho=rho, ExUncertainty=TRUE)
asyV.I = asy.CML[["asyV.I"]]

#Get the empirical Bayes (EB) estimates
gamma.EB1 = get_gamma_EB(gamma_I=gamma.I, gamma_CML=gamma.CML1, asyV.I=asyV.I)[['gamma.EB']]
gamma.EB2 = get_gamma_EB(gamma_I=gamma.I, gamma_CML=gamma.CML2, asyV.I=asyV.I)[['gamma.EB']]
```

get_OCW *Obtain the proposed Optimal Covariate-Weighted (OCW) estimates*

Description

Obtain the proposed Optimal Covariate-Weighted (OCW) estimates

Usage

```
get_OCW(k, q, data.XB, gamma.EB, V.EB)
```

Arguments

k	number of external models
q	total number of covariates (X,B) including the intercept (i.e. q=ncol(X)+ncol(B)+1)
data.XB	internal data (X,B)
gamma.EB	stack all k EB estimates in order, i.e. c(gamma.EB1,...,gamma.EBk)
V.EB	variance-covariance matrix obtained from function get_var_EB()

Value

return weights of gamma.EB's, final estimates of OCW estimates and the corresponding variance-covariance matrix

References

Reference: Gu, T., Taylor, J.M.G. and Mukherjee, B. (2020). An ensemble meta-prediction framework to integrate multiple regression models into a current study. Manuscript in preparation.

Examples

```
# Full model: Y|X1, X2, B
# Reduced model 1: Y|X1 of sample size m1
# Reduced model 2: Y|X2 of sample size m2
# (X1, X2, B) follows normal distribution with mean zero, variance one and correlation 0.3
# Y|X1, X2, B follows Bernoulli[expit(-1-0.5*X1-0.5*X2+0.5*B)], where expit(x)=exp(x)/[1+exp(x)]
set.seed(2333)
n = 1000
data.n = data.frame(matrix(ncol = 4, nrow = n))
colnames(data.n) = c('Y', 'X1', 'X2', 'B')
data.n[,c('X1', 'X2', 'B')] = MASS::mvrnorm(n, rep(0,3), diag(0.7,3)+0.3)
data.n$Y = rbinom(n, 1, expit(-1 - 0.5*data.n$X1 - 0.5*data.n$X2 + 0.5*data.n$B))

# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
# to obtain the beta estiamtes and the corresponsing estimated variance
m = m1 = m2 = 30000
```

```
data.m = data.frame(matrix(ncol = 4, nrow = m))
names(data.m) = c('Y', 'X1', 'X2', 'B')
data.m[,c('X1', 'X2', 'B')] = MASS::mvrnorm(m, rep(0,3), diag(0.7,3)+0.3)
data.m$Y = rbinom(m, 1, expit(-1 - 0.5*data.m$X1 - 0.5*data.m$X2 + 0.5*data.m$B))

#fit Y|X to obtain the external beta estimates, save the beta estimates and
# the corresponding estimated variance
fit.E1 = glm(Y ~ X1, data = data.m, family = binomial(link='logit'))
fit.E2 = glm(Y ~ X2, data = data.m, family = binomial(link='logit'))
beta.E1 = coef(fit.E1)
beta.E2 = coef(fit.E2)
names(beta.E1) = c('int', 'X1')
names(beta.E2) = c('int', 'X2')
V.E1 = vcov(fit.E1)
V.E2 = vcov(fit.E2)

#Save all the external model information into lists for later use
betaHatExt_list = list(Ext1 = beta.E1, Ext2 = beta.E2)
CovExt_list = list(Ext1 = V.E1, Ext2 = V.E2)
rho = list(Ext1 = n/m1, Ext2 = n/m2)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = glm(Y ~ X1 + X2 + B, data = data.n, family = binomial(link='logit'))
gamma.I = coef(fit.gamma.I)

#Get CML estimates using internal data and the beta estimates from the external
# model 1 and 2, respectively
gamma.CML1 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X1,
                          BInt=cbind(data.n$X2, data.n$B), betaHatExt=beta.E1,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400,factor=1)[["gammaHat"]]
gamma.CML2 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X2,
                          BInt=cbind(data.n$X1, data.n$B), betaHatExt=beta.E2,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400, factor=1)[["gammaHat"]]
#It's important to reorder gamma.CML2 so that it follows the order
# (X1, X2, X3, B) as gamma.I and gamma.CML1
gamma.CML2 = c(gamma.CML2[1], gamma.CML2[3], gamma.CML2[2], gamma.CML2[4])

#Get Variance-covariance matricx of c(gamma.I, gamma.CML1, gamma.CML2)
asy.CML = asympVar_LogReg(k=2, p=2,q=4, YInt=data.n$Y, XInt=data.n[,c('X1','X2')],
                    BInt=data.n$B,  gammaHatInt=gamma.I, betaHatExt_list=betaHatExt_list,
                          CovExt_list=CovExt_list, rho=rho, ExUncertainty=TRUE)

#Get the empirical Bayes (EB) estimates
gamma.EB1 = get_gamma_EB(gamma.I, gamma.CML1, asy.CML[["asyV.I"]])[["gamma.EB"]]
gamma.EB2 = get_gamma_EB(gamma.I, gamma.CML2, asy.CML[["asyV.I"]])[["gamma.EB"]]

#Get the asymptotic variance of the EB estimates
V.EB = get_var_EB(k=2, q=4, gamma.CML=c(gamma.CML1, gamma.CML2),
                  gamma.I = gamma.I, asy.CML=asy.CML, seed=2333, nsim=2000)

#Get the OCW estimates,  the corresponding variance-covariance matrix of the
```

```
# estimates and the weights of gamma.EB's
get_OCW(k=2,
        q=4,
        data.XB=data.n[,c('X1','X2','B')],
        gamma.EB=c(gamma.EB1, gamma.EB2),
        V.EB=V.EB)
```

`get_SCLearner` *Obtain the proposed Selective Coefficient-Learner (SC-Learner) estimates*

Description

Obtain the proposed Selective Coefficient-Learner (SC-Learner) estimates

Usage

```
get_SCLearner(k, q, pred.matrix, gamma.EB, V.EB)
```

Arguments

`k`	number of external models
`q`	total number of covariates (X,B) including the intercept (i.e. q=ncol(X)+ncol(B)+1)
`pred.matrix`	a predictor matrix (q rows by k columns) that specifies the full model variables in the rows, and the external models on the columns. An entry of 0 means that the row variable is NOT used in the column external model; 1 represents that it is used.
`gamma.EB`	bind all k EB estimates in order (q rows by k columns), i.e. cbind(gamma.EB1,...,gamma.EBk)
`V.EB`	variance-covariance matrix obtained from function get_var_EB()

Value

a list with gamma.SCLearner and var.SCLearner

Examples

```
# Full model: Y|X1, X2, B
# Reduced model 1: Y|X1 of sample size m1
# Reduced model 2: Y|X2 of sample size m2
# (X1, X2, B) follows normal distribution with mean zero, variance one and correlation 0.3
# Y|X1, X2, B follows Bernoulli[expit(-1-0.5*X1-0.5*X2+0.5*B)], where expit(x)=exp(x)/[1+exp(x)]
set.seed(2333)
n = 1000
data.n = data.frame(matrix(ncol = 4, nrow = n))
colnames(data.n) = c('Y', 'X1', 'X2', 'B')
data.n[,c('X1', 'X2', 'B')] = MASS::mvrnorm(n, rep(0,3), diag(0.7,3)+0.3)
data.n$Y = rbinom(n, 1, expit(-1 - 0.5*data.n$X1 - 0.5*data.n$X2 + 0.5*data.n$B))
```

```
# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
# to obtain the beta estiamtes and the corresponsing estimated variance
m = m1 = m2 = 30000
data.m = data.frame(matrix(ncol = 4, nrow = m))
names(data.m) = c('Y', 'X1', 'X2', 'B')
data.m[,c('X1', 'X2', 'B')] = MASS::mvrnorm(m, rep(0,3), diag(0.7,3)+0.3)
data.m$Y = rbinom(m, 1, expit(-1 - 0.5*data.m$X1 - 0.5*data.m$X2 + 0.5*data.m$B))

#fit Y|X to obtain the external beta estimates, save the beta estiamtes and
# the corresponsing estimated variance
fit.E1 = glm(Y ~ X1, data = data.m, family = binomial(link='logit'))
fit.E2 = glm(Y ~ X2, data = data.m, family = binomial(link='logit'))
beta.E1 = coef(fit.E1)
beta.E2 = coef(fit.E2)
names(beta.E1) = c('int', 'X1')
names(beta.E2) = c('int', 'X2')
V.E1 = vcov(fit.E1)
V.E2 = vcov(fit.E2)

#Save all the external model information into lists for later use
betaHatExt_list = list(Ext1 = beta.E1, Ext2 = beta.E2)
CovExt_list = list(Ext1 = V.E1, Ext2 = V.E2)
rho = list(Ext1 = n/m1, Ext2 = n/m2)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = glm(Y ~ X1 + X2 + B, data = data.n, family = binomial(link='logit'))
gamma.I = coef(fit.gamma.I)

#Get CML estimates using internal data and the beta estimates from the external
# model 1 and 2, respectively
gamma.CML1 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X1,
                          BInt=cbind(data.n$X2, data.n$B), betaHatExt=beta.E1,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400,factor=1)[["gammaHat"]]
gamma.CML2 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X2,
                          BInt=cbind(data.n$X1, data.n$B), betaHatExt=beta.E2,
                          gammaHatInt=gamma.I, n=nrow(data.n), tol=1e-8,
                          maxIter=400, factor=1)[["gammaHat"]]
#It's important to reorder gamma.CML2 so that it follows the order (X1, X2, X3, B)
# as gamma.I and gamma.CML1
gamma.CML2 = c(gamma.CML2[1], gamma.CML2[3], gamma.CML2[2], gamma.CML2[4])

#Get Variance-covariance matricx of c(gamma.I, gamma.CML1, gamma.CML2)
asy.CML = asympVar_LogReg(k=2, p=2,q=4, YInt=data.n$Y, XInt=data.n[,c('X1','X2')],
                    BInt=data.n$B,  gammaHatInt=gamma.I, betaHatExt_list=betaHatExt_list,
                          CovExt_list=CovExt_list, rho=rho, ExUncertainty=TRUE)

#Get the empirical Bayes (EB) estimates
gamma.EB1 = get_gamma_EB(gamma.I, gamma.CML1, asy.CML[["asyV.I"]])[["gamma.EB"]]
gamma.EB2 = get_gamma_EB(gamma.I, gamma.CML2, asy.CML[["asyV.I"]])[["gamma.EB"]]
```

```
#Get the asymptotic variance of the EB estimates
V.EB = get_var_EB(k=2, q=4, gamma.CML=c(gamma.CML1, gamma.CML2),
                  gamma.I = gamma.I, asy.CML=asy.CML, seed=2333, nsim=2000)

#Get the SC-Learner estimates and the corresponding variance-covariance matrix
pred.matrix = matrix(c(1,1,1,0,
                       1,1,0,0), 4, 2)
rownames(pred.matrix) = c('int','X1','X2','B')
colnames(pred.matrix) = c('E1','E2')

get_SCLearner(k=2,
              q=4,
              pred.matrix=pred.matrix,
              gamma.EB=cbind(gamma.EB1, gamma.EB2),
              V.EB)
```

`get_var_EB` *Using simulation to obtain the asymptotic variance-covariance matrix of gamma_EB, package corpcor and MASS are required*

Description

Using simulation to obtain the asymptotic variance-covariance matrix of gamma_EB, package corpcor and MASS are required

Usage

```
get_var_EB(k, q, gamma.CML, gamma.I, asy.CML, seed = 2333, nsim = 2000)
```

Arguments

`k`	number of external models
`q`	total number of covariates (X,B) including the intercept (i.e. q=ncol(X)+ncol(B)+1)
`gamma.CML`	stack all k CML estimates in order, i.e. c(gamma.CML1,...,gamma.CMLk)
`gamma.I`	direct regression estimates using the internal data only
`asy.CML`	a list of the estimated asymtotic variance-covariance matrix of c(gamma_CML, gamma_I) from the output of function asympVar_LinReg() or asympVar_LogReg()
`seed`	specify seed for simulation
`nsim`	number of simulation, default nsim=2,000

Value

a list with: Var(gamma_EB), Cov(gamma_EB, gamma_I) and Cov(gamma_EBi, gamma_EBj)

References

Gu, T., Taylor, J.M.G. and Mukherjee, B. (2020). An ensemble meta-prediction framework to integrate multiple regression models into a current study. Manuscript in preparation.

Examples

```
# Full model: Y|X1, X2, B
# Reduced model 1: Y|X1 of sample size m1
# Reduced model 2: Y|X2 of sample size m2
# (X1, X2, B) follows normal distribution with mean zero, variance one and correlation 0.3
# Y|X1, X2, B follows Bernoulli[expit(-1-0.5*X1-0.5*X2+0.5*B)], where expit(x)=exp(x)/[1+exp(x)]
set.seed(2333)
n = 1000
data.n = data.frame(matrix(ncol = 4, nrow = n))
colnames(data.n) = c('Y', 'X1', 'X2', 'B')
data.n[,c('X1', 'X2', 'B')] = MASS::mvrnorm(n, rep(0,3), diag(0.7,3)+0.3)
data.n$Y = rbinom(n, 1, expit(-1 - 0.5*data.n$X1 - 0.5*data.n$X2 + 0.5*data.n$B))

# Generate the beta estimates from the external reduced model:
# generate a data of size m from the full model first, then fit the reduced regression
# to obtain the beta estiamtes and the corresponsing estimated variance
m = m1 = m2 = 30000
data.m = data.frame(matrix(ncol = 4, nrow = m))
names(data.m) = c('Y', 'X1', 'X2', 'B')
data.m[,c('X1', 'X2', 'B')] = MASS::mvrnorm(m, rep(0,3), diag(0.7,3)+0.3)
data.m$Y = rbinom(m, 1, expit(-1 - 0.5*data.m$X1 - 0.5*data.m$X2 + 0.5*data.m$B))

#fit Y|X to obtain the external beta estimates, save the beta estiamtes and
# the corresponsing estimated variance
fit.E1 = glm(Y ~ X1, data = data.m, family = binomial(link='logit'))
fit.E2 = glm(Y ~ X2, data = data.m, family = binomial(link='logit'))
beta.E1 = coef(fit.E1)
beta.E2 = coef(fit.E2)
names(beta.E1) = c('int', 'X1')
names(beta.E2) = c('int', 'X2')
V.E1 = vcov(fit.E1)
V.E2 = vcov(fit.E2)

#Save all the external model information into lists for later use
betaHatExt_list = list(Ext1 = beta.E1, Ext2 = beta.E2)
CovExt_list = list(Ext1 = V.E1, Ext2 = V.E2)
rho = list(Ext1 = n/m1, Ext2 = n/m2)

#get full model estimate from direct regression using the internal data only
fit.gamma.I = glm(Y ~ X1 + X2 + B, data = data.n, family = binomial(link='logit'))
gamma.I = coef(fit.gamma.I)

#Get CML estimates using internal data and the beta estimates from the external
# model 1 and 2, respectively
gamma.CML1 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X1,
             BInt=cbind(data.n$X2, data.n$B), betaHatExt=beta.E1, gammaHatInt=gamma.I,
             n=nrow(data.n), tol=1e-8, maxIter=400,factor=1)[["gammaHat"]]
```

```
gamma.CML2 = fxnCC_LogReg(p=2, q=4, YInt=data.n$Y, XInt=data.n$X2,
             BInt=cbind(data.n$X1, data.n$B), betaHatExt=beta.E2, gammaHatInt=gamma.I,
             n=nrow(data.n), tol=1e-8, maxIter=400, factor=1)[["gammaHat"]]
#It's important to reorder gamma.CML2 so that it follows the order (X1, X2, X3, B)
# as gamma.I and gamma.CML1
gamma.CML2 = c(gamma.CML2[1], gamma.CML2[3], gamma.CML2[2], gamma.CML2[4])

#Get Variance-covariance matricx of c(gamma.I, gamma.CML1, gamma.CML2)
asy.CML = asympVar_LogReg(k=2, p=2,q=4, YInt=data.n$Y, XInt=data.n[,c('X1','X2')],
                   BInt=data.n$B,  gammaHatInt=gamma.I, betaHatExt_list=betaHatExt_list,
                         CovExt_list=CovExt_list, rho=rho, ExUncertainty=TRUE)

#Get the empirical Bayes (EB) estimates
gamma.EB1 = get_gamma_EB(gamma.I, gamma.CML1, asy.CML[["asyV.I"]])[["gamma.EB"]]
gamma.EB2 = get_gamma_EB(gamma.I, gamma.CML2, asy.CML[["asyV.I"]])[["gamma.EB"]]

#Get the asymptotic variance of the EB estimates
V.EB = get_var_EB(k=2,
                  q=4,
                  gamma.CML=c(gamma.CML1, gamma.CML2),
                  gamma.I = gamma.I,
                  asy.CML=asy.CML,
                  seed=2333,
                  nsim=2000)
asyV.EB1 = V.EB[['asyV.EB']][[1]]             #variance of gamma.EB1
asyV.EB2 = V.EB[['asyV.EB']][[2]]             #variance of gamma.EB2
asyCov.EB1.I = V.EB[['asyCov.EB.I']][[1]]     #covariance of gamma.EB1 and gamma.I
asyCov.EB2.I = V.EB[['asyCov.EB.I']][[2]]     #covariance of gamma.EB2 and gamma.I
asyCov.EB12 = V.EB[['asyCov.EB']][['12']]     #covariance of gamma.EB1 and gamma.EB2
```

Index