CSMT_markdown

Asmita Roy

2025-03-20

```
source("~/Downloads/CSMT/CSMT_main.R")
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
decision = function(n,p)
{
  if(sqrt(n)/0.5 < 5*p) {
    print("Dimension reduction of covariates via principal components is recommended.")
    print("Please add intercept = TRUE in CSMT_pval() when using principal components")
  }else{
    print("Dimension reduction is not necessary")
  }
}
```

- The mediator-predictor model is referred to as "model1" and the outcome-mediator model is referred to as "model2" hereafter. Both must be explicitly stated in the formula for the CSMT function. The variable names used in the model must match the variable names in the data matrix.
- Currently CSMT only supports univariate predictors, mediators and outcomes, but it supports a large number of additional covariates. To assist the user in deciding whether dimension reduction is necessary, use decision(n, p) as follows, where n is the sample size and p is the number of additional covariates. This is merely a rule-of-thumb guidance, and we leave the final choice to the statistician's discretion.
- The data to be passed into the CSMT function must be in a data.frame() format, with the columns organized in the following order - predictor, mediator, outcome, covariates. covariates must include an intercept column.
- Since the intercept is explicitly specified in the intercept, by default model1 and model2 should be specified with intercept.
- If, however, the user decides to add an intercept to the models, she must specify intercept = TRUE in the CSMTpval function.
- If the user reduces the dimension of the covariates using principal component analysis then intercept = TRUE must be specified in the CSMT function.

```
decision(n = 100, p = 4)
```

```
## [1] "Dimension reduction is not necessary"
```

```r
decision(n = 70, p = 3)
```

```
## [1] "Dimension reduction is not necessary"
```

```r
decision(n = 70, p = 5)
```

```
## [1] "Dimension reduction of covariates via principal components is recommended."
## [1] "Please add intercept = TRUE in CSMT_pval() when using principal components"
```

```r
alpha = 0.1
beta = 0.0
gamma = 0.1
n = 100
p = 3
coef_X_alpha = coef_X_beta = c(0.1,0.1,0.1)
other_cov = matrix(nrow = n, ncol = p)
other_cov[,1] = rep(1,n)

for(i in 2:p)
{
  other_cov[,i] = 2 + rnorm(n)
}

other_covariate_names <- paste0("cov", seq_len(p))  # Assuming first 3 columns are predictor, mediator,
colnames(other_cov) <- other_covariate_names
X = runif(n,3,4)
  M = X*alpha + rowSums(coef_X_alpha*other_cov) + rnorm(n)
  del = M*beta + X*gamma + rowSums(coef_X_beta*other_cov)
  Y = rpois(n, exp(del))
  my_data <- data.frame(
    predictor = X,
    mediator = M,
    outcome = Y,
    other_cov
  )

p_value <- CSMT_pval(
  model1 = mediator ~ predictor + cov1 + cov2 + cov3 - 1,
  model2 = outcome ~ mediator + predictor + cov1 cov2 + cov3 - 1,
  data = my_data,
  family1 = gaussian(),  # Gaussian model for mediator
  family2 = poisson(link = "log"),  # Poisson model for outcome
)
p_value
```

```
## [1] 0.2130008
```