Dynamic structural equation model

James T. Thorson

```
library(tinyVAST)
set.seed(101)
```

`tinyVAST` includes features to fit a dynamic structural equation model

```
data(isle_royale, package="dsem")

# Convert to long-form
data = expand.grid( "time"=isle_royale[,1], "var"=colnames(isle_royale[,2:3]) )
data$logn = unlist(log(isle_royale[2:3]))

# Define cross-lagged SEM
sem = "
  # Link, lag, param_name
  wolves -> wolves, 1, arW
  moose -> wolves, 1, MtoW
  wolves -> moose, 1, WtoM
  moose -> moose, 1, arM
  wolves -> moose, 0, corr
"

# fit model
mytiny = fit( sem = sem,
                data = data,
                times = isle_royale[,1],
                variables = colnames(isle_royale[,2:3]),
                estimate_delta0 = FALSE,
                formula = logn ~ 0 + var,
                quiet = TRUE )
#> Warning in nlminb(start = obj$par, obj = obj$fn, gr = obj$gr, control = list(eval.max = 10000, : NA/
mytiny
#> $call
#> fit(data = data, formula = logn ~ 0 + var, sem = sem, estimate_delta0 = FALSE,
#>     times = isle_royale[, 1], variables = colnames(isle_royale[,
#>         2:3]), quiet = TRUE)
#>
#> $opt
#> $opt$par
#>        alpha        alpha       beta_z       beta_z       beta_z       beta_z       beta_z       bet
#>   3.32526213   6.44165367   0.89304273   0.01420910  -0.11865010   0.86169495  -0.01539666   0.10648
#>
#> $opt$objective
#> [1] 5.781919
#>
```

```
#> $opt$convergence
#> [1] 0
#>
#> $opt$iterations
#> [1] 94
#>
#> $opt$evaluations
#> function gradient
#>      114       95
#>
#> $opt$message
#> [1] "relative convergence (4)"
#>
#>
#> $sdrep
#> sdreport(.) result
#>               Estimate   Std. Error
#> alpha       3.32526213 2.483419e-01
#> alpha       6.44165367 2.116016e-01
#> beta_z      0.89304273 8.420604e-02
#> beta_z      0.01420910 1.279147e-01
#> beta_z     -0.11865010 6.477638e-02
#> beta_z      0.86169495 7.080254e-02
#> beta_z     -0.01539666 6.067738e-02
#> beta_z      0.10648808 9.881848e-03
#> beta_z      0.04810383 4.460377e-03
#> log_sigma -12.77516609 2.336730e+04
#> Maximum gradient component: 9.036267e-06
#>
#> $run_time
#> Time difference of 7.589102 secs
```

We can then compare this with package `dsem`

```
library(dsem)

# Keep in wide-form
dsem_data = ts( log(isle_royale[,2:3]), start=1959)
family = c("normal","normal")

# initial first without delta0 (to improve starting values)
mydsem = dsem( sem = sem,
               tsdata = dsem_data,
               estimate_delta0 = FALSE,
               quiet = TRUE,
               getsd = FALSE,
               family = family )
mydsem
#> $par
#>        beta_z        beta_z        beta_z        beta_z        beta_z        beta_z        beta_z
#>   0.895834720   0.007358847  -0.109332511   0.875012562  -0.017355229   0.378795847  -0.172873038 -1
#>
#> $objective
```

```
#> [1] 7.739638
#>
#> $iterations
#> [1] 79
#>
#> $evaluations
#> function gradient
#>       96       80
#>
#> $time_for_MLE
#> Time difference of 1.043584 secs
#>
#> $max_gradient
#> [1] 7.714655e-07
#>
#> $Convergence_check
#> [1] "There is no evidence that the model is not converged"
#>
#> $number_of_coefficients
#>  Total  Fixed Random
#>    133      9    124
#>
#> $AIC
#> [1] 33.47928
#>
#> $diagnostics
#>       Param starting_value Lower           MLE Upper final_gradient
#> 1    beta_z           0.01  -Inf   0.895834720   Inf   4.785205e-09
#> 2    beta_z           0.01  -Inf   0.007358847   Inf  -5.078683e-09
#> 3    beta_z           0.01  -Inf  -0.109332511   Inf  -2.031211e-08
#> 4    beta_z           0.01  -Inf   0.875012562   Inf  -5.821149e-08
#> 5    beta_z           0.01  -Inf  -0.017355229   Inf  -5.373382e-09
#> 6    beta_z           1.00  -Inf   0.378795847   Inf   2.119351e-09
#> 7    beta_z           1.00  -Inf  -0.172873038   Inf  -7.714655e-07
#> 8 lnsigma_j           0.00  -Inf -15.799262455   Inf   1.628788e-12
#> 9 lnsigma_j           0.00  -Inf -11.977331517   Inf   2.141499e-09
#>
#> $time_for_run
#> Time difference of 1.059659 secs
```