Package ‘spatcounts’

February 15, 2012

Type Package

Title Spatial count regression

Version 1.1

Date 2009-06-26

Author Holger Schabenberger

Maintainer Vinzenz Erhardt <erhardt@ma.tum.de>

Description Fit spatial CAR count regression models using MCMC

License GPL (>= 3)

LazyLoad yes

Repository CRAN

Date/Publication 2009-06-29 08:00:32

R topics documented:

spatcounts-package 2
Clarketest 2
DIC 3
est.sc 5
LogLike 7
sim.fm.X 8
sim.gmat 9
sim.nmat 9
sim.region 10
sim.Yin 10
Vuongtest 11

Index **13**

`spatcounts-package` *Spatial count regression*

Description

Fit spatial CAR count regression models using MCMC.

Details

Package:	spatcounts
Type:	Package
Version:	1.0
Date:	2009-06-26
License:	GPL (>= 3)
LazyLoad:	yes

Spatial data can be fit by MCMC using function 'est.sc'. Functions 'Vuongtest' and 'Clarketest' allow for nonnested model comparison, 'DIC' calculates the Deviance Information Criterion, 'LogLike' calculates the loglikelihoods.

Author(s)

Holger Schabenberger

Maintainer: Vinzenz Erhardt <erhardt@ma.tum.de>

References

Masterthesis: Schabenberger, Holger (2009). Spatial count regression models with applications to health insurance data. ("http://www-m4.ma.tum.de/Diplarb/").

`Clarketest` *Clarke's test for non-nested model comparison*

Description

'Clarketest' suggests the better of two (not necessarily nested) models according to Clarke's statistic for the parameters in each of the iterations.

Usage

```
Clarketest(LogLike1, LogLike2, alpha = 0.05, p = NULL, q = NULL,
correction = TRUE)
```

Arguments

LogLike1, LogLike2
: the output of two model fits obtained by using 'LogLike'.

alpha
: significance level, defaults to 0.05.

p, q
: the number of estimated coefficients in models LogLike1 and Loglike2, respectively.

correction
: boolean, if TRUE (default), the Schwarz correction will be used on the differences of log-likelihoods.

References

Clarke, Kevin A. (2003). Nonparametric Model Discrimination in International Relations. Journal of Conflict Resolution 47(1), 72-93.

Schwarz, G. (1978). Estimating the Dimension of a Model. Annals of Statistics 6, 461-464.

See Also

Vuongtest

Examples

```
data(sim.Yin)
data(sim.fm.X)
data(sim.region)
data(sim.gmat)
data(sim.nmat)

poi <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region,
model="Poi", sim.gmat, sim.nmat, 3)
nb <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region,
model="NB", sim.gmat, sim.nmat, 3)

DIC.poi <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)
DIC.nb <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, nb)

ll.poi <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)
ll.nb <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, nb)

Clarke.poi.nb <- Clarketest(ll.poi, ll.nb, alpha = 0.05, p = DIC.poi$p.D,
q = DIC.nb$p.D, correction = TRUE)
```

DIC *Deviance information criterion*

Description

Deviance information criterion without standardizing constant.

Usage

```
DIC(Yin, fm.X, region, regmodel, burnin = 1)
```

Arguments

`Yin`	response vector of length n.
`fm.X`	formula for mean design.
`region`	region of each observation.
`regmodel`	the output of model fits obtained by using 'est.sc'.
`burnin`	number of steps dropped from the chain to allow for a burn-in phase. Defaults to 1.

Value

`DIC`	deviance information criterion.
`mean.deviance`	posterior mean of deviance.
`p.D`	effective number of parameters.

References

Gelman, Andrew, John B. Carlin, Hall S. Stern, and Donald B. Rubin (2003). Bayesian Data Analysis, Second Edition. Chapman \& Hall/CRC.

Spiegelhalter, David J.; Nicola G. Best, Bradley P. Carlin, and Angelika van der Linde (2002). Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society, Series B. 64 (4), 583-639.

Examples

```
data(sim.Yin)
data(sim.fm.X)
data(sim.region)
data(sim.gmat)
data(sim.nmat)

poi <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region,
model="Poi", sim.gmat, sim.nmat, 3)
DIC.poi <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)
```

`est.sc` *Fitting spatial count regression models*

Description

MCMC algorithm for the Poisson, the GP, the NB, the ZIP and the ZIGP regression models with spatial random effects.

Usage

```
est.sc(Yin, fm.X, region, model = "Poi", gmat, nmat, totalit,
fm.ga = TRUE, t.i = NULL, phi0 = 1, omega0 = 0, r0 = 1,
beta0 = NULL, gamma0 = NULL, sigma0 = 1, psi0 = 1, Tau = 10,
alpha = 2)
```

Arguments

`Yin`	response vector of length n.
`fm.X`	formula for mean design.
`region`	region of each observation as vector of length n.
`model`	the regression model. Currently, the regression models "Poi", "NB", "GP", "ZIP" and "ZIGP" are supported. Defaults to 'Poi'.
`gmat`	spatial adjacency matrix, where entry (i,j) is 1 if region i is a neighbor of region j and 0 else. See data(sim.gmat) for an example.
`nmat`	matrix containing the number of neighbors of each region (last column) and the neighbors of each region (first (maximual number of neighbours) columns), filled up by zeros. See data(sim.nmat) for an example.
`totalit`	number of MCMC iterations, i.e. length of the Markov chain.
`fm.ga`	should the spatial random effects be included (defaults to TRUE)?
`t.i`	exposure vector.
`phi0`	starting value for the over-dispersion parameter for GP and ZIGP model.
`omega0`	starting value for the extra proportion for ZIP and ZIGP model.
`r0`	starting value for the scale paramter for NB model.
`beta0`	starting values for the regression parameters.
`gamma0`	starting values for the spatial paramters.
`sigma0`	starting value for the spatial hyperparamter of CAR prior.
`psi0`	starting value for the spatial hyperparamter of CAR prior.
`Tau`	modifiable normal prior for the regression parameters with variance Tau^2.
`alpha`	modifiable prior distribution of hyperparamter psi (suggested values: 2, 1.5, 1, 0.5, 0).

Value

acceptb	acceptance rate for the regression parameters beta.
acceptga	range of the acceptance rate for the spatial parameters gamma.
acceptphi	acceptance rate for the GP and ZIGP model specific dispersion parameter phi.
acceptomega	acceptance rate for the ZIP and ZIGP model specific extra proportion omega.
acceptr	acceptance rate for the NB model specific scale parameter r.
acceptpsi	acceptance rate for the spatial hyperparameter psi.
beta	are the parameter estimates for the regression parameters beta.
gamma	are the parameter estimates for the spatial parameters gamma.
invsigsq	are the parameter estimates for the inverse spatial hyperparameter sigma^2.
psi	are the parameter estimates for the spatial hyperparameter psi.
phi	are the parameter estimates for the GP and ZIGP model specific dispersion parameter phi.
omega	are the parameter estimates for the ZIP and ZIGP model specific extra proportion omega.
r	are the parameter estimates for the NB model specific scale parameter r.
Coefficients	are the number of parameter estimates.
t.i	exposure vector.

References

Gschloessl, Susanne (2006). Hierarchical Bayesian spatial regression models with applications to non-life insurance. Dissertation, Centre of Mathematical Sciences, Munich University of Technology, Chair in Mathematical Statistics, Munich University of Technology, Boltzmannstr. 3, D-85748 Garching near Munich.

Masterthesis: Schabenberger, Holger (2009). Spatial count regression models with applications to health insurance data. ("http://www-m4.ma.tum.de/Diplarb/").

Czado, C., Erhardt, V., Min, A., Wagner, S. (2007). Zero-inflated generalized Poisson models with regression effects on the mean, dispersion and zero-inflation level applied to patent outsourcing rates. Statistical Modelling 7 (2), 125-153.

See Also

R-package ZIGP for fitting GP, ZIP, ZIGP regression models using MLE.

Examples

```
data(sim.Yin)
data(sim.fm.X)
data(sim.region)
data(sim.gmat)
data(sim.nmat)
# true parameters for generating this data:
# beta.true = c(-1, 0.4, 1.5)
```

```
# gamma.true = vector of spatial effects according to the CAR model with mean 0, psi = 3 and sigma = 1
# range of gamma.true = c(-0.851, 0.8405)

# run all examples with higher number of iterations if you want to approximate the true parameters
# properly

poi <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region,
model="Poi", sim.gmat, sim.nmat, totalit=10)

# posterior means not considering a burn-in or thinning of iterations
apply(poi$beta,1,mean)
apply(poi$gamma,1,mean)


# Compare Poisson to different model classes
nb <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, model="NB", sim.gmat, sim.nmat, totalit=10)

gp <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, model="GP", sim.gmat, sim.nmat, totalit=10)

zip <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, model="ZIP", sim.gmat, sim.nmat, totalit=10)

zigp <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, model="ZIGP", sim.gmat, sim.nmat, totalit=10)

DIC.poi <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)
DIC.nb <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, nb)
DIC.gp <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, gp)
DIC.zip <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, zip)
DIC.zigp <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, zigp)

ll.poi <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)
ll.nb <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, nb)
ll.gp <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, gp)
ll.zip <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, zip)
ll.zigp <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, zigp)

Vuong.poi.nb <- Vuongtest(ll.poi, ll.nb, alpha = 0.05, p = DIC.poi$p.D, q = DIC.nb$p.D, correction = TRUE)
Vuong.poi.gp <- Vuongtest(ll.poi, ll.gp, alpha = 0.05, p = DIC.poi$p.D, q = DIC.gp$p.D, correction = TRUE)
Vuong.poi.zip <- Vuongtest(ll.poi, ll.zip, alpha = 0.05, p = DIC.poi$p.D, q = DIC.zip$p.D, correction = TRUE)
Vuong.poi.zigp <- Vuongtest(ll.poi, ll.zigp, alpha = 0.05, p = DIC.poi$p.D, q = DIC.zigp$p.D, correction = TRUE)

Clarke.poi.nb <- Clarketest(ll.poi, ll.nb, alpha = 0.05, p = DIC.poi$p.D, q = DIC.nb$p.D, correction = TRUE)
Clarke.poi.gp <- Clarketest(ll.poi, ll.gp, alpha = 0.05, p = DIC.poi$p.D, q = DIC.gp$p.D, correction = TRUE)
Clarke.poi.zip <- Clarketest(ll.poi, ll.zip, alpha = 0.05, p = DIC.poi$p.D, q = DIC.zip$p.D, correction = TRUE)
Clarke.poi.zigp <- Clarketest(ll.poi, ll.zigp, alpha = 0.05, p = DIC.poi$p.D, q = DIC.zigp$p.D, correction = TRUE)
```

LogLike *Individual log-likelihood*

Description

'LogLike' calculates the individual log-likelihood which is needed for 'Clarketest' and 'Vuongtest'.

Usage

```
LogLike(Yin, fm.X, region, regmodel, thinning = 1, burnin = 1)
```

Arguments

Yin	response vector of length n.
fm.X	formula for mean design.
region	region of each observation.
regmodel	the output of model fits obtained by using 'est.sc'.
thinning	number of MCMC iterations between two writing steps (if thinning=1 (default), all states are calculated whereas if e.g. thinning=10 only each 10 iteration is calculated)
burnin	number of steps dropped from the chain to allow for a burn-in phase. Defaults to 1.

Value

ll	is a matrix of dimension (n x (totalit+2)), where each column represents one iteration and contains the likelihood contributions of each observation.

Examples

```
data(sim.Yin)
data(sim.fm.X)
data(sim.region)
data(sim.gmat)
data(sim.nmat)

poi <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region,
model="Poi", sim.gmat, sim.nmat, 3)

ll.poi <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)

# log likelihood for the single iterations
apply(ll.poi$ll,2,sum)
```

---

sim.fm.X *Design matrix*

Description

A design matrix with a binary and a numeric vector.

Usage

```
data(sim.fm.X)
```

Format

A data frame with 5000 observations on the following 2 variables.

`V1` a binary vector

`V2` a numeric vector

Examples

```
data(sim.fm.X)
## maybe str(sim.fm.X) ; plot(sim.fm.X) ...
```

---

sim.gmat — *Spatial adjacency matrix*

Description

Simulated spatial adjacency matrix with 100 regions on 10x10 grid, where entry (i,j) is 1 if region i is a neighbor of region j and 0 else.

Usage

```
data(sim.gmat)
```

Format

A data frame with 100 observations on 100 variables.

Examples

```
data(sim.gmat)
## maybe str(sim.gmat) ; plot(sim.gmat) ...
```

---

sim.nmat — *Matrix of neighbours*

Description

Matrix containing neighbors of each region and number of neighbours of each region.

Usage

```
data(sim.nmat)
```

Format

A data frame with 100 observations. Matrix of neighbours, number of neighbours in last column, number of region in first column.

Examples

```
data(sim.nmat)
## maybe str(sim.nmat) ; plot(sim.nmat) ...
```

---

sim.region *Region vector*

Description

Vector of regions in 1, ..., 100 for each of the 5000 individuals.

Usage

```
data(sim.region)
```

Format

A data frame with 5000 observations with 100 regions.

Examples

```
data(sim.region)
## maybe str(sim.region) ; plot(sim.region) ...
```

---

sim.Yin *Response vector*

Description

Simulated responses with 5000 observations.

Usage

```
data(sim.Yin)
```

Format

A data frame with 5000 observations.

Examples

```
data(sim.Yin)
## maybe str(sim.Yin) ; plot(sim.Yin) ...
```

Vuongtest *Vuong's test for non-nested model comparison*

Description

'Vuongtest' suggests the better of two (not necessarily nested) models according to Vuong's statistic for the parameters in each of the iterations.

Usage

```
Vuongtest(LogLike1, LogLike2, alpha = 0.05, p = NULL, q = NULL,
correction = TRUE)
```

Arguments

`LogLike1, LogLike2`	the output of two model fits obtained by using 'LogLike'.
`alpha`	significance level, defaults to 0.05.
`p, q`	the number of estimated coefficients in models LogLike1 and Loglike2, respectively.
`correction`	boolean, if TRUE (default), the Schwarz correction will be used on the differences of log-likelihoods.

References

Vuong, Q.H. (1989). Likelihood Ratio tests for model selection and nonnested hypotheses. Econometrica 57(2), 307-333.

Schwarz, G. (1978). Estimating the Dimension of a Model. Annals of Statistics 6, 461-464.

See Also

Clarketest

Examples

```
data(sim.Yin)
data(sim.fm.X)
data(sim.region)
data(sim.gmat)
data(sim.nmat)

poi <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region,
model="Poi", sim.gmat, sim.nmat, 3)
nb <- est.sc(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region,
model="NB", sim.gmat, sim.nmat, 3)

DIC.poi <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)
```

```
DIC.nb <- DIC(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, nb)

ll.poi <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, poi)
ll.nb <- LogLike(sim.Yin, ~1+sim.fm.X[,1]+sim.fm.X[,2], sim.region, nb)

Vuong.poi.nb <- Vuongtest(ll.poi, ll.nb, alpha = 0.05, p = DIC.poi$p.D,
q = DIC.nb$p.D, correction = TRUE)
```

Index

∗Topic **datasets**
- sim.fm.X, 8
- sim.gmat, 9
- sim.nmat, 9
- sim.region, 10
- sim.Yin, 10

∗Topic **package**
- spatcounts-package, 2

∗Topic **regression**
- spatcounts-package, 2

∗Topic **spatial**
- spatcounts-package, 2

Clarketest, 2

DIC, 3

est.sc, 5

LogLike, 7

sim.fm.X, 8
sim.gmat, 9
sim.nmat, 9
sim.region, 10
sim.Yin, 10
spatcounts (spatcounts-package), 2
spatcounts-package, 2

Vuongtest, 11