Package ‘deconvolveR’

August 30, 2020

Title  Empirical Bayes Estimation Strategies
Version 1.2-1
VignetteBuilder knitr
Suggests cowplot, ggplot2, knitr, rmarkdown
Description Empirical Bayes methods for learning prior distributions from data.
   An unknown prior distribution \( g \) has yielded (unobservable) parameters, each of
   which produces a data point from a parametric exponential family \( f \). The goal
   is to estimate the unknown prior (“\( g \)-modeling”) by deconvolution and Empirical
   Bayes methods. Details and examples are in the paper by Narasimhan and Efron
URL  https://bnaras.github.io/deconvolveR/
BugReports https://github.com/bnaras/deconvolveR/issues
Encoding UTF-8
Depends R (>= 3.0)
License GPL (>= 2)
LazyData true
Imports splines, stats
RoxygenNote 7.1.0
NeedsCompilation no
Author Bradley Efron [aut],
   Balasubramanian Narasimhan [aut, cre]
Maintainer Balasubramanian Narasimhan <naras@stat.Stanford.EDU>
Repository CRAN
Date/Publication 2020-08-30 01:00:26 UTC

R topics documented:
devolveR-package ........................................ 2
bardWordCount ........................................... 2
**deconvolveR-package**  
*R package for Empirical Bayes g-modeling using exponential families.*

**Description**

`deconvolveR` is a package for Empirical Bayes Deconvolution and Estimation. A friendly introduction is provided in the JSS paper reference below and this package includes a vignette containing a number of examples.

**References**


---

**bardWordCount**  
`Shakespeare word counts in the entire canon: 14,376 distinct words appeared exactly once, 4343 words appeared twice etc.`

**Description**

Shakespeare word counts in the entire canon: 14,376 distinct words appeared exactly once, 4343 words appeared twice etc.

**Usage**

`data(bardWordCount)`

**References**

deconv

A function to compute Empirical Bayes estimates using deconvolution

Description

A function to compute Empirical Bayes estimates using deconvolution

Usage

devonv(
  tau,
  X,
  y,
  Q,
  P,
  n = 40,
  family = c("Poisson", "Normal", "Binomial"),
  ignoreZero = TRUE,
  deltaAt = NULL,
  c0 = 1,
  scale = TRUE,
  pDegree = 5,
  aStart = 1,
  ...
)

Arguments

tau  a vector of (implicitly m) discrete support points for θ. For the Poisson and normal families, θ is the mean parameter and for the binomial, it is the probability of success.

X  the vector of sample values: a vector of counts for Poisson, a vector of z-scores for Normal, a 2-d matrix with rows consisting of pairs, (trial size \( n_i \), number of successes \( X_i \)) for Binomial. See details below

y  the multinomial counts. See details below

Q  the Q matrix, implies y and P are supplied as well; see details below

P  the P matrix, implies Q and y are supplied as well; see details below

n  the number of support points for X. Applies only to Poisson and Normal. In the former, implies that support of X is 1 to n or 0 to n-1 depending on the ignoreZero parameter below. In the latter, the range of X is divided into n bins to construct the multinomial sufficient statistic \( y_k \) (number of X in bin K) described in the references below

family  the exponential family, one of c("Poisson", "Normal", "Binomial") with "Poisson", the default
ignoreZero  if the zero values should be ignored (default = TRUE). Applies to Poisson only
and has the effect of adjusting $P$ for the truncation at zero

deltaAt  the theta value where a delta function is desired (default NULL). This applies to
the Normal case only and even then only if it is non-null.
c0  the regularization parameter (default 1)
scale  if the $Q$ matrix should be scaled so that the spline basis has mean 0 and columns
sum of squares to be one, (default TRUE)
pDegree  the degree of the splines to use (default 5). In notation used in the references
below, $p = p\text{Degree} + 1$
aStart  the starting values for the non-linear optimization, default is a vector of 1s
...

Value

a list of 9 items consisting of

nle  the maximum likelihood estimate $\hat{\alpha}$
Q  the m by p matrix $Q$
P  the n by m matrix $P$
S  the ratio of artificial to genuine information per the reference below, where it
was referred to as $R(\alpha)$
cov  the covariance matrix for the mle
cov.g  the covariance matrix for the $g$
stats  an m by 6 or 7 matrix containing columns for $\theta$, $g$, $\tilde{g}$ which is $g$ with thinning
correction applied and named $tg$, std. error of $g$, $G$ (the cdf of $g$), std. error of
$G$, and the bias of $g$

loglik  the negative log-likelihood function for the data taking a $p$-vector argument

statsFunction  a function to compute the statistics returned above

Details

The data $X$ is always required with two exceptions. In the Poisson case, $y$ alone may be specified
and $X$ omitted, in which case the sample space of the observations $\{X\}$ is assumed to be 1, 2, ...
length($y$). The second exception is for experimentation with other exponential families besides
the three implemented here: $y$, $P$ and $Q$ can be specified together.

Note also that in the Poisson case where there is zero truncation, the $stats$ matrix has an additional
column “$tg$” which accounts for the thinning correction induced by the truncation. See vignette for
details.

References

Bradley Efron. Empirical Bayes Deconvolution Estimates. Biometrika 103(1), 1-20, ISSN 0006-

Examples

```r
set.seed(238923) ## for reproducibility
N <- 1000
theta <- rchisq(N, df = 10)
X <- rpois(n = N, lambda = theta)
tau <- seq(1, 32)
result <- deconv(tau = tau, X = X, ignoreZero = FALSE)
print(result$stats)
```

```r
## Twin Towers Example
## See Brad Efron: Bayes, Oracle Bayes and Empirical Bayes
## disjointTheta is provided by deconvolveR package
theta <- disjointTheta; N <- length(disjointTheta)
z <- rnorm(n = N, mean = disjointTheta)
tau <- seq(from = -4, to = 5, by = 0.2)
result <- deconv(tau = tau, X = z, family = "Normal", pDegree = 6)
g <- result$stats[, "g"]
if (require("ggplot2")) {
  ggplot() +
  geom_histogram(mapping = aes(x = disjointTheta, y = ..count.. / sum(..count..) ),
                color = "blue", fill = "red", bins = 40, alpha = 0.5) +
  geom_histogram(mapping = aes(x = z, y = ..count.. / sum(..count..) ),
                color = "brown", bins = 40, alpha = 0.5) +
  geom_line(mapping = aes(x = tau, y = g), color = "black") +
  labs(x = paste(expression(theta), "and x"), y = paste(expression(g(theta)), " and f(x)"))
}
```

disjointTheta

A set of $\Theta$ values that have a bimodal distribution for testing

Description

A set of $\Theta$ values that have a bimodal distribution for testing

Usage

data(disjointTheta)

surg

Intestinal surgery data involving 844 cancer patients. The data consists of pairs $(n_i, s_i)$ where $n_i$ is the number of satellites removed and $s_i$ is the number of satellites found to be malignant.

Description

Intestinal surgery data involving 844 cancer patients. The data consists of pairs $(n_i, s_i)$ where $n_i$ is the number of satellites removed and $s_i$ is the number of satellites found to be malignant.
Usage

data(surg)

References

Index

* data
  bardWordCount, 2
  disjointTheta, 5
  surg, 5

bardWordCount, 2

deconv, 3
deconvolveR-package, 2
disjointTheta, 5
surg, 5