

Métodos Analíticos e Numéricos em Inferência Bayesiana: Parte 1

Prof. Caio Azevedo

Cálculo da constante de normalização

- Exemplo: posteriori associada ao modelo $\text{gama}(r, \lambda)$.
- Suponha uma amostra aleatória de tamanho n de $X \sim \text{gama}(r, \lambda)$, com r conhecido.

$$p(x|r) = \frac{1}{\lambda^r \Gamma(r)} e^{-x/\lambda} x^{r-1} \mathbb{1}_{(0, \infty)}(x)$$

- Priori: $p(r)$ alguma fdp com suporte em \mathcal{R}^+ .
- Verossimilhança:

$$p(\mathbf{x}|r) = \frac{1}{\lambda^{nr} \Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1}$$

Cont.

- Posteriori :

$$p(r|\mathbf{x}) = \frac{\frac{\lambda^{nr}}{\Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1} p(r)}{\int_0^\infty \frac{\lambda^{nr}}{\Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1} p(r) dr}$$

- A integral correspondente à constante de normalização não tem solução analítica exata, independentemente da escolha da priori.

Cont.

- Métodos Analíticos:
 - Aproximimação à distribuição normal multivariada.
 - Método de Laplace.
- Métodos Numéricos:
 - Integração/Maximização numéricas.
 - Métodos de simulação não iterativos (como no problema de Berhens-Fisher, comparação de médias de distribuições normais).
 - Métodos de simulação de Monte Carlo via Cadeias de Markov.

Aproximação à distribuição Normal Multivariada

- Lembrando: $p(\theta|\mathbf{x})$ posteriori uniparamétrica, $p(\boldsymbol{\theta}|\mathbf{x})$ posteriori multiparamétrica.
- Sob certas condições de regularidade, pode-se aproximar a distribuição à posteriori de interesse como:

$$p(\theta|\mathbf{x}) \approx \frac{\sqrt{h_n}}{\sqrt{2\pi}} \exp \left\{ -\frac{h_n}{2} (\theta - m_n)^2 \right\}$$

em que $h_n = h_0 + h(\tilde{\theta})$, $m_n = h_n^{-1} (h_0 m_0 + h(\tilde{\theta})\tilde{\theta})$.

Cont.

- m_0 é a moda da distribuição à posteriori (calculada, eventualmente, de modo numérico).
- $\tilde{\theta}$ é a estimativa de máxima verossimilhança de θ (calculada, eventualmente, de modo numérico).
- $h_0 = -\frac{d^2 \ln p(\theta)}{d\theta^2} \Big|_{\theta=m_0}$.
- $h(\tilde{\theta}) = -\frac{d^2 \ln p(\mathbf{x}|\theta)}{d\theta^2} \Big|_{\theta=\tilde{\theta}}$.

Característica da AN (aproximação à distribuição normal)

- Relativamente rápida.
- Útil quando a obtenção dos máximos e do estimador de MV é simples.
- Os estimadores (EAP, MAP, MeAP, EPAP) e intervalos de credibilidade não, necessariamente, respeitarão espaço paramétrico.
- Mesmo no caso univariado para funções complicadas: vários máximos locais, assimetria, etc.

Na prática

- Suponha que $r \sim \text{gama}(\alpha, \gamma)$, $\alpha > 0$. Assim

$$p(r) = \frac{1}{\gamma^\alpha \Gamma(\alpha)} e^{-r/\gamma} r^{\alpha-1} \mathbb{1}_{(0, \infty)}(r)$$

- Assim

$$-\frac{d^2 \ln p(r)}{dr^2} = \frac{(\alpha - 1)}{r^2}$$
$$-\frac{d^2 \ln p(\mathbf{x}|r)}{dr^2} = \frac{n}{(\Gamma(r))^2} \left[\Gamma''(r)\Gamma'(r) - (\Gamma'(r))^2 \right] = n\Psi'(r)$$

- Neste caso, $m_0 = \gamma(\alpha - 1)$. Contudo, a estimativa de mv de r não é obtível analiticamente. Assim, algum método numérico de maximização deve ser empregado.

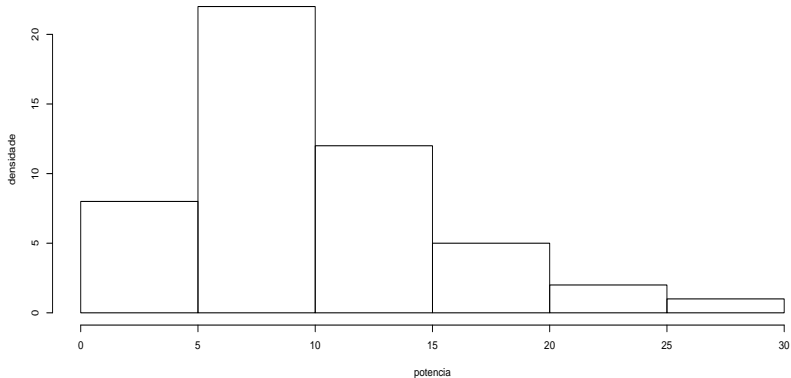
Usando o programa R

- A função *optim* do programa R, permite maximizar funções utilizando algum algoritmo numérico. Particularmente, pode-se usar o método BFGS (ver notas de aula de métodos numéricos para Estatística).
- Tal função já vem pré instalada no R.
- Para utilizar os algoritmos de Newton-Raphson ou Escore de Fisher, pode-se recorrer ao pacote maxLik (usando a função maxLik).

Exemplo: dados da potência de turbina de aviões

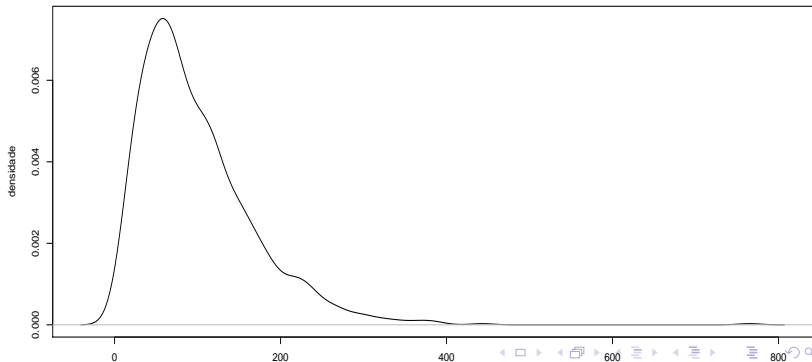
- Conjuntos de dados relativos ao desempenho de 5 tipos de turbina de avião. Tempo (em unidade de milhões de ciclos) até a perda de velocidade.
- Vamos desconsiderar que as turbinas são de diferentes tipos.
- Modelar os dados através da distribuição gama.

Histograma dos dados das turbinas



Aplicação da aproximação à normal

- Vamos considerar $\alpha = 2$, $\gamma = 50$. Isto implica que $\mathcal{E}(r) = 100$, $\mathcal{V}(r) = 5000$.



Função optim

- Comandos no R: obtenção da estimativa de máxima verossimilhança

```
sem título
fr<- function(r,lambda,n,v.x)
{
logl<- -n*loggamma(r)-n*r*log(lambda) - n*mean(v.x)/lambda + (r-1)*sum(log(v.x))
return(-logl)
}

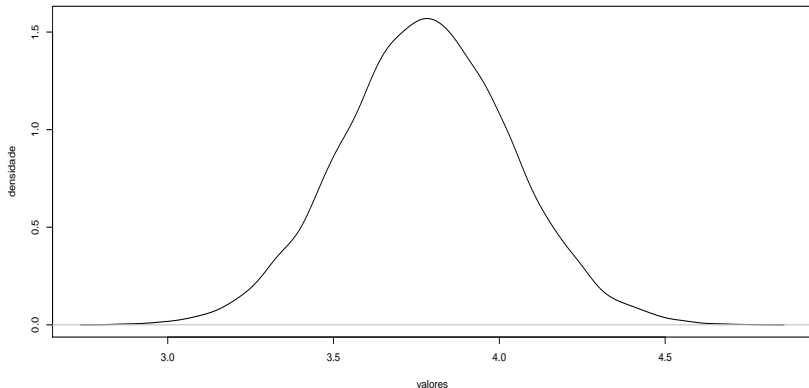
resultlik<-optim(mean(v.x)/lambda,fr,lambda=lambda,n=n,v.x=v.x,method="BFGS",hes
sian=T)

r.mv <- resultlik$par
h.r <- resultlik$hessian
h.n<- h0 + h.r
m.n <- solve(h.n)*(h0*m0 + h.r*r.mv)
```

Aplicação da aproximação à normal

- Neste caso, $m_0 = 50$, $\tilde{r} = 3,78$, $h_0 = 0,0004$, $h(\tilde{r}) = 15,12$
- Assim, $r|\mathbf{x} \approx N(3,78, 0,07)$.
- Portanto, $\hat{r}_{EAP} \approx \hat{r}_{MAP} \approx \hat{r}_{MeAP} \approx 3,78$, $EPAP(r) \approx 0,26$.
- $IC_B(r, 95\%) \approx [3,28; 4,29]$.

Aproximação à normal



Aproximação de Laplace (AL)

- Suponha que desejamos calcular $\int_a^b e^{Mf(x)} dx$. Suponha que existe um único máximo global para $f(\cdot)$
- Seja x_0 o máximo global de $f(\cdot)$. Então :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + R$$

$$R = O((x - x_0)^3)$$

- Então

$$f(x) \approx f(x_0) - \frac{1}{2}|f''(x_0)|(x - x_0)^2$$

Cont.

■ Aproximação de Laplace:

$$\begin{aligned}\int_a^b e^{Mf(x)} dx &\approx e^{Mf(x_0)} \int_a^b e^{-M|f''(x_0)|(x-x_0)^2/2} dx \\ &= \sqrt{2\pi} e^{Mf(x_0)} \sigma \left\{ \Phi\left(\frac{b-x_0}{\sigma}\right) - \Phi\left(\frac{a-x_0}{\sigma}\right) \right\}\end{aligned}$$

$\sigma = \sqrt{\sigma^2}$, $\sigma^2 = (M|f''(x_0)|)^{-1}$, $\Phi(\cdot)$ é a fda da normal padrão.

Característica da AL

- Relativamente rápida.
- Adequada quando o integrando é unimodal e o máximo pode ser obtido facilmente (ainda que seja necessário utilizar métodos numéricos).
- Inadequada: em integrais múltiplas à medida que a dimensão aumenta e/ou o máximo é complicado de ser obtido.
- Os estimadores (EAP, MAP, MeAP, EPAP) e intervalos de credibilidade não, necessariamente, respeitarão espaço paramétrico.
- Mesmo no caso univariado para funções complicadas: vários máximos locais, assimetria, etc.

Cont. do exemplo da distribuição gama

- Nosso objetivo é utilizar a aproximação de Laplace para calcular:

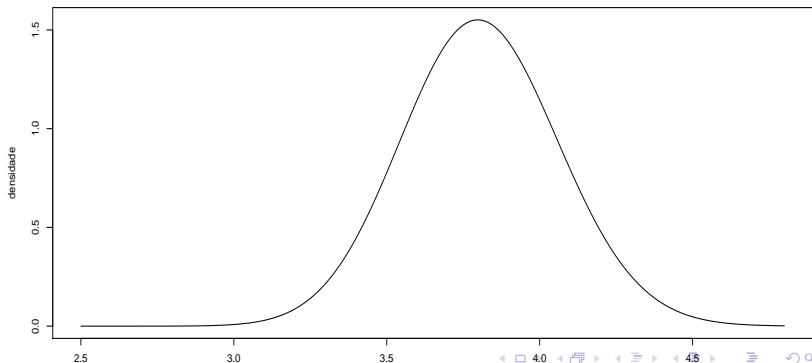
$$\int_0^\infty g(r) dr = \int_0^\infty e^{f(r)} dr, \text{ em que } f(r) = \ln g(r), M = 1$$

$$g(r) = \frac{\lambda^{nr}}{\Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1} \frac{\gamma^\alpha}{\Gamma(\alpha)} e^{-r/\gamma} r^{\alpha-1}$$

- Note que, neste caso, o ponto de máximo r_0 corresponde à moda da posteriori (MAP).
- Além disso, $f''(r) = - \left[n\Psi'(r) + \frac{\alpha-1}{r} \right]$

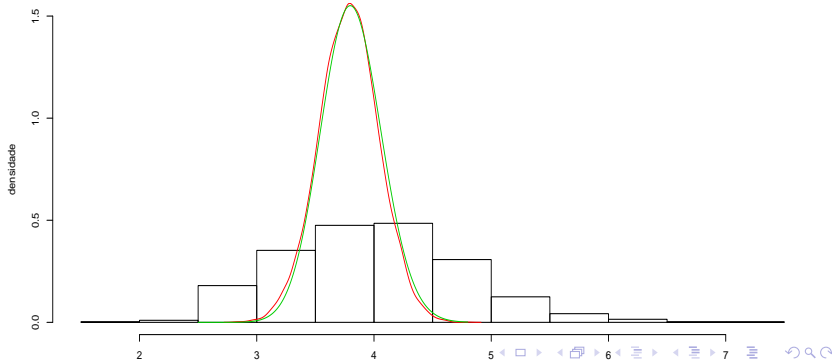
Cont. do exemplo da distribuição gama

- Assim $\int_0^{\infty} g(r) dr \approx e^{f(r_0)} \sqrt{2\pi} (f''(r_0))^{-1} = 1,526686e - 67$



Cont. do exemplo da distribuição gama

■ Comparação



Discussão

- Posteriori assimétrica (aproximações impõem comportamento simétrico).
- Tamanho da amostra ($n = 50$).
- O parâmetro λ é desconhecido (estimativa utilizada).

Como obter na aproximação de Laplace

- Note que a AL fornece uma aproximação apenas da constante de normalização.
- Como obter estimativas pontuais e intervalares usando-a?
- A rigor, teríamos de aproximar analiticamente as integrais:
 $\mathcal{E}(g(r)|\mathbf{x})$.
- Moda: Maximizar a posteriori numericamente.
- Mediana e intervalos de credibilidade: Obter quantis numericamente.
- Alternativa mais prática: aproximar as integrais numericamente.

Integração por Quadratura

- Substituir (no caso univariado) o cálculo da área sob a curva pela soma das áreas de um número finito de retângulos.
- Substituir (no caso multivariado) o cálculo da área sob a curva pela soma das áreas de um número finito de paralelepípedos.
- Ponto-chave: Definir os pontos e pesos de quadratura.
- Formas de cálculo dos pontos e pesos:
 - Não-adaptativa: mantem-se fixo os pontos e pesos.
 - Adaptativa: muda-se os pontos e pesos.

Características

- Em geral, na Estatística, deseja-se calcular $\mathcal{E}(w(X)) = \int_a^b w(x)f(x)dx$, onde $f(\cdot)$ é alguma fdp.
- Sendo assim, $f(\cdot)$ terá massa relevante em apenas um subconjunto de \mathcal{R} .
- Define-se um conjunto de pontos, x_1, x_2, \dots, x_m e os respectivos pesos associados $A_1, A_2, \dots, A_m, A_i = A(x_i)$. Em geral, os pesos correspondem aos comprimentos dos intervalos determinados pelos pontos de quadraturas.
- Pode-se imaginar que cada ponto corresponde ao valor médio de intervalos de mesmo comprimento.

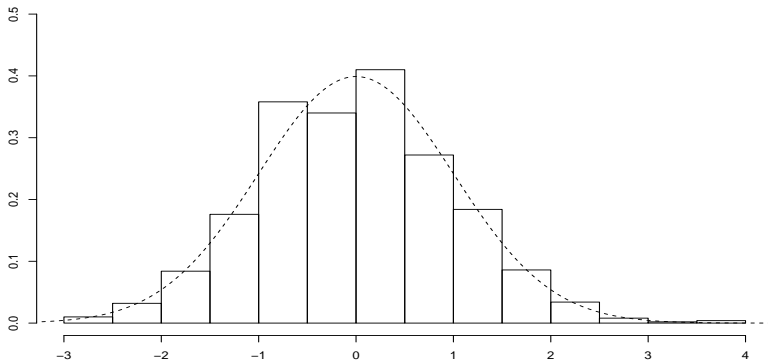
Cont.

- Assim

$$\int_a^b w(x)f(x)dx \approx \sum_{i=1}^m A_i w(x_i)f(x_i)$$

- Existem várias formas de se determinar os pontos e pesos. Em geral, faz-se, em torno do máximo (moda) de $f(\cdot)$. Pontos ótimos, em geral, são obtidos através de certos polinômios.
- Polinômios de Gauss-Hermite, Gauss-Legendre, Jacobi, Legendre, Gauss-Laguerre etc. Veja Abramowitz & Stegun (1972).

Graficamente



Exemplo: família de localização escala

- Seja $Y = \mu + \sigma Z$, $Z \sim D(\theta)$, $\mu \in \mathcal{R}$, $\sigma \in \mathcal{R}^2$, $\theta \in \mathcal{R}^p$.
- Exemplos:
 - 1 $Z_1 \sim N(0, 1)$.
 - 2 $Z_2 \sim t_\nu$.
 - 3 $Z_3 \sim SN(0, 1, \lambda)$.
- Particularmente, o máximo de Z_1 e Z_2 são iguais à 0.
- Qualquer integral envolvendo estas variáveis podem ser facilmente resolvidas.

Distribuição Normal

- Seja $f(x) \sim N(\mu, \sigma^2)$ e defina $h = \int_a^b w(x)f(x)dx$.
- Defina m pontos igualmente espaçados, x_1, x_2, \dots, x_m , $x_1 = \mu - 4\sigma$ e $x_m = \mu + 4\sigma$.
- Se $x_1 < a$, faça $x_1 = a$, se $x_m > b$, faça $x_m = b$.
- Calcule $h \approx \sum_{i=1}^m A_i w(x_i) f(x_i)$.

Distribuição t de Student

- Seja $f(x) \sim t_\nu(\mu, \sigma^2)$ e defina $h = \int_a^b w(x)f(x)dx$.
- Defina p_1 e p_2 ($p_1 < p_2$) probabilidades de interesse.
- Dado ν , calcule y_1, y_2 tais que $p_1 = F(y_1)$,
 $p_2 = F(y_m)$, $Y \sim t_\nu(0, 1)$.
- Defina m pontos igualmente espaçados, y_1, y_2, \dots, y_m . Depois,
calcule $x_i = \sigma y_i + \mu$.
- Se $x_1 < a$, faça $x_1 = a$, se $x_m > b$, faça $x_m = b$.
- Calcule $h \approx \sum_{i=1}^m A_i w(x_i) f(x_i)$.

Distribuição $\text{gama}(r, \lambda)$

- Seja $f(x) \sim \text{gama}(r, \lambda)$ e defina $h = \int_a^b w(x)f(x)dx$.
- Defina p_1 e p_2 probabilidades de interesse.
- Dado $\text{gama}(r, \lambda)$, calcule x_1, x_2 tais que $p_1 = F(x_1)$, $p_2 = F(x_m)$.
- Defina m pontos igualmente espaçados, x_1, x_2, \dots, x_m .
- Se $x_1 < a$, faça $x_1 = a$, se $x_m > b$, faça $x_m = b$.
- Calcule $h \approx \sum_{i=1}^m A_i w(x_i) f(x_i)$.

Quadratura Adptativa

- Atualização dos pontos e pesos de quadratura.
- Nosso interesse continua sendo $h = \int_a^b w(x)f(x)dx$.
- Algoritmo
 - Calcule duas aproximações para h,
 $S_1 = S(f(x); a, b) = \sum_{i=1}^m A_i w(x_i) f(x_i)$ e
 $S_2 = S(f(x); a, (a + b)/2) + S(f(x); (a + b)/2, b)$
 - Calcule $\epsilon = |S_1 - S_2|$.
 - Se $\epsilon < \tau$, pare, caso contrário, aplique o passo 1, recursivamente, até que a precisão requerida seja atingida.

Quadratura Adptativa no R

- Função **integrate** (já vem instalada nas versões mais recentes).
- Permite calcular integrais, usando a quadratura adaptativa, para funções pré-definidas pelo usuário, em um dado intervalo.

Voltando ao exemplo

- Posteriori :

$$p(r|\mathbf{x}) = \frac{\frac{\lambda^{nr}}{\Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1} p(r)}{\int_0^\infty \frac{\lambda^{nr}}{\Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1} p(r) dr}$$

- Momentos à posteriori

$$\mathcal{E}(h(r)|\mathbf{x}) = \frac{\int_0^\infty h(r) \frac{\lambda^{nr}}{\Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1} p(r) dr}{\int_0^\infty \frac{\lambda^{nr}}{\Gamma(r)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r-1} p(r) dr}$$

- Numericamente

$$\mathcal{E}(h(r)|\mathbf{x}) \approx \frac{\sum_{j=1}^m h(r_j) \frac{\lambda^{nr_j}}{\Gamma(r_j)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r_j-1} p(r_j) A_j}{\sum_{j=1}^m \frac{\lambda^{nr_j}}{\Gamma(r_j)^n} e^{-n\bar{x}/\lambda} \prod_{i=1}^n x_i^{r_j-1} p(r_j)}$$

Função integrate

■ Comandos no R: cálculo da densidade

```
fpost<- function(r,lambda,n,v.x,alphap,gammap)
  Sem título
{
logl<- -n*loggamma(r)-n*r*log(lambda) - n*mean(v.x)/lambda + (r-1)*sum(log(v.x))
-r/gammap +(alphap-1)*log(r) - alphap*log(gammap) - loggamma(alphap)

return(exp(logl))
}

cn<-integrate(fpost,0,Inf,lambda,n,v.x,alphap,gammap)$value

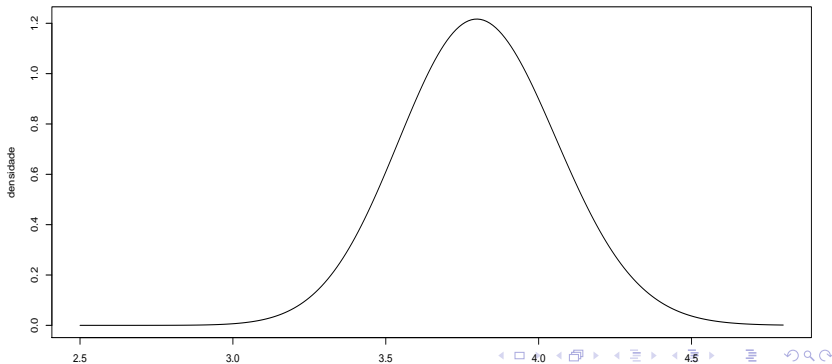
auxpost <- seq(2.5,4.8,0.01)

fpostAL <- (fpost(auxpost,lambda,n,v.x,alphap,gammap))/cn

plot(auxpost,fpostAL,type="l",xlab="valores",ylab="densidade",main="")
```

Função integrate

■ Posteriori



Função integrate

■ Comandos no R: cálculo da esperança a posteriori

```
                                Sem título
fpostEAP<- function(r,lambda,n,v.x,alphap,gammap)

{
logl<- -n*loggamma(r)-n*r*log(lambda) - n*mean(v.x)/lambda + (r-1)*sum(log(v.x))
-r/gammap +(alphap-1)*log(r) - alphap*log(gammap) - loggamma(alphap)

return(r*exp(logl))
}

EAP<-integrate(fpostEAP,0,Inf,lambda,n,v.x,alphap,gammap)$value/cn
```

Função integrate

■ Comandos no R: cálculo do desvio-padrão a posteriori

```

                                Sem título
fpostEAP2<- function(r,lambda,n,v.x,alphap,gammap)
{
logl<- -n*loggamma(r)-n*r*log(lambda) - n*mean(v.x)/lambda + (r-1)*sum(log(v.x))
-r/gammap +(alphap-1)*log(r) - alphap*log(gammap) - loggamma(alphap)

return((r^2)*exp((logl)))
}

EAP<-sqrt(integrate(fpostEAP2,0,Inf,lambda,n,v.x,alphap,gammap)$value/cn
-EAP^2)
```

Continuação: Moda a Posteriori

- Maximização numérica da posteriori (para obter o MAP) usando a função optim. Note que, é equivalente a maximizar a log a posteriori, ou seja:

$$\ln p(r|\mathbf{x}) = \ln p(\mathbf{x}|r) + \ln p(r) + cn$$

cn = constante de normalização.

- Pode-se usar $-\left(\frac{d^2 \ln p(r|\mathbf{x})}{dr^2}\right)^{-1/2} |_{r=r_0}$ como medida de precisão associada ao MAP, em que r_0 é a estimativa MAP.

Continuação: Mediana a Posteriori

- Por definição

$$F(r_{MeAP}|\mathbf{x}) = 0,5$$

- Podemos resolver a equação acima usando o algoritmo de Newton Raphson:

$$r^{(t)} = r^{(t-1)} - \left(p(r^{(t-1)}|\mathbf{x}) \right)^{-1} F(r^{(t-1)}|\mathbf{x}), t = 1, 2, \dots$$

- A quantidade $\sqrt{p(r_{MeAP}|\mathbf{x})^{-1/2}}$ pode ser usada como medida de precisão associada ao $MeAP(r)$.

Função integrate

■ Comandos no R: cálculo da mediana a posteriori

Sem título

```
q10<-4
gammaIC <-0.95
iter <-1
while(iter<=10)
{
gr1 <- integrate(fpost,0,q10,lambda,n,v.x,alpha,gamma)$value/cn -0.5
hr1 <- fpost(q10,lambda,n,v.x,alpha,gamma)/cn
q1t <- q10 - solve(hr1)*gr1
iter <- iter + 1
q10 <- q1t
}
```