TERGM&RSiena

Kirk Li

Thursday, August 07, 2014

```r
library(tergm)
```

```
## Loading required package: statnet.common
## Loading required package: ergm
## Loading required package: network
## network: Classes for Relational Data
## Version 1.10.2 created on 2014-06-12.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                     Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Martina Morris, University of Washington
##                     Skye Bender-deMoll, University of Washington
##  For citation information, type citation("network").
##  Type help("network-package") to get started.
##
##
## ergm: version 3.1.2, created on 2014-01-30
## Copyright (c) 2014, Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Carter T. Butts, University of California -- Irvine
##                     Steven M. Goodreau, University of Washington
##                     Pavel N. Krivitsky, University of Wollongong
##                     Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("ergm").
##
## NOTE: If you use custom ERGM terms based on 'ergm.userterms'
## version prior to 3.1, you will need to perform a one-time update
## of the package boilerplate files (the files that you did not write
## or modify) from 'ergm.userterms' 3.1 or later. See
## help('eut-upgrade') for instructions.
##
## NOTE: Dynamic network modeling functionality (STERGMs) has been
## moved to a new package, 'tergm'.
##
## Loading required package: networkDynamic
##
## networkDynamic: version 0.6.3, created on 2014-02-27
## Copyright (c) 2014, Carter T. Butts, University of California -- Irvine
##                     Ayn Leslie-Cook, University of Washington
##                     Pavel N. Krivitsky, University of Wollongong
##                     Skye Bender-deMoll, University of Washington
##                     with contributions from
##                     Zack Almquist, University of California -- Irvine
##                     David R. Hunter, Penn State University
##                     Li Wang
```

```
##                           Kirk Li, University of Washington
##                           Steven M. Goodreau, University of Washington
##                           Martina Morris, University of Washington
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("networkDynamic").
##
##
## tergm: version 3.1.4, created on 2014-01-16
## Copyright (c) 2014, Pavel N. Krivitsky, University of Wollongong
##                           Mark S. Handcock, University of California -- Los Angeles
##                           with contributions from
##                           David R. Hunter, Penn State University
##                           Steven M. Goodreau, University of Washington
##                           Martina Morris, University of Washington
##                           Nicole Bohme Carnegie, New York University
##                           Carter T. Butts, University of California -- Irvine
##                           Ayn Leslie-Cook, University of Washington
##                           Skye Bender-deMoll
## Based on "statnet" project software (statnet.org).
## For license and citation information see statnet.org/attribution
## or type citation("tergm").
```

```r
library(RSiena)
data(samplk)
```

```r
samplk12 <- stergm(list(samplk1, samplk2),
                   formation=~ostar(1)+mutual+odegreepopularity+ttriple+transitiveties+balance+cycle(3)
                   dissolution=~ostar(1)+mutual+odegreepopularity+ttriple+transitiveties+balance+cycle(3
    estimate="CMLE")
```

```
## Warning: Time points not specified for a list. Modeling transition from
## the between successive networks jointly. This behavior may change in the
## future.
```

```
## Fitting formation...
## Iteration 1 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 0.05628
## Iteration 2 of at most 20:
## Convergence test P-value: 9.9e-203
## The log-likelihood improved by 0.0265
## Iteration 3 of at most 20:
## Convergence test P-value: 1.5e-47
## The log-likelihood improved by 0.006322
## Iteration 4 of at most 20:
## Convergence test P-value: 9.4e-11
## The log-likelihood improved by 0.001592
## Iteration 5 of at most 20:
## Convergence test P-value: 3.8e-05
## The log-likelihood improved by 0.0008905
## Iteration 6 of at most 20:
## Convergence test P-value: 5.8e-02
```

```
## The log-likelihood improved by 0.0003709
## Iteration 7 of at most 20:
## Convergence test P-value: 3.3e-01
## The log-likelihood improved by 0.0002053
## Iteration 8 of at most 20:
## Convergence test P-value: 2.9e-02
## The log-likelihood improved by 0.0004431
## Iteration 9 of at most 20:
## Convergence test P-value: 2.1e-01
## The log-likelihood improved by 0.000314
## Iteration 10 of at most 20:
## Convergence test P-value: 2.9e-01
## The log-likelihood improved by 0.0001493
## Iteration 11 of at most 20:
## Convergence test P-value: 7.5e-01
## Convergence detected. Stopping.
## The log-likelihood improved by 0.000113
##
## This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the mcmc.
## Fitting dissolution...
## Iteration 1 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 4.672
## Iteration 2 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 14.18
## Iteration 3 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood did not improve.
## Iteration 4 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 16.64
## Iteration 5 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 19.79
## Iteration 6 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 19.62
## Iteration 7 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 1.135
## Iteration 8 of at most 20:
## Convergence test P-value: 0e+00
## The log-likelihood improved by 0.2635
## Iteration 9 of at most 20:
## Convergence test P-value: 2.4e-266
## The log-likelihood improved by 0.04987
## Iteration 10 of at most 20:
## Convergence test P-value: 3.3e-71
## The log-likelihood improved by 0.01229
## Iteration 11 of at most 20:
## Convergence test P-value: 2.4e-24
## The log-likelihood improved by 0.0044
## Iteration 12 of at most 20:
```

```
## Convergence test P-value: 4.6e-02
## The log-likelihood improved by 0.0007071
## Iteration 13 of at most 20:
## Convergence test P-value: 2.7e-01
## The log-likelihood improved by 0.0003027
## Iteration 14 of at most 20:
## Convergence test P-value: 8.5e-02
## The log-likelihood improved by 0.0005677
## Iteration 15 of at most 20:
## Convergence test P-value: 3.3e-03
## The log-likelihood improved by 0.0006838
## Iteration 16 of at most 20:
## Convergence test P-value: 4.6e-01
## The log-likelihood improved by 0.000143
## Iteration 17 of at most 20:
## Convergence test P-value: 7.3e-02
## The log-likelihood improved by 0.0003401
## Iteration 18 of at most 20:
## Convergence test P-value: 4e-01
## The log-likelihood improved by 0.0002526
## Iteration 19 of at most 20:
## Convergence test P-value: 5.4e-01
## Convergence detected. Stopping.
## The log-likelihood improved by 0.0001573
##
## This model was fit using MCMC.  To examine model diagnostics and check for degeneracy, use the mcmc.
```

```r
samplk12.s <- sienaDependent(
    array( c(as.matrix(samplk1), as.matrix(samplk2)),dim=c(18,18,2)))

mydata <- sienaDataCreate(samplk12.s)
mydata
```

```
## Dependent variables:  samplk12.s
## Number of observations: 2
##
## Nodeset                  Actors
## Number of nodes              18
##
## Dependent variable samplk12.s
## Type             oneMode
## Observations     2
## Nodeset          Actors
## Densities        0.18 0.19
```

```r
myeff <- getEffects( mydata )
myeff <- includeEffects( myeff, outPop, transTies, balance,transTrip, cycle3 )
```

```
##                     name shortName type interaction1 interaction2
## samplk12.s.eval.3  samplk12.s transTrip eval
## samplk12.s.eval.6  samplk12.s    cycle3 eval
## samplk12.s.eval.7  samplk12.s transTies eval
## samplk12.s.eval.9  samplk12.s   balance eval
```

```
## samplk12.s.eval.20 samplk12.s    outPop eval
##                  include
## samplk12.s.eval.3    TRUE
## samplk12.s.eval.6    TRUE
## samplk12.s.eval.7    TRUE
## samplk12.s.eval.9    TRUE
## samplk12.s.eval.20   TRUE
```

```
myeff
```

```
##    effectName                    include fix    test   initialValue parm
## 1 basic rate parameter samplk12.s TRUE   FALSE FALSE      4.9368   0
## 2 outdegree (density)            TRUE   FALSE FALSE     -0.7114   0
## 3 reciprocity                    TRUE   FALSE FALSE      0.0000   0
## 4 transitive triplets            TRUE   FALSE FALSE      0.0000   0
## 5 3-cycles                       TRUE   FALSE FALSE      0.0000   0
## 6 transitive ties                TRUE   FALSE FALSE      0.0000   0
## 7 balance                        TRUE   FALSE FALSE      0.0000   0
## 8 outdegree - popularity         TRUE   FALSE FALSE      0.0000   0
```

```
myalgorithm <- sienaAlgorithmCreate(useStdInits = FALSE, projname = 'samp')
ans <- siena07( myalgorithm, data = mydata, effects = myeff)
```

```
(summary(samplk12)$formation$coefs)
```

```
##                   Estimate Std. Error MCMC % p-value
## ostar1             7.55214    5.2285      0 0.14991
## mutual             3.66849    1.7713      0 0.03941
## odegreepopularity -3.77399    1.7709      0 0.03408
## ttriple            0.35426    0.1420      0 0.01324
## transitiveties    -0.03531    0.3189      0 0.91190
## balance           -0.10071    0.1433      0 0.48289
## cycle3            -0.28119    0.4597      0 0.54135
```

```
(summary(samplk12)$dissolution$coefs)
```

```
##                   Estimate Std. Error MCMC % p-value
## ostar1             0.7706    1.8881       0 0.68500
## mutual            -2.9474    3.8080       0 0.44272
## odegreepopularity -0.1899    0.9314       0 0.83933
## ttriple            2.4315    1.0511       0 0.02504
## transitiveties    -2.6849    1.5386       0 0.08737
## balance            0.3367    0.2795       0 0.23418
## cycle3            -1.3232    0.8795       0 0.13901
```

```
summary(samplk12)$formation$coefs[,1,drop=FALSE]+summary(samplk12)$dissolution$coefs[,1,drop=FALSE]
```

```
##                Estimate
## ostar1          8.3227
## mutual          0.7211
```

```
## odegreepopularity   -3.9639
## ttriple              2.7857
## transitiveties       -2.7202
## balance               0.2360
## cycle3               -1.6044
```

ans

```
## Estimates, standard errors and convergence t-ratios
##
##                                 Estimate   Standard   Convergence
##                                             Error      t-ratio
##
## Rate parameters:
##   0      Rate parameter          4.0316  ( 0.8026   )
##   1. eval outdegree (density)    0.2819  ( 2.2934   )    -0.0621
##   2. eval reciprocity            0.8617  ( 0.5617   )     0.0608
##   3. eval transitive triplets   -0.5457  ( 0.7037   )     0.1240
##   4. eval 3-cycles              -0.1739  ( 0.4048   )     0.0404
##   5. eval transitive ties       -0.5948  ( 0.7389   )     0.0801
##   6. eval balance                0.7717  ( 0.3721   )    -0.1584
##   7. eval outdegree - popularity 0.2454  ( 0.8467   )    -0.0560
##
## Total of 2550 iteration steps.
```

]