## Package ‘sglr’

February 20, 2015

**Type**  Package

**Title**  An R package for power and boundary calculations in pre-licensure vaccine trials using a sequential generalized likelihood ratio test

**Version**  0.7

**Date**  2012-12-07

**Author**  Balasubramanian Narasimhan [aut, cre], Mei-Chiung Shih [aut]

**Maintainer**  Balasubramanian Narasimhan <naras@stat.stanford.edu>

**Description**  Functions for computing power and boundaries for pre-licensure vaccine trials using the Generalized Likelihood Ratio tests proposed by Shih, Lai, Heyse and Chen

**Depends**  R (>= 3.0), ggplot2, shiny

**License**  GPL (>= 2)

**NeedsCompilation**  no

**Repository**  CRAN

**Date/Publication**  2014-05-03 00:54:29

### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>sglr-package</td>
<td>2</td>
</tr>
<tr>
<td>computeBoundary</td>
<td>3</td>
</tr>
<tr>
<td>glrSearch</td>
<td>4</td>
</tr>
<tr>
<td>plotBoundary</td>
<td>6</td>
</tr>
<tr>
<td>sglrWebapp</td>
<td>7</td>
</tr>
</tbody>
</table>

**Index** 9
An R package for power and boundary calculations in pre-licensure vaccine trials using a sequential generalized likelihood ratio test

**Description**
This package is an implementation of the methodology of Shih, Lai, Heyse, and Chen (see reference below) for computing Generalized Likelihood Ratio test boundaries in pre-licensure vaccine studies.

**Details**

<table>
<thead>
<tr>
<th>Package:</th>
<th>sglr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>0.05</td>
</tr>
<tr>
<td>Date:</td>
<td>2010-04-20</td>
</tr>
<tr>
<td>License:</td>
<td>GPL (version 2 or later)</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

The package provides several functions. The function glrSearch computes boundaries for testing a given $p_0$ versus $p_1$ (specified as a two-dimensional vector) given a significance level $\alpha$ and a type II error $\beta$. The function computeBoundary computes the boundary in terms of a more understandable and usable quantity, such as the number of adverse events in a pre-licensure vaccine study for example. It takes as input a set of given boundaries for the GLR statistic. The third function is plotBoundary which also takes the same arguments as computeBoundary and produces a plot. The last two functions can make use of statistics computed previously for the problem, which can be specified as an argument; otherwise, the statistics are computed from scratch.

**Author(s)**
Balasubramanian Narasimhan with input from Tze Lai and Mei-Chiung Shih. Maintainer: Balasubramanian Narasimhan <naras@stat.stanford.edu>

**References**

Please also consult the website [http://med.stanford.edu/biostatistics/ClinicalTrialMethodology/](http://med.stanford.edu/biostatistics/ClinicalTrialMethodology/) for further developments.

**Examples**
```r
library(sglr)
result <- glrSearch(p=c(.5, .75), alpha=0.05, beta=0.10)
# print(result)  # large amounts of output possible!
result[,3]
```
computeBoundary

A function to compute the boundary of the decision region in terms of the number of adverse events (AEs) of interest, such as vaccine AEs.

Description

This function computes the boundary of the decision region in a manner that can be employed in the field, as a table, for example. See section 4.2 of the reference below.

Usage

computeBoundary(b1, b0, p, glrTables = NULL, tol=1e-7)

Arguments

- **b1**: The acceptance boundary value (corresponds to the boundary $b_1$ in the appendix of reference)
- **b0**: The rejection boundary value (corresponds to the boundary $b_0$ in the appendix of reference)
- **p**: The vector of probabilities, $(p_0, p_1)$ with $p_0 < p_1$.
- **glrTables**: A previously computed set of likelihood functions, to speed up computation for the same hypothesis testing problem. Otherwise, it is computed ab initio, resulting in a longer running time.
- **tol**: A numerical tolerance, defaults to 1e-7

Details

This essentially computes the probabilities of hitting the boundaries using a recursion.

Value

- **upper**: The upper boundary that indicates rejection of the null hypothesis
- **lower**: The upper boundary that indicates acceptance of the null hypothesis
- **estimate**: The estimated $\alpha$ and $\beta$ values corresponding to the two boundaries

Author(s)

Balasubramanian Narasimhan

References


Please also consult the website http://med.stanford.edu/biostatistics/ClinicalTrialMethodology/ for further developments.
glrSearch

This function searches through a space of design boundaries (scalar values a and b) to find values that achieve close to specified type I and type II errors for the sequential generalized likelihood ratio test of p0 versus p1 (specified respectively as vector of length 2) in pre-licensure vaccine trials

Description

The search through the space of b1 (corresponds to b1 in paper) and b0 (corresponds to b0 in paper) is greedy initially. Then refinements to the boundary are made by adjusting the boundaries by the step-size. It is entirely possible that the step-size is so small that a maximum number of iterations can be reached. Depending on how close p0 and p1 are the memory usage can grow significantly. The process is computationally intensive being dominated by a recursion deep in the search.

Usage

```r
glrSearch(p, alpha, beta, stepSize = 0.5, tol = 1e-7,
           startB1 = log(1/beta), startB0 = log(1/alpha),
           maxIter = 25, gridIt = FALSE, nGrid = 5,
           verbose = FALSE)
```

Arguments

- `p`: The vector of p0 and p1, with p0 < p1
- `alpha`: A value for type I error \( \alpha \) between 0 and 1 typically 0.05 which is the default value
- `beta`: A value for type II error \( \beta \) between 0 and 1 typically below .2, default 0.10
- `stepSize`: A value to use for moving the boundaries during the search, 0.5 default seems to work.
- `tol`: A value that is used for deciding when to terminate the search. A euclidean metric is used. Default 1e-7.
- `startB1`: A starting value for the futility boundary, default is log of reciprocal type I error
- `startB0`: A starting value for the rejection boundary, default is log of reciprocal type II error
- `maxIter`: A maximum number of iterations to be used for the search. This allows for a bailout if the step size is too small.
gridIt  A logical value indicating if a grid of values should be evaluated once the boundaries are bracketed in the search.

nGrid  The number of grid points to use, default 5

verbose  A logical flag indicating if you want verbose output during search. Useful for situations where the code gets confused.

Details

One should not use this code without a basic understanding of the Shih, Lai, Heyse and Chen paper cited below, particularly the section on the pre-licensure vaccine trials.

As the search can be computationally intensive, the program needs to use some variables internally by reference, particularly large tables that stay constant.

In our experiments, starting off with the default step size has usually worked, but in other cases the step size and the maximum number of iterations may need to be adjusted.

Value

b1  The explored values of the futility boundary \( b_1 \) (corresponds to the boundary \( b_1 \) in the appendix of reference)

b0  The explored values of the rejection boundary \( b_0 \) (corresponds to the boundary \( b_0 \) in the appendix of reference)

estimate  The estimated \( \alpha \) and \( \beta \) values corresponding to the explored boundaries (a 2-column matrix); first column is \( \alpha \), second is \( \beta \)

glrTables  The constant values of the log likelihoods under \( p_0 \), \( p_1 \) and the estimate probability of terminating at that step. The first two, are, in turn, lists of length \( n \) where \( n \) is the maximum number of adverse events that might be needed for the test. The last element is a matrix of 2 columns, specifying the probability of terminating at each value of \( n \)

alphaTable  a matrix (nGrid x nGrid) of \( \alpha \) values corresponding to the combinations of boundaries \( b \) and \( a \) (which are the row and column names of the matrix). This is computed only if gridIt=TRUE

betaTable  a matrix (nGrid x nGrid) of \( \beta \) values corresponding to the combinations of boundaries \( b \) and \( a \) (which are the row and column names of the matrix). This is computed only if gridIt=TRUE

b1Vals  the vector of \( b_1 \) (or equivalently \( b_1 \)) values used in the grid, computed only if gridIt=TRUE

b0Vals  the vector of \( b_0 \) (or equivalently \( b_0 \)) values used in the grid, computed only if gridIt=TRUE

iterations  The number of iterations actually used

Author(s)

Balasubramanian Narasimhan
References


Please also consult the website [http://med.stanford.edu/biostatistics/ClinicalTrialMethodology/](http://med.stanford.edu/biostatistics/ClinicalTrialMethodology/) for further developments.

Examples

```r
library(sglr)
result <- glrSearch(p=c(.5, .75), alpha=0.05, beta=0.10)

result <- glrSearch(p=c(.5, .75), alpha=0.05, beta=0.10, verbose=TRUE)

result <- glrSearch(p=c(.5, .75), alpha=0.05, beta=0.10, gridIt=TRUE)
print(result$alphaTable)
print(result$betaTable)

## takes a while
result <- glrSearch(p=c(.5, 2/3), alpha=0.05, beta=0.10)
print(names(result))

## result <- glrSearch(p=c(.5, 2/3), alpha=0.05, beta=0.10, gridIt=TRUE)
## print(result$alphaTable)
## print(result$betaTable)
```

plotBoundary

A function to plot the boundary of the decision region

Description

This function attempts to plot the boundary of the decision region, but currently falls flat. Will be rewritten.

Usage

```r
plotBoundary(b1, b0, p, glrTables = NULL, tol=1e-7,
legend=FALSE, textXOffset=2, textYSkip=2)
```

Arguments

- **b1**
  - The acceptance boundary value (corresponds to the boundary \( b_1 \) in the appendix of reference)

- **b0**
  - The rejection boundary value (corresponds to the boundary \( b_0 \) in the appendix of reference)

- **p**
  - The vector of probabilities, \( (p_0, p_1) \) with \( p_0 < p_1 \).
A previously computed set of likelihood functions, to speed up computation for the same hypothesis testing problem. This can speed up computations.

The tolerance, default of 1e-7

A flag indicating if a legend is desired or not, default false

Horizontal offset for legend text

Vertical skip for legend text

This essentially computes the recursion and the probabilities of hitting the boundaries and returns a ggplot2 object.

A ggplot2 object

Balasubramanian Narasimhan

See Also

glrSearch

plotBoundary(b1=2.8, b0=3.3, p=c(.5, .75))

sglrWebapp

A function to run a web application to compute boundaries and plotting the decision region

A function to run a shiny web application to compute boundaries and plotting the decision region. Menus are provided for a point-and-click interface

sglrWebapp()

The web application runs on your local machine. Internet Explorer is known to have trouble at times, so it might be best to use Firefox or Chrome as the default browser to use this web application.
Author(s)
Balasubramanian Narasimhan

Examples

```r
## Not run:
sglrWebapp()

## End(Not run)
```
Index

*Topic design
  computeBoundary, 3
  glrSearch, 4
  plotBoundary, 6
  sglr-package, 2

computeBoundary, 3

glrSearch, 4, 4, 7

plotBoundary, 6

sglr (sglr-package), 2
sglr-package, 2
sglrWebapp, 7