Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Implementação Computacional do Modelo Log Binomial

Mateus Carbone Ananias

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, para obtenção do título de Bacharel em Estatística.

Brasília
2015

Mateus Carbone Ananias

Implementação Computacional do Modelo Log Binomial

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, para obtenção do título de Bacharel em Estatística.

Orientador: Prof. Dr. Bernardo Borba de Andrade

Brasília
2015

Agradecimentos

Primeiramente agradeço de forma especial minha mãe Helenice Carbone, por todos os anos de cuidado e dedicação, da qual me possibilitou estar aqui apesar de todas a dificuldades.

Agradeço também aos amigos que fiz durante todos esse anos, pois foram de fundamental importância na minha formação, em especial ao alunos Rodrigo Ferrari, Geiziane Oliveira.

Por fim agradeço aos professores dos departamentos de Estatística. Em especial ao meu orientador Prof. Dr Bernardo Borba de Andrade, pela ajuda e paciência durante todo este ano, e aos professores George F. von Borries e Joanlise M. de L. Andrade por aceitarem participar da banca examinadora.

Resumo

Este trabalho teve como principal objetivo, criar um pacote em **R** para estimar os parâmetros da regressão log binomial a partir do algoritmo de barreira adaptativa. Na maioria dos pacotes estatísticos que estimam modelos lineares generalizados, é usado o algoritmo de *iterative weighted least squares* (IWLS), porém, este algoritmo tem problemas nas estimações do modelo log binomial, devido a uma restrição do estaço paramétrico. Para resolver este problema foi proposto o algoritmo de barreira adaptativa, que permite a implementação de restrições não lineares, garantindo que as estimações dos parâmetros permaneçam dentro do espaço paramétrico. Então, foi elaborado um pacote que não apenas estimasse os parâmetros, mas que tivesse outras funcionalidades.

Palavras-chave: Regressão log binomial, Algoritmo de barreira adaptativa, Risco relativo.

Sumário

Introdução e Justificativa

Em algumas áreas de estudo, como por exemplo, na área médica ou na indústria há a necessidade de saber se, existe associação entre alguma característica ou variável, e algum tipo de efeito no evento de interesse do qual pesquisador deseja verificar. Uma maneira de avaliar o efeito de entre duas ou mais variáveis são as medidas da razão de chance (**RC**) e do risco relativo (**RR**)

O **RR** e a **RC** são duas medidas de associação que têm como objetivo avaliar relação entre fatores de risco. Essas medidas têm suas próprias peculiaridades como por exemplo, diferem em suas interpretações e também na situação em que devem ser usadas. Em estudos epidemiológicos por exemplo o **RR** é definido como a razão de incidência ou prevalência entre indivíduos expostos pela incidência ou prevalência de indivíduos não-expostos e, geralmente são usados em estudos de coorte ou estudos transversais. Já a **RC** pode ser usado em estudos caso controle, coorte ou transversal, podendo ser definido como a razão entre a chance de ocorrência do evento de interesse em um grupo pela chance de ocorrência do evento de interesse em outro grupo.

A razão de chances é a medida mais usada atualmente quando se trata de modelos lineares com variável resposta binária, pois em geral são mais fáceis de se estimar. O risco relativo é de mais fácil interpretação, mas como a estimação de seus parâmetros exige métodos computacionais mais complexos, geralmente é deixado em segundo plano. Além disso, o **RR** e a **RC** são muito próximas quando a resposta de interesse se trata de um evento raro, levando muitos pesquisadores a interpretações errôneas quando a incidência de sucesso da variável resposta é mais comum.

Ressalta-se que os métodos computacionais atuais são efetivos quando se trata da estimação de parâmetros da regressão logística, sendo esta a que estima as **RC's** das covariáveis com relação a variável resposta, porém esses métodos não são efetivos quando se trata da regressão log-binomial. A regressão log-binomial estima os **RR's** de modo que, com a metodologia atual de estimação de modelos lineares generalizados (**GLM**) as estimativas para as variáveis explicativas podem ficar fora do espaço paramétrico. Para resolver este problema ANDRADE e CARABIN(2011) e ANDRADE(artigo submetido), sugeriram o uso do algoritmo de barreira adaptativa para modelagem da regressão log binomial.

Usando configurações simuladas para estimação da regressão log-binomial apresentadas por Blizzard e Hosmer(2006), foi observado que as taxas de convergência destes modelos não passou de 76%.Então, foi verificado por ANDRADE e CARABIN(2011) que, com o algoritmo proposto foi obtido 100% de convergências com as mesmas configurações.

O presente trabalho teve por objetivo o desenvolvimento de um pacote para o software **R** para a estimação do parâmetros da regressão log binomial e demais funcionalidades associadas ao ajuste do modelo.

Capítulo 1

Revisão Bibliográfica

1.1 Desenvolvimento de pacotes em R

1.1.1 Definição de um pacote

R é um ambiente e linguagem para análises estatísticas e gráficas desenvolvido nos laboratórios Bell por John Chambers e colegas. A linguagem **R** é de uma classe de linguagens chamada de orientada a objeto, de modo que o R pode ser considerado uma implementação diferente da linguagem **S**, no entanto, é possível se afirmar que a maioria dos códigos escritos em **S** funcionam de forma inalterada no **R**.

A inclusão de pacotes é atualmente um dos grandes motivos para o sucesso do **R**, pois em geral são uma forma simples e transparente de expansão e desenvolvimento da plataforma **R**. Como o **R** é um software livre e de código aberto, qualquer pesquisador pode submeter seu pacotes para que outras pessoas possam usar, levando a uma integração rápida para o desenvolvimento da ciência como um todo, e contribuindo com o próprio **R**. Pacotes também podem ser uma forma de manter códigos e funções particulares para uso pessoal ou local de um grupo de pessoas.

Estes pacotes contem uma formatação básica bem definida de documentação em LaTeX, fornecendo uma compreensão de todas as funcionalidades e aplicabilidades de cada pacote. Deste modo podemos dizer que um pacote pode ser uma forma diferente para a transmissão de técnica computacionais e estatísticas.

1.1.2 Estrutura de um pacote

Para o desenvolvimento de um pacote é necessário se criar um diretório no disco rígido e este diretório deverá ter um mesmo nome do pretendido pacote. Essa Pasta deve conter alguns outros arquivos e diretórios, sendo estes, um arquivos DESCRIPTION e subdiretórios, 'R', 'data', 'demo', 'exec', 'inst', 'man', 'src', e 'tests', de modo que alguns não são essenciais. Cada arquivo e subdiretório tem um respectiva função:

- NAMESPACE: Arquivo que descreve as funções exportadas, métodos a outros tipos

de objetos a serem carregados.

- DESCRIPTION: Um arquivo com a descrição do pacote, autor, e condições de licenças em uma estrutura legível para pessoas e computadores.
- man: Um subdiretório contendo a documentação de todas as funções ou base de dados dentro do pacote.
- R: Um subdiretório contendo os códigos e funções em R.
- data: Um subdiretório com base de dados.

  Todos os aquivos acima são essenciais para um bom pacote. os itens abaixo são opcionais.
- src: Um subdiretório contendo códigos em C, Fortran ou C++.
- tests: Um subdiretório contendo testes de validação.
- exec: Outros arquivos executáveis como por exemplo e Java ou Perl
- inst: Um subdiretório contendo qualquer coisa que o autor deseja que seja copiado junto com a instalação do pacote.
- demo: Um subdiretório que contem arquivos para demostração(via função demo()) de funcionalidades do pacote.

  Além dos diretórios citados, existem outras funcionalidade que podem ser adicionadas ao pacote.

1.1.3 Criando e submetendo o pacote

De maneira resumida, para se criar um pacote que seja utilizável, é necessário que este contenha algumas funcionalidades básicas além dos objetivos básicos propostos. Para isso, é necessário que se crie métodos e funções extras, que, deixarão a utilização do pacote seja fácil, prática e padronizada.

Primeiramente criam-se as funções básicas, que vão gerenciar e calcular as operações do pacote. Então é possível se criar uma classe de objetos, de forma que, possam ou não ter funcionalidades únicas. Geralmente essas funções dependem de outras menores para operações externas. Como todas essas funções são importantes, é necessário que sejam programadas de maneira eficiente para redução do custo computacional de operação envolvido.

Cria-se então métodos, que são compostos por uma série de funções ou ações que atuam em diferentes tipos de objetos. Cada objeto no **R** tem uma classe que indica o tipo de informação contida neste objeto. Exemplos seriam as classe `numeric`, `matrix` e `lm`, em que o método `print`, apresentará um resultado diferente em cada classe. Esta é uma

etapa que pode geral algumas dificuldades, pois são muitas funções com defaults próprios interagindo ao mesmo tempo.

Em seguida, edita-se o arquivo NAMESPACE, que contem informação sobre as funções inclusas no pacote, diferenciando métodos e funções. Cria-se também o arquivo DESCRIPTION, contendo dados de propriedade, licença, requerimento de outros pacotes caso seja necessário, entre outras informações básicas.

Após este processo, segue a definição e geração dos manuais, todos em linguem LaTeX, com referências e exemplos para cada uma da funções exportadas para utilização.

Utiliza-se então, o comando `R CMD build` para a obtenção de um pacote comprimido, omitindo os códigos, arquivos de de backup e outros, resultando em um arquivo fácil de ser transferido e instalado em diferentes sistemas. Em seguida executa-se `R CMD check` para checar se o pacote será instalado, se rodará os exemplos, se a documentação esta completa e se executa corretamente as operações.

O pacote então é submetido ao CRAN, que é uma rede de páginas na internet que mantém a distribuição e contribuição de códigos, especialmente pacotes em R.

Para detalhes mais aprofundado da criação de um pacote, consultar Creating R Packages: A Tutorial de LEISCH (2009) e Writing R Extensions (2006).

1.2 Regressão Log-binomial

1.2.1 Modelo

A regressão log-binomial é um modelo linear generalizado com variável resposta dicotômica predita por covariáveis $\mathbf{x_i} = (\mathbf{1}, \mathbf{x_1}, \ldots, \mathbf{x_k}) \in \mathbb{R}^{k+1}$, na qual a variável resposta $Y_i \sim Bin(n_i, \pi_i)$ com $\pi_i(x) = P(y_i = 1|\mathbf{X})$ e com função de ligação igual ao $\log(\pi)$. Deste modo podemos definir o modelo linear como:

$$\log(\pi_i(x)) = \mathbf{x_i}'\boldsymbol{\beta} \tag{1.1}$$

$$\pi_i(x) = e^{\mathbf{x_i}'\boldsymbol{\beta}}$$

1.2.2 Estimação

A estimação dos parâmetros do modelo se dá através do método de máxima verossimilhança. A verossimilhança no modelo log-binomial é dado por:

$$L(\boldsymbol{\beta}; \mathbf{X}) = \prod_{i=1}^{n} P(y_i = 1|\mathbf{x_i})$$

Dado a expressão (1.1) a log-verossimilhança, com $n_i = 1$ é

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \log(\pi_i(x)) + (1 - y_i) \log(1 - \pi_i(x)) \tag{1.2}$$

em que o espaço paramétrico é restringido por $0 \leq \pi_i \leq 1$ e como $\pi_i = e^{\mathbf{x_i}\boldsymbol{\beta}}$ é possível afirmar que $\Theta(\mathbf{x_i'}) = \cap_i^k \{\boldsymbol{\beta} \in \mathbb{R} : \mathbf{x_i'}\boldsymbol{\beta} \leq 0\}$. A partir de (1.2) podemos calcular o vetor gradiente e a matriz hessiana que serão usadas para estimação dos parâmetros. Sendo eles respectivamente:

$$\nabla l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x_i'} \frac{y_i - \pi_i(x)}{1 - \pi_i(x)} \tag{1.3}$$

$$\nabla^2 l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x_i} \mathbf{x_i'} \frac{\pi_i(x)(y_i - 1)}{(1 - \pi_i(x))^2} \tag{1.4}$$

dado (1.4), é possível estimar a matriz de variâncias e covariâncias que é o inverso da informação de Fisher, $I(\hat{\boldsymbol{\beta}})^{-1} = -[\nabla l^2(\hat{\boldsymbol{\beta}}; \mathbf{x_i})]$. Somente se $\hat{\boldsymbol{\beta}} \in int(\Theta)$, caso $\hat{\boldsymbol{\beta}}$ pertença à fronteira do espaço paramétrico, ver seção 1.3.4.

1.2.3 Risco relativo

Com o modelo definido e os parâmetros estimados é possível calcular o risco relativo. Um exemplo com apenas uma covariável também dicotômica pode ser usado para demonstração. O risco relativo pode ser definido para o exemplo como:

$$RR = \frac{\text{indivíduos sob risco dado uma resposta de interesse}}{\text{indivíduos fora de risco dado uma resposta de interesse}}$$

aplicando esse exemplo em 1.1 temos:

$$\log(\pi) = \beta_0 + \mathbf{x_1}\beta_1$$

$$RR = \frac{P(y = 1|\mathbf{x_1} = 1)}{P(y = 1|\mathbf{x_1} = 0)} = \frac{e^{\beta_0 + \beta_1}}{e^{\beta_0}} = e^{\beta_1} \tag{1.5}$$

generalizando o resultado (1.5) para covariáveis continuas e categóricas temos:

$$RR = \frac{P(y = 1|\mathbf{x_1} = a)}{P(y = 1|\mathbf{x_1} = b)} = \frac{e^{\beta_0 + a\beta_1}}{e^{\beta_0 + b\beta_1}} = e^{(a-b)\beta_1}$$

$$RR_i = e^{(a-b)\beta_i} \; \forall \; i = 1, 2, \ldots, k$$

.

Com o **RR** neste contexto, é possível dizer qual o risco de ocorrer uma resposta de interesse em relação a sua interação ao níveis de uma ou mais variáveis explicativas.

1.3 Algoritmo de Barreira Adaptativa

1.3.1 Definição

Dada um função $\varphi : \mathbb{R}^k \longrightarrow \mathbb{R}$ e restrições não lineares $c_i : \mathbb{R}^k \longrightarrow \mathbb{R}, i = 1, \ldots, k$, os métodos de barreiras são desenvolvidos para problemas na forma:

$$\min \varphi(\theta), \text{para } \theta \in \Theta = \{\theta \in \mathbb{R}^k | c(\theta) \leq 0\} \tag{1.6}$$

sendo $c = (c_1, \ldots, c_k)$ um conjunto de inequações. Também é necessário assumir que Θ é um espaço convexo e com interior não vazio. A função de verossimilhança da regressão log-binomial pode ser associada ao método em (1.6), de maneira que, $\varphi(\boldsymbol{\theta}) = -l(\boldsymbol{\beta})$ e $c(\boldsymbol{\theta}) = \mathbf{x_i}'\boldsymbol{\beta}$, pois a log-verossimilhança do modelo é côncava com apena um ponto de máximo, desta forma se multiplicarmos $l(\boldsymbol{\beta})$ por -1 conseguiremos uma função convexa.

1.3.2 Barreira adaptativa

Lange (1994) propôs um melhoramento dos métodos de barreira disponível no R com o `constrOptim`. Este algoritmo relaciona a iteração atual $\boldsymbol{\theta}^{(t)}$ ao problema da falta de restrição na seguinte forma:

$$\min_{\theta} R_\delta(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

$$\min_{\theta} R_\delta(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \varphi(\boldsymbol{\theta}) + \delta \left[\sum_{i=1}^{m} c_i(\boldsymbol{\theta}^{(t)}) \log(-c_i(\boldsymbol{\theta})) + c_i(\boldsymbol{\theta}) \right]$$

sendo δ um parâmetro de barreira multiplicativa maior do que 0. Se minimizarmos $R_\delta(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ para as restrições lineares, então o logaritmo na função de barreira, para o modelo log binomial pode ser definido como

$$\min_{\theta} R_\delta(\boldsymbol{\beta}|\boldsymbol{\beta}^{(t)}) = -l(\boldsymbol{\beta}) + \delta \left[\sum_{i=1}^{n} \mathbf{x_i'}\boldsymbol{\beta}^{(t)} \log(-\mathbf{x_i'}\boldsymbol{\beta}) + \mathbf{x_i'}\boldsymbol{\beta} \right]$$

que força a solução $\boldsymbol{\beta}^{(t+1)}$ a ficar dentro do espaço paramétrico.

1.3.3 Algoritmo

Dada uma função φ e um conjunto de restrições $\Theta = \{\theta \in \mathbb{R}^k | c(\theta) \leq 0\}$, escolha um δ e um chute inicial $\theta^{(1)}$ dentro de θ, um algoritmo de minimização MIN e um critério de parada. Na interação t:

1. Use MIN para resolver (1.6), dada a atual iteração $\theta^{(t)}$. Se a solução está no interior de Θ então calcule $\theta^{(t+1)}$, caso contrário, recomece com outro $\theta^{(1)}$.

2. Se o critério de parada é satisfeito, então $\theta^{(t+1)}$ é solução, caso contrário, $t = t + 1$ e retorne para 1.

1.3.4 Distribuição assintótica

Em geral, quando se garante algumas propriedades de estimadores de máxima verossimilhança, para modelos de regressão com variáveis aleatórias dicotômicas, é possível assumir que:

$$\sqrt{n}(\hat{\beta} - \beta_0) \underset{\longrightarrow}{D} N(0, I^{-1}(\beta)) \ , \ n \longrightarrow \infty$$

Porém quando há pontos na fronteira do espaço paramétrico, os estimadores não terão distribuição normal assintótica. Então, Andrade (artigo submetido), mostra que por multiplicadores de lagrange, é possível obter seguinte distribuição para os parâmetros:

$$\begin{bmatrix} \sqrt{n}(\hat{\beta} - \beta_0) \\ \frac{1}{\sqrt{n}}\hat{\lambda} \end{bmatrix} \underset{\longrightarrow}{D} N\left(0, \begin{bmatrix} \Sigma & 0 \\ 0 & Q \end{bmatrix}\right) \ , \ n \longrightarrow \infty \tag{1.7}$$

Onde $\hat{\lambda}$ é um vetor de multiplicadores de Lagrange, com Σ e Q matrizes

$$\Sigma = I^{-1} - I^{-1}A^{'}\left(AI^{-1}A^{'}\right)^{-1}AI^{-1}$$

$$Q = \left(AI^{-1}A^{'}\right)^{-1}$$

Sendo $A_{q\times(k+1)}$ a matriz com restrições ativas. Esta matriz é formada pelas observações que ficaram no limite do espaço paramétrico, e caso não houvesse restrição, sua probabilidade estimada seria maior do que a unidade, porém, como há restrição sua probabilidade estimada é aproximadamente 1.

A partir destas definições é possível obter intervalos de confiança, teste de hipóteses e erros padrão das estimativas dos parâmetros.

1.3.5 Predição

Comumente definimos a predição como $\hat{y} = \mathbf{x}^{'}\hat{\beta}$ para modelos de regressão linear simples, com $\mathbf{x}$ sendo a matriz das variáveis preditoras. A ideia é similar para modelos lineares generalizados com resposta dicotômica. A diferença é que, neste caso queremos prever a probabilidade de ocorrer um evento dadas algumas variáveis resposta, ou seja, $\tilde{p}(\mathbf{z}) = \exp(\mathbf{z}^{'}\tilde{\beta})$, dado que, $\mathbf{z}$ é a matriz de novas observações.

No caso da RLB, Andrade (artigo submetido), diz que quando há uma nova observação na borda, as probabilidades estimadas podem ser maiores do que 1. Deste modo é proposto a seguinte alternativa:

$$\tilde{\beta} = \text{argmax } l(\beta) \; t.q \; \beta \in \Theta \begin{bmatrix} \mathbf{x} \\ \mathbf{z}' \end{bmatrix}$$

Então para cada conjunto de dados novos, é necessário recalcular os parâmetros, de modo que, quando não há pontos novos na borda do espaço paramétrico, $\tilde{\beta} = \hat{\beta}$ e $\tilde{p} = \hat{p}$.

Nos exemplos da seção 3.2, isso será mostrado.

1.4 IWLS

`IWLS` sigla dada para *iterative weighted least squares*, é um método iterativo para resolver problemas de otimização baseado no método de Newton-Rapshon. Geralmente usado pra estimação da verossimilhança de modelos lineares generalizados, podendo ser definido como

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - [\nabla^2 l(\boldsymbol{\beta}^{(t)}|\mathbf{x_i})]^{-1} \nabla l(\boldsymbol{\beta}^{(t)}|\mathbf{x_i}) \tag{1.8}$$

Substituindo as equações obtidas pro máxima verossimilhança em (1.3) e (1.4) aplicando em (1.8) temos o seguinte processo interativo

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left[\sum_i \mathbf{x_i} \mathbf{x_i}' \frac{\pi_i(x)^{(t)}(y_i - 1)}{(1 - \pi_i(x)^{(t)})^2} \right]^{-1} \sum_i \mathbf{x_i} \frac{y_i - \pi_i(x)^{(t)}}{(1 - \pi_i(x)^{(t)})}$$

Sendo que $\pi_i(x)^{(t)} = e^{\mathbf{x_i}\boldsymbol{\beta}^{(t)}}$, e portanto quando $\pi_i(x) \approx 1$, os denominadores das expressões formadas pelo produto de (1.3) e (1.4) é ≈ 0, gerando problemas computacionais e impedindo a convergência do processo iterativo.

Capítulo 2

Metodologia

A regressão log-binomial é um modelo linear com variável resposta dicotômica explicada por covariáveis quantitativas e qualitativas com função de ligação dado pelo $\log(\pi)$, onde π é a probabilidade de sucesso na variável resposta y_i com distribuição Binomial$(1, \pi)$.

Então, para estimar os parâmetros da regressão log-binomial é possível usar o método de máxima verossimilhança(**MV**), mas para encontrar esses parâmetros é necessário usar um método numérico iterativo. Em geral os pacotes estatísticos como **SAS**, **Stata** e **R** estimam os betas da regressão log-binomial através de variações do algoritmo de mínimos quadrados ponderados iterativamente(**IWLS**), que é uma variação do algoritmo de Newton-Raphson.

Para este trabalho foi produzido um pacote para o software **R**, que por se tratar de código aberto, é um mecanismo de difusão fácil e transparente para usuários do procedimento proposto. Um pacote pode ser considerado como um artigo que ao invés de divulgar ideias, tem por finalidade disponibilizar metodologias estatísticas e computacionais através de seus manuais e códigos. A flexibilidade e a facilidade de distribuir tais pacotes têm sido os principais motivos do sucesso do **R**.

A abordagem da estimação de **MV** proposto é baseado em um algoritmo de maximização restrita que por sua vez, também já existe na maioria dos softwares estatísticos. A grande diferença do algoritmo de barreira adaptativa (Nocedal and Wright, 2006) com o IWLS, é que ele permite uma restrição ou um limite, neste caso mantendo as estimações dentro do espaço paramétrico.

Foi proposto então como principal objetivo, o desenvolvimento de um pacote do **R** que pudesse ser enviado ao CRAN, e que mantenha em parte as funcionalidades como predição, apresentação e efetividade da função `glm`, utilizando programação não linear.

Capítulo 3

Resultados

3.1 Funções e resultados do pacote

Primeiramente é necessário apresentar as funções básicas do pacote e suas saídas mais importantes, para que os exemplos que virão a seguir sejam melhor interpretados. As principais funções são as `lbreg`, `relrisk` e `predict`. Também exitem funções que trabalham internamente, mas que para conferência foram exportadas, estas seriam as funções `hess` que calcula a matriz hessiana, e a função `ll` que calcula a log-verossimilhança. No pacote também há métodos S3 básicos como `summary`, `print` alem de outras.

A função `lbreg` é a mais importante do pacote, pois esta estima os parâmetros da regressão, matriz de variância e covariância, matriz de restrições ativas e outros. O seu funcionamento se assemelha muito a as funções `lm` e `glm` do **R**, porém com menos opções.

A entrada na função fica na seguinte forma:

```
lbreg(formula, data=list(), start, tol, delta, ...)
```

- `start`: chute inicial dos parâmetros, onde o operador deve usar um chute para cada parâmetro que será estimado, sendo esta entrada por default opcional.
- `tol`: tolerância de barreira, que por default é igual a 0.9999.
- `delta`: parâmetro de barreira, por default é 1.
- `data`: base de dados, também opcional, pois é possível entrar direto com os vetores.
- fórmulas padrão do R.

A principais saídas dessa função são:

- `coefficients`: Coeficientes estimados.
- `se`: Erros padrão dos coeficientes.

- `vcov`: Matriz de variância e covariância estimada a partir da expressão 1.7, caso haja pontos na borda.
- `vcov1`: Matriz de variância e covariância estimada sem restrição.
- `Active`: Matriz de barreiras ativas.
- `loglik`: Log-verossimilhança.

A função `relrisk`, trabalha sobre o objeto lbreg calculando o risco relativo e os intervalos de confiança para cada covariável.

Esta função tem a seguinte forma:

```
relrisk(objeto, alpha)
```

- `objeto`: objeto lbreg gerado pelo ajuste do modelo através da função `lbreg`.
- `alpha`: nível de significância do intervalo de confiança. por default alpha é igual a 0.05.

A saída se dá em uma tabela com os risco relativos e intervalos de confiança.

A função `predict` é um método S3 em que se aplica objetos lbreg. Esta é a função com o maior número de especificidades, pois como é necessário levar em consideração as obervações anteriores, para refazer as estimações dos parâmetros.

A função tem as especificações:

```
predict(object, newdata, start,...)
```

- `objeto`: objeto lbreg gerado pelo ajuste do modelo através da função `lbreg`.
- `newdata`: novos dados a serem preditos.
- `start`: chute inicial dos parâmetros, onde o operador deve usar um chute para cada parâmetro que será estimado, sendo esta entrada por default opcional.

A saída do `predict` é uma lista com basicamente as mesmas saídas do `lbreg`, porém haverá também os valores preditos e os novos parâmetros.

3.2 Exemplos

Utilizando agora o pacote e as funções descritas acima, serão feitos 3 exemplos, também apresentados por ANDRADE (artigo submetido), mostrando todas a funcionalidades do pacote proposto. Além disso, como estes exemplos são baseados em bancos de dados reais, com características bastante específicas, é possível exemplificar de maneira bem clara, os problemas gerados pelo **IWLS** na estimação do parâmetros da regressão log-binomial.

É importante a ressalva que, neste trabalho serão comparados apenas o pacote `lbreg` e a função `glm` do **R**, a partir de um ponto de e vista computacional, não levando em consideração as análises das bases de dados citadas.

3.2.1 Exemplo 1

Este exemplo serve de demostração de todas a função básicas na prática. Será usada uma base de dados fictícia utilizada por Andrade (Artigo submetido) com apenas duas variáveis, uma resposta y e uma covariável x.

Abaixo seguem as formas de utilização da função `lbreg` e seus defaults.

```
> m1<-lbreg(y~x+I(x^2),data=PDex02)
> m1

Call:
lbreg.formula(formula = y ~ x + I(x^2), data = PDex02)

Coefficients:
(Intercept)           x      I(x^2)
     0.6171     -0.6732      0.0561
```

Neste caso o chute inicial é dado por uma aproximação de uma regressão de Poisson, de modo que esta sempre converge.

```
> m1<-lbreg(y~x+I(x^2),data=PDex02,start=c(-1,-1,-1),delta=1,tol=.9999)
> m1

Call:
lbreg.formula(formula = y ~ x + I(x^2), data = PDex02, start = c(-1,
    -1, -1), tol = 0.9999, delta = 1)

Coefficients:
(Intercept)           x      I(x^2)
    0.61793    -0.67410     0.05618
```

É possível observar que os parâmetros tiveram uma leve mudança quando o chute inicial é alterado. Por default não é necessário um chute inicial.

Ajustado o modelo, tem-se os métodos `summary`, `predict`, `print` e a função `relrisk`.

```
> summary(m1)

Call:
lbreg.formula(formula = y ~ x + I(x^2), data = PDex02, start = c(-1,
    -1, -1), tol = 0.9999, delta = 1)

            Estimate   StdErr z.value p.value
(Intercept)  0.61793  0.08704     7.1 1.3e-12 ***
x           -0.67410  0.09495    -7.1 1.3e-12 ***
I(x^2)       0.05618  0.00791     7.1 1.3e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> print(m1)
```

```
Call:
lbreg.formula(formula = y ~ x + I(x^2), data = PDex02, start = c(-1,
    -1, -1), tol = 0.9999, delta = 1)

Coefficients:
(Intercept)           x      I(x^2)
    0.61793    -0.67410     0.05618
```

```
> relrisk(m1)

CI limits correspond to level 0.95
                RR  lowCI   upCI
(Intercept) 1.8551 1.5641 2.2001
x           0.5096 0.4231 0.6139
I(x^2)      1.0578 1.0415 1.0743
```

Criando algumas observações para demonstração da função `predict`, temos:

```
> dados<-data.frame(x=c(1,2,3,10,5,4,3,2,1,3,5,4,3,2,1,2,3,4,5,6,6,7,5,3,7))
> predict(m1,dados)$ypred

 [1] 1.0000 0.7024 0.5336 0.7024 0.3898 0.4385 0.5336 0.7024 1.0000 0.5336
[11] 0.3898 0.4385 0.5336 0.7024 1.0000 0.7024 0.5336 0.4385 0.3898 0.3748
[21] 0.3748 0.3898 0.3898 0.5336 0.3898
```

Abaixo podemos ver a diferença nos parâmetros quando necessário prever observações novas. Com os parâmetros originais sendo

```
> coef(m1)

(Intercept)           x      I(x^2)
    0.61793    -0.67410     0.05618
```

e com os parâmetros reestimados para predição

```
> predict(m1,dados)$coef.pred

(Intercept)           x      I(x^2)
    0.43178    -0.47104     0.03925
```

3.2.2 Exemplo 2

O segundo exemplo trata da base de dados Death, disponível na documentação do **SAS** usada por Petersen e Deddens (2010), que contem como resposta os resultados de julgamentos onde os réus foram condenados a pena de morte ou prisão perpétua (1 = pena de morte, 0 = prisão perpétua). Como covariáveis serão usados a raça do réu com 2 níveis, raça da vítima com 2 níveis, uma medida de culpabilidade e algumas medidas de seriedade do crime cometido. A característica mais importante desta base, é que, aproximadamente

34% das observações estão na borda do espaço paramétrico, ou seja, suas probabilidades estimadas são aproximadamente 1.

Primeiramente foi rodado o modelo com todas as covariáveis através `glm` do **R**, e abaixo é possível observar que pelo default de 25 iterações da função, não houve convergência. Porem quando é aumentado o número máximo de iterações, o processo interativo converge com mensagens de avisos.

```
> md2<-glm(death~blackd+whitvic+serious+factor(culp)+serious2,
+ family = binomial(link = "log"),data = Death, start = rep(-1, 9))
> md2$converged

[1] FALSE
```

Agora, rodando o mesmo modelo com a função `lbreg`, há convergência sem problemas.

```
> md<-lbreg(death~blackd+whitvic+serious+culp+serious2,data=Death)
> md$convergence

[1] 0
```

Utilizando a o método `summary` para o objeto lbreg, é possível observar os erros padrão e significância dos parâmetros.

```
> summary(md)

Call:
lbreg.formula(formula = death ~ blackd + whitvic + serious +
    culp + serious2, data = Death)

            Estimate  StdErr z.value p.value
(Intercept)  -2.8212  0.4976   -5.67 1.4e-08 ***
blackd        0.2925  0.1710    1.71   0.087 .
whitvic       0.1044  0.2247    0.46   0.642
serious      -0.0224  0.0774   -0.29   0.772
culp2         1.2693  0.5657    2.24   0.025 *
culp3         2.1836  0.5166    4.23 2.4e-05 ***
culp4         2.4773  0.4407    5.62 1.9e-08 ***
culp5         2.4917  0.4495    5.54 3.0e-08 ***
serious2      0.0321  0.2015    0.16   0.874
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

E estimar os riscos relativos através da função `relrisk`

```
> relrisk(md)

CI limits correspond to level 0.95
                  RR   lowCI    upCI
(Intercept)  0.05953 0.02245  0.1579
```

```
blackd       1.33973 0.95825  1.8731
whitvic      1.11008 0.71463  1.7244
serious      0.97783 0.84023  1.1380
culp2        3.55849 1.17431 10.7832
culp3        8.87825 3.22521 24.4397
culp4       11.90886 5.02054 28.2482
culp5       12.08137 5.00584 29.1578
serious2     1.03260 0.69563  1.5328
```

Por fim, vamos comparar as predições para ambas as funções. Como mostrado por Andrade, quando a predição se da de uma observação dentro do espaço paramétrico, não há alteração nos parâmetros estimados.

Para exemplificar, considere uma base fictícia com as observações que tiveram probabilidades estimadas iguais a 1 e portanto estão na borda da base de dados Death.

```
    death blackd whitvic serious culp serious2
40      1      1       1     7.8    4      3.8
147     1      1       1     9.3    5      4.4
```

Então temos os parâmetros para o **glm**:

```
  (Intercept)        blackd       whitvic       serious factor(culp)2
     -2.87146       0.31834       0.12955      -0.01687       1.26836
factor(culp)3 factor(culp)4 factor(culp)5      serious2
      2.16460       2.47051       2.48272       0.02221
```

e para o **lbreg**:

```
(Intercept)      blackd     whitvic     serious       culp2       culp3
   -2.82024     0.29201     0.10394    -0.02262     1.26859     2.18344
      culp4       culp5    serious2
    2.47701     2.49141     0.03257
```

Quando colocamos uma nova observação na borda do espaço paramétrico, o `glm` estima probabilidades maiores do que 1.

```
> obs<-rbind(obs,c(1,1,1,8,5,4))
> predict(md2,obs,type='response')

    40    147      3
0.9998 1.0000 1.0131
```

Já a predição feita através do algoritmo de barreira adaptativa não tem este problema, pois o coeficientes são recalculados.

```
> predict(md,obs)$coef.pred

(Intercept)      blackd     whitvic     serious       culp2       culp3
  -2.901397    0.309334    0.105794   -0.006663    1.246624    2.118671
      culp4       culp5    serious2
   2.456182    2.453196    0.021594

> predict(md,obs)$ypred

[1] 1 1 1
```

3.2.3 Exemplo 3

O terceiro exemplo vai tratar da base de dados Heart utilizado por Marschner e Gillett (2011), uma base de dados com 16.949 observações, com as seguintes covariáveis:

- age: 3 níveis, menos 65 anos,entre 65 e 75 anos,mais de 75 anos;
- severity: Classificação de seriedade de Killip com 3 níveis (1,2,3);
- delay: Tempo de espera para o tratamento em horas com 3 níveis, menos de 2 horas, entre 2 e 4 horas e mais de 4 horas;
- region: Região do paciente com 3 níveis, países ocidentais, América latina e Europa ocidental;

Mesmo com as estimações da verossimilhança dentro do espaço paramétrico para essa base de dados, o processo iterativos para estimação dos parâmetros não convergem com o `glm`, mesmo quando o número máximo de iterações é aumentado para 1000.

```
> mh2<-glm(Heart~age+severity+region+onset,
+ family = binomial(link = "log"),data = Heart, start = rep(-1, 9),maxit=1000)
> mh2$converged

[1] FALSE
```

Com as funções do pacote proposto, podemos obter os seguintes riscos relativos estimados:

```
> mh<-lbreg(Heart~age+severity+region+onset,data=Heart,start=rep(-1, 9))
> summary(mh)

Call:
lbreg.formula(formula = Heart ~ age + severity + region + onset,
    data = Heart, start = rep(-1, 9))

            Estimate  StdErr z.value p.value
(Intercept)  -4.0269  0.0888  -45.34 < 2e-16 ***
age2          1.1035  0.0894   12.34 < 2e-16 ***
age3          1.9262  0.0929   20.73 < 2e-16 ***
severity2     0.7030  0.0700   10.05 < 2e-16 ***
severity3     1.3771  0.0880   15.65 < 2e-16 ***
region2       0.0754  0.1806    0.42    0.68
region3       0.4828  0.0872    5.53 3.1e-08 ***
onset2        0.0588  0.0691    0.85    0.40
onset3        0.1719  0.0791    2.17    0.03 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> relrisk(mh)
```

```
CI limits correspond to level 0.95
                 RR   lowCI    upCI
(Intercept) 0.01783 0.01498 0.02122
age2        3.01485 2.53013 3.59243
age3        6.86315 5.72059 8.23391
severity2   2.01976 1.76097 2.31658
severity3   3.96323 3.33560 4.70894
region2     1.07830 0.75683 1.53631
region3     1.62053 1.36582 1.92275
onset2      1.06054 0.92614 1.21444
onset3      1.18757 1.01700 1.38674
```

Fica evidente com este exemplo que em alguns casos, mesmo aumentando o número de iterações, não é possível obter um resultado satisfatório. Mostrado que o uso do algoritmo de barreira adaptativa deve ser utilizado.

O pacote será disponibilizado no site http://probest-unb.weebly.com/

Capítulo 4

Considerações finais

É possível dizer que objetivo principal deste trabalho foi alcançado, sendo este a elaboração de um pacote em R que utilizasse programação não linear para estimação dos parâmetros da regressão log-binomial. Um fato importante é que, em comparação com o `glm` do **R**, houve um ganho considerável nas estimativas, principalmente quando se trata de predição.

importante ressaltar que, há certas limitações do pacote `lbreg` em relação ao `glm`, como por exemplo, a falta de uma estrutura que permita trabalhar com pesos, ou a quantidade inferior de métodos ja pré definidos. Ainda neste trabalho não houve tempo hábil para a implementação de medidas de diagnóstico, mostrado que os trabalhos ainda deverão ser continuados, para melhoramento do pacote e para que pessoa ser submetido ao CRAN e divulgado a todos os usuários do **R**.

Referências Bibliográficas

[1] Andrade, B. B.; **Estimation of Log-Binomial Regression via Nonlinear Programming in R** Artigo submetido.

[2] Andrade, B.B.; Carabin. H.; **Estimação de Riscos Relativos via Regressão Log Binomial.** Rev Bras Biom, 2011.

[3] BLIZZARD, L.; HOSMER, D. W. **Parameter estimation and goodness-of-fit in log binomial regression**. Biom. J., Weinheim, v.48, n.1, p.5-22, 2006.

[4] LEISCH, F.;**Creating R Packages: A Tutorial** Department of Statistics, Ludwig-Maximilians-Universitat Munchen, 2009.

[5] Marschner IC, Gillett AC . **Relative risk regression: reliable and exible methods for log-binomial models.** Biostatistics, 13, 179-192, 2011.

[6] Petersen MR, Deddens JA. **Maximum Likelihood Estimation of the Log-Binomial Model**. Communications in Statistics: Theory and Methods, 39, 874-883, 2010.

[7] R Development Core Team.; **Writing R Extensions**, 2006.

[8] WACHOLDER, S.; **Binomial Regression in GLMI: Estimatig Risk Ratios and Risk Differences**.American Jornal Epidemiologi, Oxford, v.123, n.1, p.174-184, 1986.

[9] Wagner B.M; Callegari-Jaques S.M, **Medidas de associação em estudos epidemiológicos: risco relativo e odds ratio**. Jornal de Pediatria ,1998.