

Aproximação de Laplace: exemplo de Tokyo

Elias T. Krainski / eliaskr@ufpr.br /

September 2017

Introdução

Neste tutorial mostramos uma aplicação da aproximação de Laplace para integrar um efeito aleatório Gaussiano.

Inicialmente definimos **funções** para calcular a derivada numérica de primeira e segunda ordem. A seguir consideramos essas funções numa função para fazer a aproximação Gaussiana, como detalhada na seção 4.4.1 de Rue and Held (2005). Essa aproximação foi proposta para inferência em modelos hierárquicos Bayesianos em Rue and Martino (2007).

Vamos exemplificar essa técnica para os dados de **Tokyo**. Inicialmente, calculamos a aproximação para três valores diferentes do parâmetro de variância do efeito aleatório para ilustração do seu efeito no efeito aleatório.

Ao final consideramos o problema de encontrar o valor do parâmetro de variância de **máxima verossimilhança**. Fazemos a predição do efeito aleatório considerando esse valor e comparamos com o resultado obtido pelo pacote **INLA**, que também considera aproximações de Laplace para a distribuição marginal do efeito aleatório, Rue, Martino, and Chopin (2009).

Funções

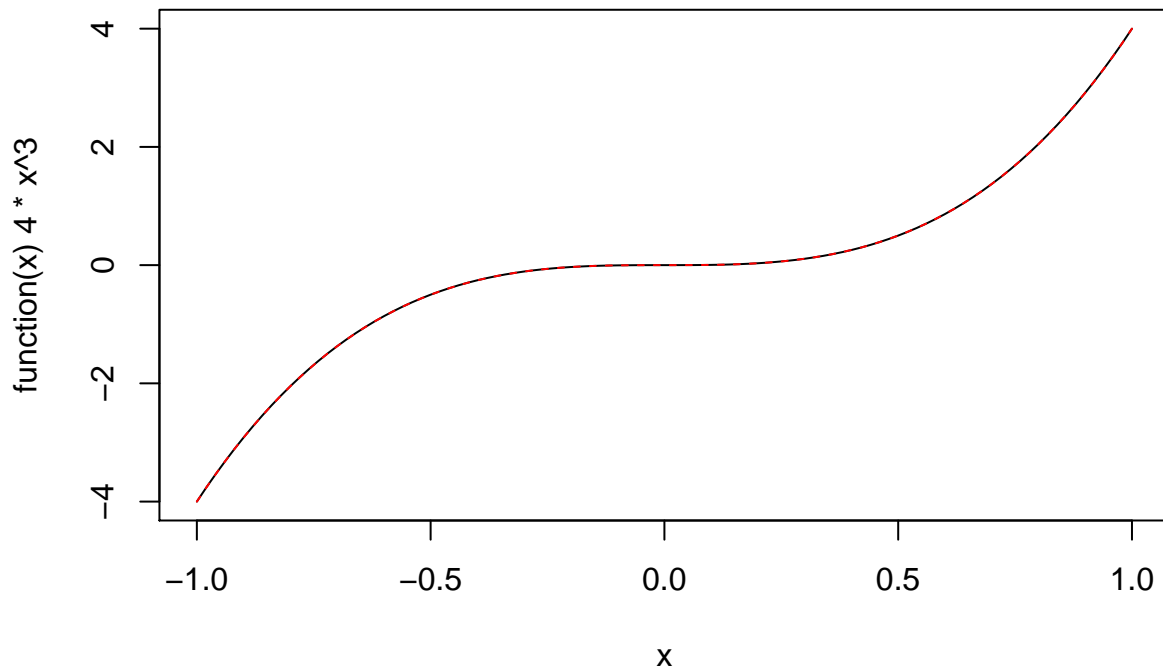
Funções para derivada de 1a e 2a ordem

Derivadas numéricas podem ser calculadas numericamente considerando os coeficientes neste link.

```
### função para calcular a primeira derivada numérica de uma função
numDeriv1 <- function(x, f, h)
  (f(x+h) - f(x-h))/(2*h)
```

Vamos considerar a função $f(x) = x^4$ para ilustração. A seguir visualizamos a derivada analítica, $f'(x) = 4x^3$, (em preto) e a numérica (vermelho tracejado):

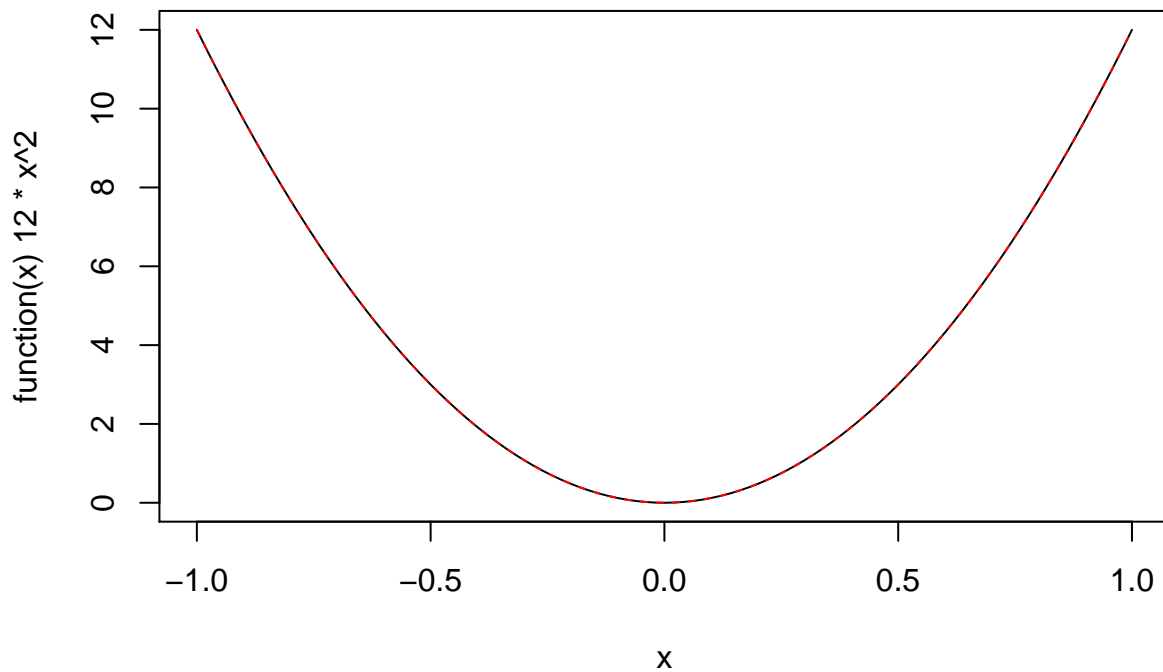
```
plot(function(x) 4*x^3, -1, 1)
d1 <- numDeriv1(
  x = seq(-1, 1, length=101),
  f = function(x) x^4, h=0.01)
lines(seq(-1, 1, length=101), d1, col=2, lty=2)
```



```
### função para calcular a segunda derivada numérica de uma função
numDeriv2 <- function(x, f, h)
  (f(x+h) - 2*f(x) + f(x-h))/(h^2)
```

Vamos também comparar a derivada analítica de segunda ordem, $f''(x) = 12x^2$, com a derivada calculada numericamente.

```
plot(function(x) 12*x^2, -1, 1)
d2 <- numDeriv2(
  x = seq(-1, 1, length=101),
  f = function(x) x^4, h=0.01)
lines(seq(-1, 1, length=101), d2, col=2, lty=2)
```



Aproximação Gaussiana

Os detalhes estão na seção 4.4.1 de Rue and Held (2005). Vamos simplificar o modelo hierárquico ao máximo para ilustrar a técnica.

Inicialmente, assumimos uma distribuição para cada observação y_i , condicional a um efeito aleatório latente b_i : $\pi(y_i|b_i)$. Assumimos que a distribuição conjunta dos dados pode ser escrita como

$$\pi(\mathbf{y}|\mathbf{b}) = \prod_{i=1}^n \pi(y_i|b_i)$$

ou seja, a distribuição de cada observação condicional a b_i é independente das demais.

No segundo nível da hierarquia assumimos uma distribuição Gaussiana para os efeitos aleatórios. Um exemplo é o passeio aleatório Gaussiano

$$\pi(\mathbf{b}|\sigma^2) \propto (2\pi\sigma^2)^{(n-1)/2} e^{-\frac{\mathbf{b}\mathbf{R}\mathbf{b}}{2\sigma^2}}$$

onde \mathbf{R} é a estrutura da matriz de precisão e σ^2 é um parâmetro de variância.

O problema de inferência envolve estimar σ^2 e prever \mathbf{b} . O primeiro passo é calcular a verossimilhança. Porém, essa verossimilhança precisa ser aproximada. Uma forma é

$$L(\sigma^2) = \frac{\pi(\mathbf{b}|\sigma^2)\pi(\mathbf{y}|\mathbf{b})}{\pi(\mathbf{b}|\sigma^2, \mathbf{y})} \approx \frac{\pi(\mathbf{b}^*|\sigma^2)\pi(\mathbf{y}|\mathbf{b}^*)}{\pi_G(\mathbf{b}^*|\sigma^2, \mathbf{y})}$$

em que

$$\log(\pi_G(\mathbf{b}^*|\sigma^2, \mathbf{y})) \propto \log(|\mathbf{R}/\sigma^2 + \text{diag}(\mathbf{c})|)$$

em que \mathbf{b}^* é a moda de $\pi_G(\cdot|\sigma^2, \mathbf{y})$ e $\mathbf{c} = \{c_1, \dots, c_n\}$, onde $c_i = -\frac{\partial^2 \pi(y_i|b_i)}{\partial^2 b_i}$.

Além disso, a predição de \mathbf{b} pode ser feita resolvendo o seguinte sistema

$$[\mathbf{R}/\sigma^2 + \text{diag}(\mathbf{c})]\mathbf{b} = \mathbf{x}$$

com $\mathbf{x} = \{x_1, \dots, x_n\}$, sendo $x_i = \frac{\partial \pi(y_i|b_i^{(0)})}{\partial b_i^{(0)}} - b_i^{(0)} c_i$. Esse sistema é resolvido a para um valor inicial de \mathbf{b} , $\mathbf{b}^{(0)}$, obtendo-se $\mathbf{b}^{(1)}$ e sucessivamente por algumas iterações, sendo que cinco iterações é tipicamente suficiente para se encontrar a moda, \mathbf{b}^* .

A função a seguir implementa esse algoritmo:

```
### função para calcular a aproximação Gaussiana
Ga <- function(b, Q, f, niter=5, h=.Machine$double.eps^0.3) {
  for (j in 1:niter) {
    cc <- -numDeriv2(b, f, h)
    cc[cc<0] <- 0
    xx <- numDeriv1(b, f, h) + b*cc
    Qn <- Q + Diagonal(x=cc)
    L <- chol(Qn)
    new <- drop(solve(L, solve(t(L), xx)))
    b <- new
  }
  return(list(mu=b, bb=xx, cc=cc, Q=Qn, L=L,
             sldL=sum(log(diag(L)))))
}
```

Aplicação para os dados de Tokyo

Dados de Tóquio

```
### carrega os dados de Tokyo
data(Tokyo, package='INLA')
Tokyo[1:3,]
```

```
##   y n time
## 1 0 2     1
## 2 0 2     2
## 3 1 2     3
n <- nrow(Tokyo)
```

Modelo

Verossimilhança

$$y_i \sim \text{binomial}(n_i, p_i)$$

onde

$$p_i = \text{plogis}(b_i) = 1/(1 + e^{-b_i})$$

```
### função para calcular o log da função de verossimilhança
llfbt <- function(x)
  dbinom(Tokyo$y, Tokyo$n, plogis(x), log=TRUE)
```

A distribuição de **b** considera a matriz de estrutura do modelo de passeio aleatório cíclico:

```
### define R matrix
library(Matrix)
Rt <- Matrix(toeplitz(c(2,-1,rep(0,n-3),-1)))
```

Avalia em três valores de σ^2

Para efeito de comparação mais adiante com os resultados do INLA, Rue, Martino, and Chopin (2009), vamos considerar uma reparametrização. Definimos o parâmetro de precisão $\tau = 1/\sigma^2$. Vamos considerar três valores diferentes para τ : 10, 100, 1000. Para cada um desses valores, vamos calcular a aproximação Gaussiana:

```
### define valor de sigma e valor inicial de b
inicial.b <- rep(0, n)
taus <- c(1e1, 1e2, 1e3)

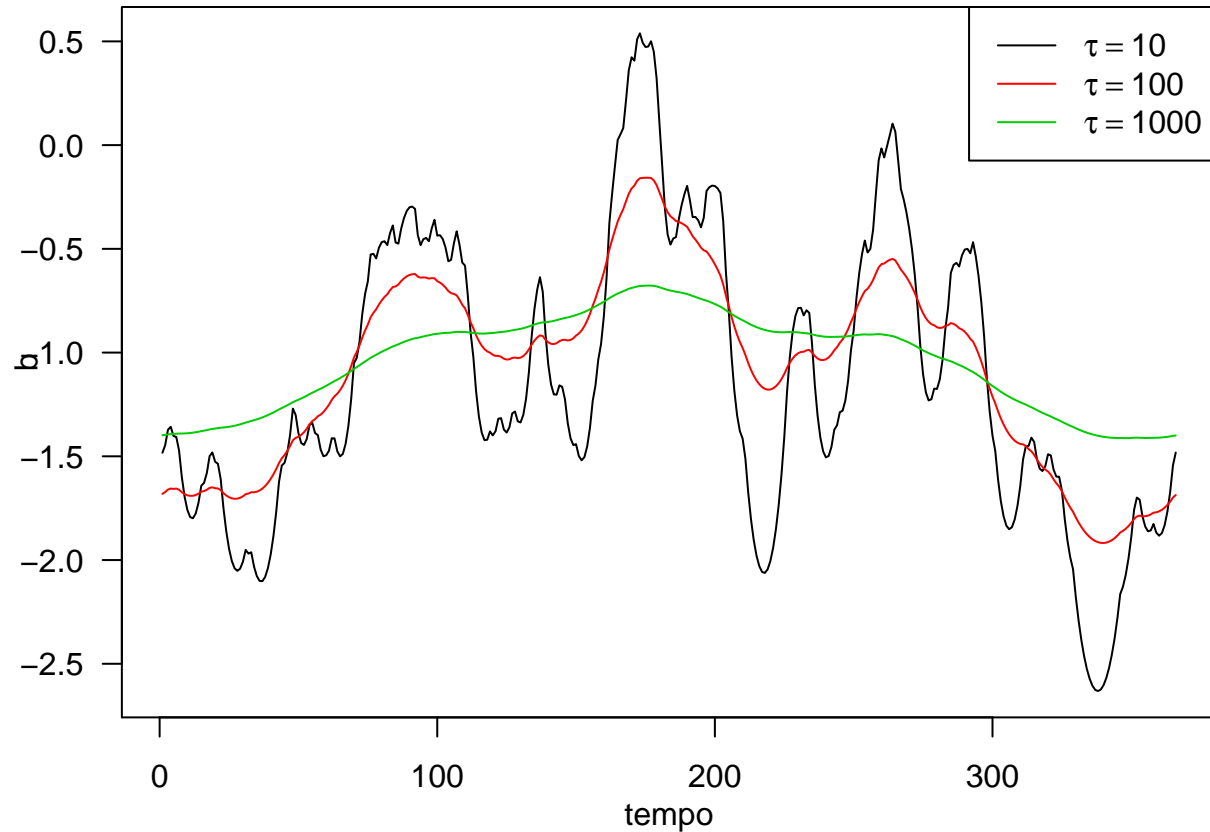
### calcula aproximação para os três casos
a1 <- Ga(inicial.b, taus[1]*Rt, llfbt)
a2 <- Ga(inicial.b, taus[2]*Rt, llfbt)
a3 <- Ga(inicial.b, taus[3]*Rt, llfbt)
```

Visualiza a moda de **b** condicional à cada valor de τ .

```

par(mar=c(3,3,1,1), mgp=c(2,1,0), las=1)
plot(a1$mu, xlab='tempo', ylab='b', type='l', col=1)
lines(a2$mu, col=2)
lines(a3$mu, col=3)
legend('topright', as.expression(lapply(taus, function(tt) bquote(tau == .(tt)))),
      lty=1, lwd=1, col=1:3)

```



Para qual valor de σ^2 o resultado é melhor?

Máxima verossimilhança

A função de log-verossimilhança aproximada de $\tau = 1/\sigma^2$ é implementada a seguir

```

ltau <- function(ltau){
  tau <- exp(ltau)
  ga <- Ga(rep(0,n), tau*Rt, llfbt)
  p.b <- (n*ltau - tau*sum((ga$mu*%Rt)*ga$mu))/2
  lik <- sum(llfbt(ga$mu))
  p.G <- ga$sldL
  return(p.G - p.b - lik)
}

```

Para encontrar a moda à *posteriori* fazemos:

```
(otau <- optim(1, ltau, method='BFGS'))$par
```

```
## [1] 3.257291
```

Moda de b na moda de τ

```
ga <- Ga(rep(0,n), exp(otau$par)*Rt, llfbt)
```

Compara com resultado do INLA: inferência Bayesiana

Vamos considerar paradigma Bayesiano e com o uso de duas aproximações de Laplace aninhadas e integradas, *Integrated Nested Laplace Approximations* - *INLA*, proposta em Rue, Martino, and Chopin (2009). Vamos usar o pacote **INLA**, disponível aqui. Além da aproximação de Laplace para $p(\tau)$, o algoritmo INLA calcula uma aproximação de Laplace para cada $p(b_i)$ e inclui a integração/marginalização sobre τ .

```
library(INLA)
res <- inla(y ~ 0 + f(time, model='rw1', cyclic=TRUE, constr=FALSE),
           'binomial', data=Tokyo, Ntrials=n,
           control.predictor=list(compute=TRUE))
```

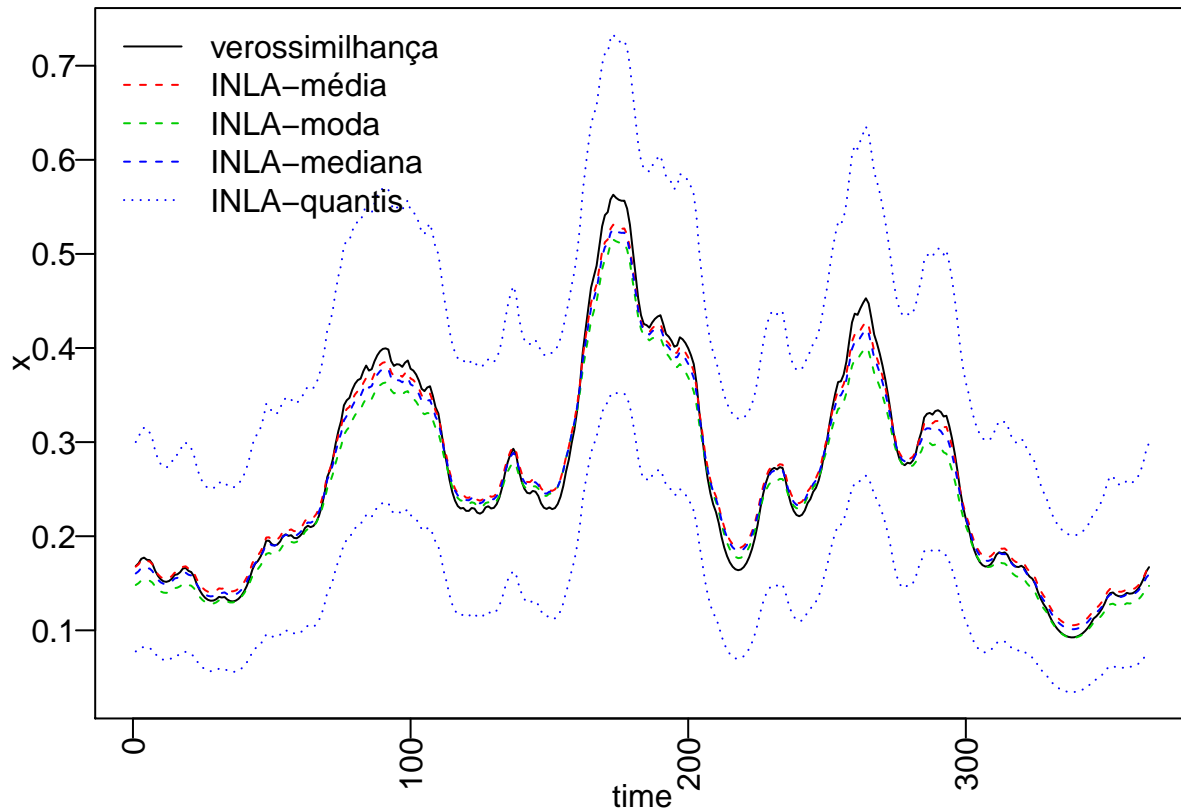
Valor de τ :

```
c(otau$par, res$mode$theta)
```

```
##                      Log precision for time
##                3.257291                3.494527
```

Probabilidades de chuva para cada dia

```
par(mfrow=c(1,1), mar=c(3,3,1,1), mgp=c(1.5,0.5,0), las=2)
plot(plogis(ga$mu), type='l', xlab='time', ylab='x',
     ylim=range(res$summary.fitted.val[,c(3,5)]))
lines(res$summary.fitted.values$mean, col=2, lty=2)
lines(res$summary.fitted.values$mode, col=3, lty=2)
lines(res$summary.fitted.values$'0.5q', col=4, lty=2)
for(j in c(3,5)) ## 0.025q e 0.975q
  lines(res$summary.fitted.val[,j], col=4, lty=3)
legend('topleft', c('verossimilhança', 'INLA-média', 'INLA-moda', 'INLA-mediana', 'INLA-quantis'),
      lty=c(1,2,2,2,3), col=c(1:4,4), bty='n')
```



Observamos que a moda de \mathbf{b} para o valor de τ que maximiza a verossimilhança aproximada é ligeiramente menos suave que a média, moda e mediana da distribuição à posteriori.

Referências bibliográficas

- Rue, H., and L. Held. 2005. *Gaussian Markov Random Fields: Theory and Applications*. Monographs on Statistics & Applied Probability. Boca Raton: Chapman; Hall.
- Rue, H., and S. Martino. 2007. “Approximate Bayesian Inference for Hierarchical Gaussian Markov Random Field Models.” *Journal of Statistical Planning and Inference* 137:3711–3192.
- Rue, H., S. Martino, and N. Chopin. 2009. “Approximate Bayesian Inference for Latent Gaussian Models Using Integrated Nested Laplace Approximations (with Discussion).” *Journal of the Royal Statistical Society, Series B* 71 (2):319–92.