Inferência baseada em Verossimilhança para o modelo de regressão Poisson

1 Introdução

O objetivo deste relatório é mostrar o procedimento computacional para inferência baseada em Verossimilhança para dados com distribuição POisson. O processo será detalhado passo-a-passo.

- Conhecer a função densidade probabilidade da variável aleatória, como exemplo considere que $X \sim P(\lambda)$, então a função densidade probabilidade:

$$P(Y = y) = \frac{\exp^{-\lambda} \lambda^y}{y!}, \qquad y = 0, 1, 2, \ldots$$

- O objetivo da inferência neste modelo é inferir sobre o valor do parâmetro λ, a partir de uma amostra y_i observada. Sendo assim, vamos obter uma amostra do modelo, para isto, precisamos fixar um valor para o parâmetro do modelo, neste caso usamos $\lambda = 10$.

```
> set.seed(123)
> y <- rpois(100, lambda = 10)
```

- Obtida a amostra, começamos o procedimento de inferência baseado na Verossimilhança escrevendo a função de Verossimilhança.

$$L(\lambda) = \prod_{i=1}^{n} \frac{\exp^{-\lambda} \lambda_i^y}{y_i!}$$

- Vamos escrever uma função R com a expressão da Verossimilhança

```
> verossimilhanca <- function(lambda, amostra) {
+     ll <- prod(dpois(amostra, lambda = lambda))
+     return(ll)
+ }
```

- Para simplificar as contas analiticas podemos usar a log-Verossimilhança,

$$l(\lambda) = -\lambda n + \sum_{i=1}^{n} y_i \log(\lambda) - \sum_{i=1}^{n} \log(y_i!)$$

- Da mesma forma escrevemos esta função em R,

```
> log.verossimilhanca <- function(lambda, amostra) {
+     ll <- sum(dpois(amostra, lambda = lambda, log = TRUE))
+     return(ll)
+ }
```

- Nosso objetivo é encontrar o ponto de máximo desta função, podemos fazer de três formas básicas: graficamente, analiticamente usando ferramentas de cálculo e numericamente. Para o primeiro caso podemos construir três gráficos explorando a verossimilhança $L(\lambda)$, a log-Verossimilhança $l(\theta)$ e a verossimilhança relativa $L(\lambda)/L(\hat{\lambda})$. Estes gráficos são apresentados na figura abaixo.

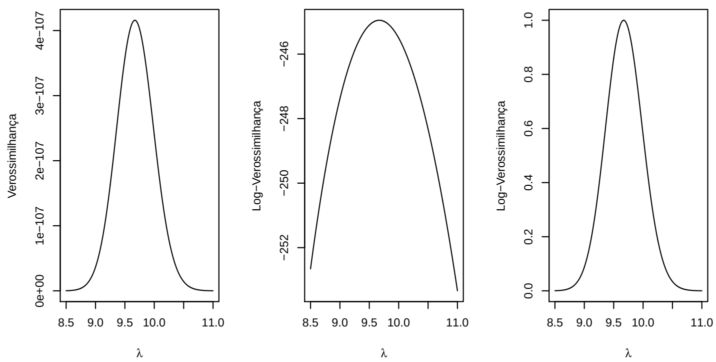

Figura 1: Verossimilhança, log-Verossimilhanca e Verossimilhança Relativa.

- Feitos os gráficos podemos encontrar o ponto de máximo, com os seguintes comandos

```
> saida <- data.frame(log.vero, grid.theta)
> EMV <- saida[which(saida$log.vero == max(log.vero)), ]$grid.theta
> EMV

[1] 9.670117
```

- Uma forma mais eficiente de encontrar o ponto de máximo é quando possível usar ferramentas de cálculo, começamos derivando a log-Verossimilhança em relação ao parâmetros de interesse:

$$U(\lambda) = -n + \frac{\sum_{I=1}^{n} y_i}{\lambda}$$

- Igualando $U(\lambda)$ a zero encontramos o EMV,

$$\hat{\lambda} = \frac{\sum_{i=1}^{n} y_i}{n}$$

- Aplicando no conjunto de dados temos,

```
> EMV.analitico = sum(y)/length(y)
> EMV.analitico

[1] 9.67
```

- Uma outra forma mais geral é encontrar o ponto de máximo da função através de um algoritmo numérico, no R a função *optim*() implementa vários algoritmos

```
> resultado <- optim(c(5), log.verossimilhanca, amostra = y, method = "BFGS",
+     control = list(fnscale = -1), hessian = TRUE)
> resultado$par

[1] 9.67
```

- Encontrado o EMV o próximo passo é encontrar um intervalo de confiança, o método padrão é usar a função *deviance*,

$$D(\lambda) = 2(l(\hat{\lambda}) - l(\lambda))$$

- Uma opção mais fácil é usar o seguinte resultado,

$$\hat{\lambda} \sim aN(\lambda, I(\hat{\lambda})^{-1})$$

onde

$$I(\hat{\lambda}) = E - \frac{d^2}{d\lambda^2} l(\lambda)$$

- No caso da Poisson temos,

$$I(\hat{\lambda})^{-1} = E(\frac{\hat{\lambda}^2}{\sum_{i=1}^{n} y_i}$$

lembrando que da esperança da Poisson temos,

$$I(\hat{\lambda})^{-1} = \frac{\hat{\lambda}}{n}$$

- Fazendo as contas no R temos,

```
> ic.assintotico = c(EMV.analitico - qnorm(0.975) * sqrt(EMV.analitico/length(y)),
+     EMV.analitico + qnorm(0.975) * sqrt(EMV.analitico/length(y)))
> ic.assintotico

[1]  9.060517 10.279483
```

- Usando a função *deviance* temos,

```
> deviance <- function(lambda, lambda.estimado, amostra) {
+     dev <- 2 * (log.verossimilhanca(lambda.estimado, amostra = amostra) -
+         log.verossimilhanca(lambda, amostra = amostra))
+     return(dev)
+ }
```

- Da mesma forma podemos fazer um gráfico,
- Com base no gráfico podemos encontrar os limites do intervalo,

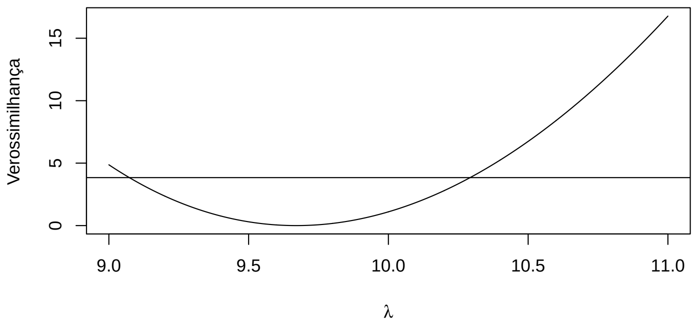

Figura 2: Função *deviance*

```
> saida.dev <- data.frame(grid.theta, desvios)
> ic.deviance = range(saida.dev[which(saida.dev$desvios < qchisq(0.95,
+     df = 1)), ]$grid.theta)
> ic.deviance

[1]  9.073407 10.292329
```

- Em situações mais gerais onde não é simples encontrar os limites do intervalo olhando diretamente para a *deviance* podemos fazer uma aproximação desta por Séries de Taylor, o que resulta em

$$-(\lambda - \hat{\lambda})^2 l''(\hat{\lambda})$$

. Podemos fazer um gráfico da aproximação,

```
> dev.approx <- function(lambda, lambda.estimado, amostra) {
+     dev.app = (lambda - lambda.estimado)^2 * (length(amostra)/lambda.estimado)
+     return(dev.app)
+ }
```

- Obtendo o intervalo baseado na deviance aproximada

```
> saida.dev <- data.frame(grid.theta, desvios.app)
> ic.deviance.app = range(saida.dev[which(saida.dev$desvios.app <
+     qchisq(0.95, df = 1)), ]$grid.theta)
> ic.deviance.app

[1]  9.060606 10.279328
```

- Resumindo os intervalos

```
> ic.assintotico

[1]  9.060517 10.279483

> ic.deviance
```

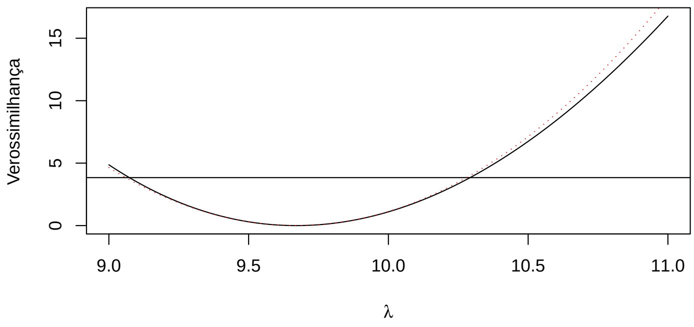

Figura 3: Função *deviance* aproximada sobreposta a deviance exata.

```
[1]  9.073407 10.292329
> ic.deviance.app
[1]  9.060606 10.279328
```