Demonstrating functional mediation

John J. Dziak

2020-06-05

First we load the required packages.

```
library(tvem)
#> Loading required package: mgcv
#> Loading required package: nlme
#> This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
library(refund)
library(boot)
#>
#> Attaching package: 'boot'
#> The following object is masked from 'package:refund':
#>
#>     cd4
```

We then simulate data. The info1 object will contain not only the simulated dataset, but the true values of the simulated parameters, including the indirect effect.

```
set.seed(123);
info1 <- simulate_functional_mediation_example();
the_data <- info1$dataset;
```

Now call the functional mediation function. Only 19 bootstrap samples are used below for the illustration. Only 9 bootstraps allows possible p-values of .05, .10, . . . , 1.0. At least 199 bootstraps is recommended in practice in order to increase precision and power, allowing p-values of .005, .010, .015, ...

```
model1 <- funreg_mediation(data=the_data,
                           treatment=X,
                           mediator=M,
                           outcome=Y,
                           id=subject_id,
                           time=t,
                           nboot=19);
#> Ran original results.
#> Working on bootstrap results:
#> 1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.18.19.Done bootstrapping.
#> Warning in norm.inter(t, (1 + c(conf, -conf))/2): extreme order statistics used
#> as endpoints
#> Warning in norm.inter(t, alpha): extreme order statistics used as endpoints
```

We can print and plot the results.

```
print(model1);
#> =======================================================
#> Functional Regression Mediation Function Output
#> =======================================================
```

```
#> Indirect effect bootstrap estimate:
#> -0.1006469
#> Indirect effect bootstrap confidence interval:
#> ... by normal method:
#> -0.177 ,  -0.0242
#> ... by percentile method:
#> -0.1394 ,  -0.0122
#> Computation time:
#> Time difference of 34.27996 secs
#> =========================================================
#> TVEM Model for Predicting Mediator from Treatment:
#> Response variable:   mediator
#> Time interval:   0.01 to 1
#> Number of subjects:  250
#> Effects specified as time-varying:  (Intercept), treatment
#> You can use the plot_tvem function to view their plots.
#>
#> Back-end model fitted in mgcv::bam function:
#> Method fREML
#> Formula:
#> mediator ~ treatment + s(time, bs = "ps", by = NA, pc = 0, k = 7,
#>     fx = FALSE) + s(time, bs = "ps", by = treatment, pc = 0,
#>     m = c(2, 1), k = 7, fx = FALSE)
#> Pseudolikelihood AIC: 42181.28
#> Pseudolikelihood BIC: 42213.8
#>
#> =========================================================
#> Functional Mediation Model for Predicting Mediator from Treatment:
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> wide_outcome ~ s(x = wide_mediator.tmat, by = L.wide_mediator) +
#>     wide_treatment
#>
#> Estimated degrees of freedom:
#> 2  total = 4
#>
#> REML score: 332.6466
#> Scalar terms:
#>    (Intercept) wide_treatment
#>     0.15405471     0.08210633
#> =========================================================
#> Parametric model for Predicting Outcome from Treatment and Mediator:
#>
#> Call:
#> glm(formula = glm_formula)
#>
#> Deviance Residuals:
#>      Min        1Q    Median        3Q       Max
#> -2.50315  -0.59245  -0.04527   0.60490   2.24758
#>
```

```
#> Coefficients:
#>                  Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    0.2640562  0.0802042   3.292  0.00114 **
#> wide_treatment 0.0001341  0.1148120   0.001  0.99907
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> (Dispersion parameter for gaussian family taken to be 0.8233878)
#>
#>     Null deviance: 204.2  on 249  degrees of freedom
#> Residual deviance: 204.2  on 248  degrees of freedom
#> AIC: 664.88
#>
#> Number of Fisher Scoring iterations: 2
#>
#> =======================================================
plot(model1);
```