Apresentação SAR

Modelo Espacial AutoRegressivo (SAR)

Rafael Morciani/GRR:20160217

2017-11-29

Resumo

- Modelo autoregressivo espacial.
- Medir a dependência espacial ρ.
- Políticas públicas.
- Máxima verossimilhança.

Introdução

- Simulação no R.
- Funções de log e verossimilhança.
- Não calcularemos a função escore.
- Função "optim".
- Se $\rho = 0$, regressão linear.

Modelo

Y $= \rho\ W$ Y + X $\beta + \epsilon$, $\epsilon \sim N(0; \sigma^2 I)$

Y: Variável dependente;

ρ: Parâmetro espacial responsável por mensurar o grau de dependência espacial;

W: Matriz de vizinhança;

X: Vetor das variáveis independentes;

β: Vetor dos coeficientes de regressão;

ϵ: Erro aleatório;

σ^2: Variância do modelo;

I: Matriz identidade.

Suporte da distribuição: Y $\in \mathbb{R}$

Espaço paramétrico: $\rho \in [-1; 1],\ \sigma^2 \in \mathbb{R}_+^*,\ \beta \in \mathbb{R},\ \mu \in \mathbb{R}$

Inferência

- Estimadores (μ, ρ ,σ^2 e $\beta' s$).
- $\hat{\rho}$ máxima verossimilhança $N_{multivariada}$
- Z $\sim N_m(\mu; \Sigma)$
- $\mu = \beta_0 + \beta_1 X$
- $\Sigma = (I - \rho W)^{-1} \sigma^2 I (I - \rho W')^{-1}$
- $f(y_i; \rho; \mu; \sigma^2) = (2\pi)^{\frac{-p}{2}} \left|\Sigma\right|^{\frac{-1}{2}} e^{\frac{-1}{2}(y_i - \mu)\Sigma^{-1}(y_i - \mu)'}$
- p = Total de observações
- Uma observação (n=1).

Inferência

1. **Verossimilhança**

- $L(\rho; \mu; \sigma^2; y_i) = (2\pi)^{\frac{-p}{2}} \, |\Sigma|^{\frac{-1}{2}} \, e^{\frac{-1}{2}(y_i-\mu)\Sigma^{-1}(y_i-\mu)'}$

2. **log-verossimilhança**

- $l(\rho; \mu; \sigma^2; y_i) = -\frac{1}{2}\left[p\log(2\pi) + log(|\Sigma|) + (Y-\mu)\Sigma^{-1}(Y-\mu)'\right]$

3. **Maximizando a função**

- "optim" ($\hat{\mu}, \hat{\rho}, \hat{\sigma^2}$ e $\hat{\beta'}s$).
- Consistentes, não viciados e eficientes.

4. **TH e IC**

- Teste Wald: $Z_n = \frac{\hat{\theta}-\theta}{\sqrt{V[\hat{\theta}]}} \sim N(0;1)$.
- $IC_{95\%}$.

Implementação Computacional

1. **Dados simulados**

```r
# Criando a estrutura espacial
# Lista de vizinhos
nb <- cell2nb(nrow = 10, ncol = 10)

# Matriz de vizinhanca
matriz = cell2nb(nrow=10,ncol=10,type="rook")
W = nb2mat(matriz)

#Demais variaveis
Sigma <- function(rho, sigma2, W) {
  I <- diag(rep(1,ncol(W)))
  B <- rho*W
  p1 <- solve( (I - B) )
  p2 <- solve( (I - t(B)) )
  out <- sigma2*(p1%*%p2)
  return(out)
}

Sigma2 <- as.matrix(Sigma(rho = 0.5, sigma2 = 1, W = W))

x1 <- seq(-1,1, l = 100)

mu <- 2 + 3*x1

Y <- as.numeric(rmvnorm(1, mean = mu, sigma = Sigma2))
```

2. **Função de Verossimilhança e Log-Verossimilhança**

```
#Verossimilhanca
L <- function(Sigma,Y) {
  p <- length(Y)
  out <- prod( ((2*pi)^(-p/2)) * (1/determinant(Sigma,logarithm=T)) *
               exp((-1/2)*Y%*%solve(Sigma)%*%t(Y)) )
  return(out)
  }

#Log-verossimilhanca
lv <- function(par, W, Y, x1) {
  beta0 <- par[1]
  beta1 <- par[2]
  rho <- par[3]
  sigma2 <- par[4]
  p <- length(Y)
  Sigma2 <- as.matrix(Sigma(rho = rho, sigma2 = sigma2, W = W))
  mu <- beta0 + beta1*x1
  out <- dmvnorm(x = Y, mean = mu, sigma = Sigma2, log = TRUE)
  return(as.numeric(out))
}
```

Implementação Computacional

3. **Maximizando a LV pela "optim"**

Para utilização desta função, precisou definir o espaço paramétrico de cada estimados, para isso foi utilizado o método "L-BFGS-B".

```
est <- optim(par = c(0,0, 0, 1), fn = lv, W = W, Y = Y, x1 = x1,
             method = "L-BFGS-B",
             lower = c(-Inf, -Inf, -0.99, 0.0001), upper = c(Inf, Inf, 0.99, Inf),
             control = list(fnscale = -1), hessian = T)
```

4. **Estimativas obtidas**

```
Beta0_hat <- est$par[1]
Beta1_hat <- est$par[2]
rho_hat <- est$par[3]
sigma2_hat <- est$par[4]
```

Implementação Computacional

5. **Matriz Io e variâncias**

```
#Matriz de informação observada
I_o <- (-est$hessian)
I_o
```

```
##               [,1]          [,2]       [,3]          [,4]
## [1,]  3.784938e+01 -7.105427e-09 -1.9280782 -3.258052e-04
## [2,] -7.105427e-09  1.327925e+01 -0.1868373 -9.445955e-05
## [3,] -1.928078e+00 -1.868373e-01 57.8764267  6.080453e+00
## [4,] -3.258052e-04 -9.445955e-05  6.0804526  2.671964e+01
```

```
V <- sqrt(diag(solve(I_o)))
V
```

```
## [1] 0.1626854 0.2744248 0.1331655 0.1958162
```

Implementação Computacional

6. **Teste Wald e IC de 95%**

Teste Wald: $H_0 : \rho = 0,\ H_1 : \rho \neq 0$

```
rho_0 <- 0
Zn <- (rho_hat - rho_0)/(V[3])
Zn
```

```
## [1] 2.106
```

Intervalo com 95% de confiança

```
inf <- rho_hat - (1.96*V[3])
sup <- rho_hat + (1.96*V[3])
IC <- c(inf,sup)
IC
```

```
## [1] 0.01944221 0.54145091
```

Conclusão

- Valores definidos para os estimadores:

$\rho = 0.5$

$\sigma^2 = 1$

$\beta_0 = 2$

$\beta_1 = 3$

- Estimativas obtidas:

$\hat{\rho} = 0.2804466$

$\hat{\sigma^2} = 1.3679411$

$\hat{\beta_0} = 1.8191485$

$\hat{\beta_1} = 3.4850914$

- Teste Wald

$H_0 : \rho = 0$, $H_1 : \rho \neq 0$

$Z_n = 2.1060004$

Rejeita-se H_0

- Intervalo com 95% de confiança para $\hat{\rho}$

$IC_{95\%}$: [0.0194422, 0.5414509]