Package 'NSUM'

March 3, 2015

Type Package

Title Network Scale Up Method

Version 1.0

Date 2014-12-17

Imports MASS, MCMCpack

Author Rachael Maltiel and Aaron J. Baraff

Maintainer Aaron J. Baraff <ajbaraff@uw.edu>

Description A Bayesian framework for population group size estimation using the Network Scale Up Method (NSUM). Size estimates are based on a random degree model and include options to adjust for barrier and transmission effects.

License GPL-2 | GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2015-03-03 21:37:49

R topics documented:

NSUM-package *NSUM: Network Scale Up Method*

Description

A Bayesian framework for subpopulation size estimation using the Network Scale Up Method (NSUM). Size estimates are based on a random degree model and include options to adjust for barrier and transmission effects.

Details

Package:	NSUM
Type:	Package
Version:	1.0
Date:	2014-12-17
License:	GPL-2 | GPL-3

The main estimation function is `nsum.mcmc`. It produces a Markov chain Monte Carlo (MCMC) sample from the posterior distributions of the subpopulation size parameters from a random degree model based upon the Network Scale Up Method (NSUM). Options allow for the inclusion of barrier and transmission effects, both separately and combined, resulting in four models altogether. Also included are functions to simulate data from any of these four models (`nsum.simulate`) and to estimate reasonable starting values for the MCMC sampler (`killworth.start`). Two data sets have been provided for testing purposes (`McCarty` and `Curitiba`).

Author(s)

Rachael Maltiel and Aaron J. Baraff

Maintainer: Aaron J. Baraff <ajbaraff at uw.edu>

References

Killworth, P., Johnsen, E., McCarty, C., Shelley, G., and Bernard, H. (1998a), "A Social Network Approach to Estimating Seroprevalence in the United States," Social Networks, 20, 23-50.

Killworth, P., McCarty, C., Bernard, H., Shelley, G., and Johnsen, E. (1998b), "Estimation of Seroprevalence, Rape, and Homelessness in the United States using a Social Network Approach," Evaluation Review, 22, 289-308.

Maltiel, R., Raftery, A. E., McCormick, T. H., and Baraff, A. J., "Estimating Population Size Using the Network Scale Up Method." CSSS Working Paper 129. Retrieved from https://www.csss.washington.edu/Papers/2013/wp129.pdf

McCarty, C., Killworth, P. D., Bernard, H. R., Johnsen, E. C., and Shelley, G. A. (2001), "Comparing Two Methods for Estimating Network Size," Human Organization, 60, 28-39.

Salganik, M., Fazito, D., Bertoni, N., Abdo, A., Mello, M., and Bastos, F. (2011a), "Assessing Network Scale-up Estimates for Groups Most at Risk of HIV/AIDS: Evidence From a Multiple-Method Study of Heavy Drug Users in Curitiba, Brazil," American Journal of Epidemiology, 174, 1190-1196.

See Also

killworth.start, nsum.mcmc, nsum.simulate

Examples

```
## load data
data(McCarty)

## simulate from model with barrier effects
sim.bar <- with(McCarty, nsum.simulate(100, known, unknown, N, model="barrier",
                                       mu, sigma, rho))

## estimate unknown population size
dat.bar <- sim.bar$y
mcmc <- with(McCarty, nsum.mcmc(dat.bar, known, N, model="barrier", iterations=100,
                                burnin=50))

## view posterior distribution
hist(mcmc$NK.values[1,])
```

Curitiba *Curitiba Dataset*

Description

This dataset contains the subpopulation sizes and parameters used for simulations involving the Curitiba data.

Usage

```
data("Curitiba")
```

Format

A list with the following 7 variables.

known a vector of positive numbers, the sizes of known subpopulations.

unknown a vector of positive numbers, the sizes of unknown subpopulations.

N a positive number, the (known) total population size.

mu a real number, the location parameter for the log-normal distribution of network degrees, with default 5.

sigma a positive number, the scale parameter for the log-normal distribution of network degrees, with default 1.

rho a vector of numbers between 0 and 1 with length equal to the total number of subpopulations, known and unknown, the dispersion parameters for the barrier effects, with defaults 0.1.

tauK a vector of numbers between 0 and 1 with length equal to the total number of unknown subpopulations, the multipliers for the transmission biases, with defaults 1.

Details

The Curitiba dataset consists of 500 adult residents of Curitiba, Brazil and was collected through a household-based random sample in 2010.

Source

Salganik, M., Fazito, D., Bertoni, N., Abdo, A., Mello, M., and Bastos, F. (2011a), "Assessing Network Scale-up Estimates for Groups Most at Risk of HIV/AIDS: Evidence From a Multiple-Method Study of Heavy Drug Users in Curitiba, Brazil," American Journal of Epidemiology, 174, 1190-1196.

Examples

```
## load data
data(Curitiba)

## simulate from model with transmission bias
sim.trans <- with(Curitiba, nsum.simulate(100, known, unknown, N, model="transmission",
                                          mu, sigma, tauK))
```

`killworth` *Calculate Killworth Estimates*

Description

This function calculates the Killworth estimates for unknown subpopulation sizes based on NSUM data.

Usage

```
killworth(dat, known, N)
```

Arguments

`dat`	a matrix of non-negagtive integers, the `(i,k)`-th entry represents the number of people that the `i`-th individual knows from the `k`-th subpopulation.
`known`	a vector of positive numbers, the sizes of known subpopulations. All additional columns of `dat` are treated as unknown.
`N`	a positive number, the (known) total population size.

Details

The function `killworth` allows for the estimation of subpopulation sizes from Killworth's network scale-up model. These estimates can be used to compare with the MCMC results in this package. For reasonable starting values for the MCMC function `nsum.mcmc`, see the function `killworth.start`.

Value

A vector of positive numbers with length equal to the number of unknown subpopulations, the Killworth estimates of the subpopulation sizes.

Author(s)

Rachael Maltiel and Aaron J. Baraff

Maintainer: Aaron J. Baraff <ajbaraff at uw.edu>

References

Killworth, P., Johnsen, E., McCarty, C., Shelley, G., and Bernard, H. (1998a), "A Social Network Approach to Estimating Seroprevalence in the United States," Social Networks, 20, 23-50.

Killworth, P., McCarty, C., Bernard, H., Shelley, G., and Johnsen, E. (1998b), "Estimation of Seroprevalence, Rape, and Homelessness in the United States using a Social Network Approach," Evaluation Review, 22, 289-308.

See Also

`killworth.start`

Examples

```
## load data
data(McCarty)

## simulate from model with barrier effects
sim.bar <- with(McCarty, nsum.simulate(100, known, unknown, N, model="barrier",
                                       mu, sigma, rho))

## estimate unknown population sizes
dat.bar <- sim.bar$y
NK.killworth <- with(McCarty, killworth(dat.bar, known, N))
```

`killworth.start` *Killworth Starting Values for MCMC*

Description

This function uses the Killworth estimates to calculate reasonable starting values for the MCMC estimation.

Usage

```
killworth.start(dat, known, N)
```

Arguments

`dat`	a matrix of non-negagtive integers, the `(i,k)`-th entry represents the number of people that the `i`-th individual knows from the `k`-th subpopulation.
`known`	a vector of positive numbers, the sizes of known subpopulations. All additional columns of `dat` are treated as unknown.
`N`	a positive number, the (known) total population size.

Details

The function `killworth.start` allows for the estimation reasonable starting values for many of the parameters in the MCMC function `nsum.mcmc` based on Killworth's network scale-up model. These are the default starting values where applicable. For simple subpopulation size estimation using Killworth's model, see the function `killworth`.

Value

A list with four components:

`NK.start`	a vector of positive numbers with length equal to the total number of unknown subpopulations, the starting values for the sizes of the unknown subpopulations\.
`d.start`	a vector of positive numbers with length equal to the number of individuals, the starting values for the network degrees.
`mu.start`	a real number, the starting value for the location parameter for the log-normal distribution of network degrees.
`sigma.start`	a positive number, the starting value for the scale parameter for the log-normal distribution of network degrees.

Author(s)

Rachael Maltiel and Aaron J. Baraff

Maintainer: Aaron J. Baraff <ajbaraff at uw.edu>

References

Killworth, P., Johnsen, E., McCarty, C., Shelley, G., and Bernard, H. (1998a), "A Social Network Approach to Estimating Seroprevalence in the United States," Social Networks, 20, 23-50.

Killworth, P., McCarty, C., Bernard, H., Shelley, G., and Johnsen, E. (1998b), "Estimation of Seroprevalence, Rape, and Homelessness in the United States using a Social Network Approach," Evaluation Review, 22, 289-308.

Maltiel, R., Raftery, A. E., McCormick, T. H., and Baraff, A. J., "Estimating Population Size Using the Network Scale Up Method." CSSS Working Paper 129. Retrieved from https://www.csss.washington.edu/Papers/2013/wp129.pdf

See Also

killworth.start, nsum.mcmc

Examples

```
## load data
data(McCarty)

## simulate from model with barrier effects
sim.bar <- with(McCarty, nsum.simulate(100, known, unknown, N, model="barrier",
                                       mu, sigma, rho))

## estimate Killworth starting values
dat.bar <- sim.bar$y
start <- with(McCarty, killworth.start(dat.bar, known, N))

## estimate unknown population size from MCMC
mcmc <- with(McCarty, nsum.mcmc(dat.bar, known, N, model="barrier", iterations=100,
                                burnin=50, NK.start=start$NK.start, d.start=start$d.start,
                                  mu.start=start$mu.start, sigma.start=start$sigma.start))
```

McCarty *McCarty Dataset*

Description

This dataset contains the subpopulation sizes and parameters used for simulations involving the McCarty data.

Usage

```
data("McCarty")
```

Format

A list with the following 7 variables.

known a vector of positive numbers, the sizes of known subpopulations.

unknown a vector of positive numbers, the sizes of unknown subpopulations.

N a positive number, the (known) total population size.

mu a real number, the location parameter for the log-normal distribution of network degrees, with default 5.

sigma a positive number, the scale parameter for the log-normal distribution of network degrees, with default 1.

rho a vector of numbers between 0 and 1 with length equal to the total number of subpopulations, known and unknown, the dispersion parameters for the barrier effects, with defaults 0.1.

tauK a vector of numbers between 0 and 1 with length equal to the total number of unknown subpopulations, the multipliers for the transmission biases, with defaults 1.

Details

The McCarty data set was obtained through random digit dialing within the United States. It contains responses from 1,375 adults from two surveys: survey 1 with 801 responses conducted in January 1998 and survey 2 with 574 responses conducted in January 1999.

Source

Killworth, P., Johnsen, E., McCarty, C., Shelley, G., and Bernard, H. (1998a), "A Social Network Approach to Estimating Seroprevalence in the United States," Social Networks, 20, 23-50.

Killworth, P., McCarty, C., Bernard, H., Shelley, G., and Johnsen, E. (1998b), "Estimation of Seroprevalence, Rape, and Homelessness in the United States using a Social Network Approach," Evaluation Review, 22, 289-308.

Examples

```
## load data
data(McCarty)

## simulate from model with barrier effects
sim.bar <- with(McCarty, nsum.simulate(100, known, unknown, N, model="barrier",
                                       mu, sigma, rho))
```

`nsum.mcmc` *Run MCMC for NSUM Parameters*

Description

This function produces an MCMC sample from the posterior distributions of the subpopulation size parameters from an NSUM model.

Usage

```
nsum.mcmc(dat, known, N, indices.k = (length(known)+1):(dim(dat)[2]),
          iterations = 1000, burnin = 100, size = iterations,
          model = "degree", ...)
```

Arguments

dat	a matrix of non-negagtive integers, the (i,k)-th entry represents the number of people that the i-th individual knows from the k-th subpopulation with the columns representing known subpopulations coming before the columns representing unknown subpopulations.
known	a vector of positive numbers, the sizes of known subpopulations.
N	a positive number, the (known) total population size.
indices.k	a vector of positive integers, the indices of the columns of dat representing the unknown subpopulations of interest, with defaults of all unknown subpopulations in dat.
iterations	a positive integer, the total number of MCMC iterations after burn-in, with default 1000.
burnin	a non-negative integer, the number of burn-in MCMC iterations, with default 100.
size	a positive integer, the number of MCMC iterations kept after thinning, with default equal to iterations.
model	a character string, the model to be simulated from. This must be one of "degree", "barrier", "transmission", or "combined", with default "degree".
...	additional arguments to be passed to methods, such as starting values, prior parameters, and tuning parameters. Many methods will accept the following arguments:

NK.start a vector of positive numbers with length equal to the total number of unknown subpopulations, the starting values for the sizes of the unknown subpopulations, with defaults based on the Killworth estimates.

d.start a vector of positive numbers with length equal to the number of individuals, the starting values for the network degrees, with defaults based on the Killworth estimates.

mu.start a real number, the starting value for the location parameter for the log-normal distribution of network degrees, with default based on the Killworth estimates.

sigma.start a positive number, the starting value for the scale parameter for the log-normal distribution of network degrees, with default based on the Killworth estimates.

rho.start a vector of numbers between 0 and 1 with length equal to the total number of subpopulations, known and unknown, the starting values for the dispersion parameters for the barrier effects, with defaults 0.1.

tauK.start a vector of numbers between 0 and 1 with length equal to the total number of unknown subpopulations, the starting values for the multipliers for the transmission biases, with defaults 0.5.

`q.start` a matrix of numbers between 0 and 1, the `(i,k)`-th entry is the starting value for the binomial probability of the number of people that the `i`-th individual knows from the `k`-th subpopulation, with defaults of simple proportions based on the known subpopulation sizes and the Killworth estimates for unknown population sizes.

`mu.prior` a vector of two real numbers, the parameters of the uniform prior for the location parameter of the log-normal distribution of network degrees, with default `c(3,8)`.

`sigma.prior` a vector of two positive numbers, the parameters of the uniform prior for the scale parameter of the log-normal distribution of network degrees, with default `c(1/4,2)`.

`rho.prior` a vector of two numbers between 0 and 1, the parameters of the uniform prior for the dispersion parameters for the barrier effects, with default `c(0,1)`.

`tauK.prior` a matrix of numbers between 0 and 1 with two columns and rows equal to the total number of unknown subpopulations, the parameters of the beta priors for the multipliers for the transmission biases, with defaults `c(1,1)`.

`NK.tuning` a vector of positive numbers with length equal to the total number of unknown subpopulations, the standard deviations of the normal MCMC transitions for the sizes of the unknown subpopulations, with defaults of 0.25 times the starting values.

`d.tuning` a vector of positive numbers with length equal to the number of individuals, the standard deviation of the normal MCMC transitions for the network degrees, with defaults of 0.25 times the starting values.

`rho.tuning` a vector of numbers between 0 and 1 with length equal to the total number of subpopulations, known and unknown, the standard deviations of the normal MCMC transitions for the dispersion parameters for the barrier effects, with defaults of 0.25 times the starting values.

`tauK.tuning` a vector of numbers between 0 and 1 with length equal to the total number of unknown subpopulations, the standard deviations of the normal MCMC transitions for the multipliers for the transmission biases, with defaults of 0.25 times the starting values.

`q.tuning` a matrix of numbers between 0 and 1, the `(i,k)`-th entry is the standard deviation of the normal MCMC transitions for the binomial probability of the number of people that the `i`-th individual knows from the `k`-th subpopulation, with defaults of 0.25 times the starting values.

Details

The function `nsum.mcmc` allows for the estimation of the various parameters from a random degree model based upon the Network Scale Up Method (NSUM) by producing Markov chain Monte Carlo (MCMC) samples from their posterior distributions. Options allow for the inclusion of barrier and transmission effects, both separately and combined, resulting in four models altogether. A large number of iterations may be required for accurate inference due to slow mixing, so the resulting chain can be thinned using the `size` argument. It should be noted that subpopulation size estimation in the presence of transmission bias can be greatly improved when the priors for the multipliers `tauK` are highly informative.

Value

A list with up to nine components:

`NK.values`	a matrix of positive numbers with a row for each unknown subpopulation, the thinned MCMC chains representing the posterior distributions of the sizes of the unknown subpopulations.
`d.values`	a matrix of positive numbers with a row for each individual, the thinned MCMC chains representing the posterior distributions of the network degrees.
`mu.values`	a vector of real numbers, the thinned MCMC chain representing the posterior distribution of the location parameter of the log-normal distribution of network degrees.
`sigma.values`	a vector of positive numbers, the thinned MCMC chain representing the posterior distribution of the scale parameter of the log-normal distribution of network degrees.
`rho.values`	a matrix of numbers between 0 and 1 with a row for each subpopulation, known and unknown, the thinned MCMC chains representing the posterior distributions of the dispersion parameters for the barrier effects.
`tauK.values`	a matrix of numbers between 0 and 1 with a row for each unknown subpopulation, the thinned MCMC chains representing the posterior distributions of the multipliers for the transmission biases.
`q.values`	a three-dimensional array of numbers between 0 and 1 with a row for each pairing of individual and subpopulation, the thinned MCMC chains representing the binomial probabilities of the number of people that the individual knows from the subpopulation.
`NK.values`	a matrix of positive numbers with a row for each unknown subpopulation, the thinned MCMC chains representing the posterior distributions of the sizes of the unknown subpopulations.
`iterations`	a positive integer, the total number of MCMC iterations after burn-in.
`burnin`	a non-negative integer, the number of burn-in MCMC iterations.

Author(s)

Rachael Maltiel and Aaron J. Baraff

Maintainer: Aaron J. Baraff <ajbaraff at uw.edu>

References

Maltiel, R., Raftery, A. E., McCormick, T. H., and Baraff, A. J., "Estimating Population Size Using the Network Scale Up Method." CSSS Working Paper 129. Retrieved from https://www.csss.washington.edu/Papers/2013/wp129.pdf

See Also

`killworth.start`

Examples

```
## load data
data(McCarty)

## simulate from model with barrier effects
sim.bar <- with(McCarty, nsum.simulate(100, known, unknown, N, model="barrier",
                                       mu, sigma, rho))

## estimate unknown population size
dat.bar <- sim.bar$y
mcmc <- with(McCarty, nsum.mcmc(dat.bar, known, N, model="barrier", iterations=100,
                                burnin=50))

## view posterior distribution of subpopulation sizes for the first subpopulation
hist(mcmc$NK.values[1,])

## view posterior distribution of barrier effect parameters for the first subpopulation
hist(mcmc$rho.values[1,])
```

`nsum.simulate` *Simulate NSUM Data*

Description

This function simulates data from one of the four NSUM models.

Usage

```
nsum.simulate(n, known, unknown, N, model = "degree", ...)
```

Arguments

`n`	a non-negative integer, the number respondents in the sample.
`known`	a vector of positive numbers, the sizes of known subpopulations.
`unknown`	a vector of positive numbers, the sizes of unknown subpopulations.
`N`	a positive number, the (known) total population size.
`model`	a character string, the model to be simulated from. This must be one of `"degree"`, `"barrier"`, `"transmission"`, or `"combined"`, with default `"degree"`.
`...`	additional arguments to be passed to methods, such as starting values, prior parameters, and tuning parameters. Many methods will accept the following arguments:
	`mu` a real number, the location parameter for the log-normal distribution of network degrees, with default 5.
	`sigma` a positive number, the scale parameter for the log-normal distribution of network degrees, with default 1.

`rho` a vector of numbers between 0 and 1 with length equal to the total number of subpopulations, known and unknown, the dispersion parameters for the barrier effects, with defaults 0.1.

`tauK` a vector of numbers between 0 and 1 with length equal to the total number of unknown subpopulations, the multipliers for the transmission biases, with defaults 1.

Details

The function `nsum.simulate` allows for the simulation of data from a random degree model based upon the Network Scale Up Method (NSUM). Options allow for the inclusion of barrier and transmission effects, both separately and combined, resulting in four models altogether. Each call to the function results in the simulation of a single realization of data.

Value

A list with two components:

`y` a matrix of non-negagtive integers, the `(i,k)`-th entry represents the number of people that the `i`-th individual knows from the `k`-th subpopulation with the columns representing known subpopulations coming before the columns representing unknown subpopulations.

`d` a vector of positive numbers, the network degrees of the individuals. Only the integer parts were used for simulation.

Author(s)

Rachael Maltiel and Aaron J. Baraff

Maintainer: Aaron J. Baraff <ajbaraff at uw.edu>

References

Maltiel, R., Raftery, A. E., McCormick, T. H., and Baraff, A. J., "Estimating Population Size Using the Network Scale Up Method." CSSS Working Paper 129. Retrieved from https://www.csss.washington.edu/Papers/2013/wp129.pdf

See Also

`nsum.mcmc`

Examples

```
## load data
data(McCarty)

## simulate from model with barrier effects
sim.bar <- with(McCarty, nsum.simulate(100, known, unknown, N, model="barrier",
                                       mu, sigma, rho))

## simulate from model with both barrier effects and transmission biases
```

```
sim.comb <- with(McCarty, nsum.simulate(100, known, unknown, N, model="combined",
                                        mu, sigma, rho, tauK))

## extract data for use in MCMC
dat.bar <- sim.bar$y

## view degree distribution
hist(sim.bar$d)
```

Index