netnew: a replication package for Mele (2015)

Angelo Mele

Sunday, March 01, 2015

Contents

Introduction

This vignette explains how to use the replication package that is distributed with *A structural model of segregation in social networks*. Most of the structure is similar to the `ergm` package in `statnet`. The package contains a set of R wrapper functions that call faster Fortran 90 routines for the simulation and estimation of the models.

Installation

To install the package from the Github repository, you need to have `devtools` already installed.
If you don't have the `devtools`, install it using the following command

```
install.packages("devtools")
```

Then open `devtools` and use the `install_github` command to download and install `netnew`:

```
library(devtools)
install_github("meleangelo/netnew")
```

This should install the package. This works well in Windows, using RStudio.
To use the package just open it as an R library

```
library(netnew)
```

Now you should be able to run the codes.

Simulating a network

We want to simulate a model with $n = 30$ players and only direct links and indirect links utility, starting the simulations at the empty network. We want to use a local chain and get a sapmle of $R = 10000$ networks obtained sampling every 15 iterations of the sampler.

```
library(netnew)
set.seed(1977)
n <- 30 # size of network
a <- c(-3,0,3/n) # parameters for simulations

g <- matrix(data = rbinom(n*n,1,0), nrow = n, ncol = n)
diag(g)<-0
net <- simulatemodelbham(model = "3params", g, a, skip = 15, samplesize = 10000)
```

The `simulatemodelbham` generates a `data.frame` containing a sample of network statistics for the model: number of links, number of mutual links, and number of indirect links. To see the first few simulated statistics we use

```
head(net)
```

```
##   links mutual indirect
## 1     0      0        0
## 2     0      0        0
## 3     0      0        0
## 4     1      0        0
## 5     2      0        0
## 6     2      0        0
```

and the last few statistics are

```
tail(net)
```

```
##       links mutual indirect
## 9995     38      0       48
## 9996     41      0       55
## 9997     39      0       50
## 9998     39      0       50
## 9999     40      0       53
## 10000    40      0       52
```

How to simulate models that are not in the default.

If the model is not in the default, you can use the general method. The method follows closely the way the ERGM models are specified in the package `ergm` in `statnet`.
The previous model can be estimated using the following method

```
library(netnew)
set.seed(1977)
n <- 30 # size of network
a <- c(-3,0,3/n) # parameters for simulations
g <- matrix(data = rbinom(n*n,1,0), nrow = n, ncol = n)
diag(g)<-0
```

This is the same as before. In addition we need a matrix of covariates

```
x  <- matrix(data = 0, nrow = dim(g)[1], ncol = 1)
```

Now we need to provide several parameters: `dt` and `p`.

- `dt` is a matrix that contains indicators for which covariate and what type of sufficient statistics we want to use. The first column contains the indicator of which columns in the covariate matrix `x` we want to use for the sufficient statistic. The second column contains indicators of which particular change statistics we use.
- `p` is a vector with 3 entries and tells which column of `dt` refers to the last direct utility (first entry), mutual utility (second entry) and indirect utility (third entry). For example, if `p` is `(1,2,3)` and `dt` is

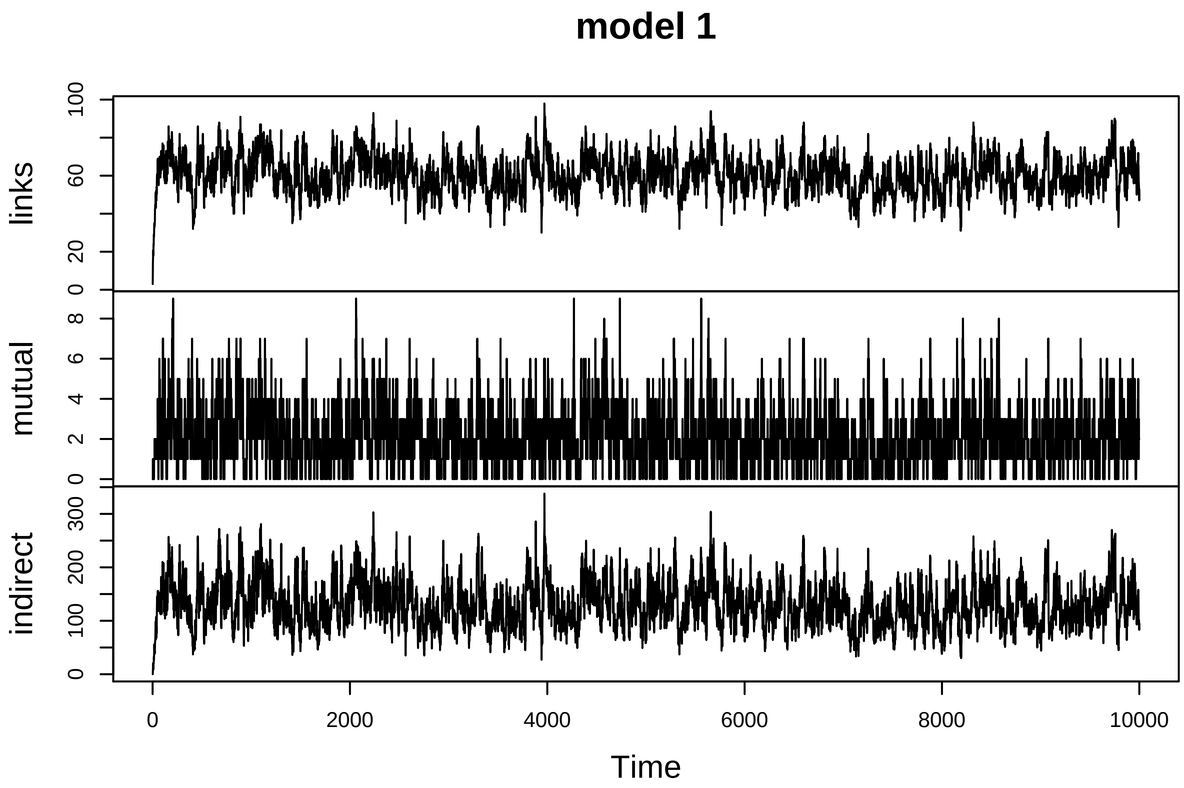
```
1 2
1 1
1 1
```

Looking at `p` we know that the first row of `dt` is relative to the direct utility, while the second is mutual utility and the third row is the indirect utility. In the first row of the matrix `dt` we are telling that we want to use column 1 of matrix `x`, and compute sufficient statistics 2: this corresponds to the difference $ x_i-x_j $; in the second row of `dt` is relative to the mutual utility, we want to use column 1 of `x` and compute sufficient statistics 1: this corresponds to the change in mutual links; the third row of `dt` is the indirect utility, and we are using column 1 of `x` and change statistics 1, which corresponds to counting the change in indirect links. For our example we will use

```
p  <- c(1,2,3)
dt <- matrix(data = c(1,1,1,1,1,1), nrow = 3, ncol = 2)
```

We can now run the simulation. We will get a sample of 10000 (`samplesize = 10000`) and we will sample every 100 iterations (`skip = 100`)

```
tsamp <- samplemanybham(g = g, x = x, a = a, ppp = p, dt = dt,
                        skip = 100 , samplesize = 10000)
colnames(tsamp) <- c("links","mutual","indirect")
plot.ts(tsamp, main = "model 1")
```



The model that we simulate has potential

$$Q(g) = -3\sum_{ij} g_{ij} + .1\sum_{ij}\sum_{ijk} g_{ij}g_{jk}$$

Let's simulate another model. We will now create a matrix of continuous covariates, with 2 covariates/columns. Each x is an iid draw from a normal $N(0,1)$.

```
set.seed(1977)
n <- 30 # size of network
a <- c(-3,.2,.5,3/n) # parameters for simulations
g <- matrix(data = rbinom(n*n,1,0), nrow = n, ncol = n)
diag(g)<-0

x  <- matrix(data = rnorm(n = 60, mean = 0, sd = 1),
             nrow = dim(g)[1], ncol = 2)
```

Now we set the parameters that describe the model

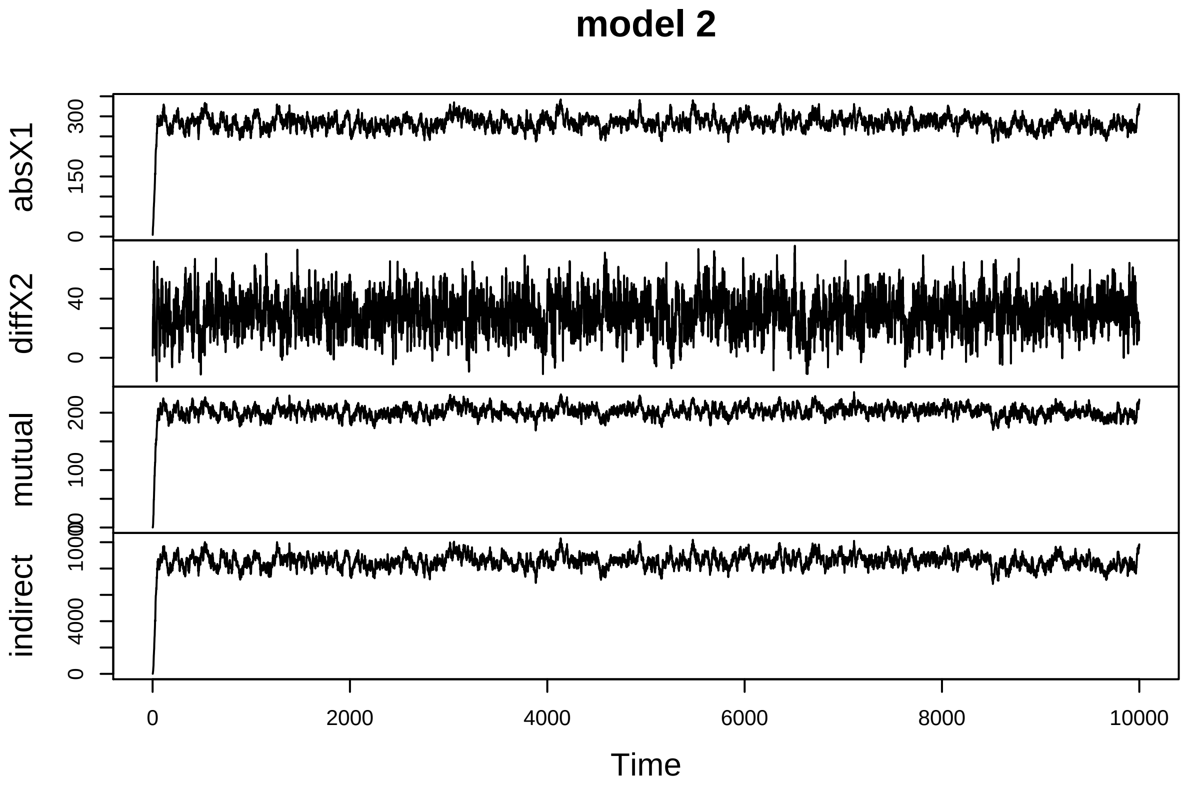
```
p <- c(2,3,4)
dt <- matrix(0, nrow = 4, ncol = 2)
dt[,1] <- c(1,2,1,1)
dt[,2] <- c(3,2,1,1)
dt
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    2
## [3,]    1    1
## [4,]    1    1
```

In this example, the direct utility contains 2 sufficient statistics, the first is the absolute difference of the first covariate $|x_{1i} - x_{1j}|$; the second is the difference of the second covariate $(x_{2i} - x_{2j})$.
The mutual utility is in the third entry of `dt` and corresponds to the change in mutual links; the indirect utility is the forth entry and contains only the change in indirect links.

We can now run the simulation. We will get a sample of 10000 (`samplesize = 10000`) and we will sample every 100 iterations (`skip = 100`)

```r
tsamp <- samplemanybham(g = g, x = x, a = a, ppp = p, dt = dt,
                        skip = 100 , samplesize = 10000)
colnames(tsamp) <- c("absX1", "diffX2","mutual","indirect")
plot.ts(tsamp, main = "model 2")
```



The model we simulated has potential

$$Q(g) = -3\sum_{ij} g_{ij}|x_{1i} - x_{1j}| + 0.2\sum_{ij} g_{ij}(x_{2i} - x_{2j}) + 0.1\sum_{ij} g_{ij}g_{ji} + 0.1\sum_{ijk} g_{ij}g_{jk}$$