Package ‘ELMSO’

October 12, 2022

Type Package

Title Implementation of the Efficient Large-Scale Online Display Advertising Algorithm

Version 1.0.1

Date 2020-1-17

Maintainer Courtney Paulson <courtneypaulson@suu.edu>

Description An implementation of the algorithm described in "Efficient Large-Scale Internet Media Selection Optimization for Online Display Advertising" by Paulson, Luo, and James (Journal of Marketing Research 2018; see URL below for journal text/citation and <http://faculty.marshall.usc.edu/gareth-james/Research/ELMSO.pdf> for a full-text version of the paper). The algorithm here is designed to allocate budget across a set of online advertising opportunities using a coordinate-descent approach, but it can be used in any resource-allocation problem with a matrix of visitation (in the case of the paper, website page-views) and channels (in the paper, websites). The package contains allocation functions both in the presence of bidding, when allocation is dependent on channel-specific cost curves, and when advertising costs are fixed at each channel.

Depends R (>= 3.4.0)

License GPL-3

URL <https://journals.sagepub.com/doi/abs/10.1509/jmr.15.0307>

Repository CRAN

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Courtney Paulson [aut, cre],
Lan Luo [ctb],
Gareth James [ctb]

Date/Publication 2020-01-18 08:00:02 UTC
R topics documented:

ELMSO ................................................................. 2
ELMSO.fixed ......................................................... 3
reach.ELMSO .......................................................... 4
reach.ELMSO.fixed .................................................... 5

Index 7

ELMSO  Main ELMSO Function

Description
This function allows you to allocate budget to a set of websites based on the cost curve of the websites and a matrix of pageviews for those sites.

Usage
ELMSO(z, CPM = NULL, a = NULL, tau = NULL, step = 0.05,
size = 100, tol = 10^-3, iters = 200)

Arguments

z An n by p matrix of pageviews

CPM A p-dimensional vector of the average CPM values at each website. This is used to calculate the cost curve from a shifted logistic function. You may instead enter values for a p-dimensional "a" vector to define your own shifted logistic cost curve.

a A p-dimensional vector of values controlling the steepness of the shifted logistic cost curve. You may instead enter values for a p-dimensional vector of average CPM values to have the curve calculated for you.

tau A p-dimensional vector of total pageviews (in thousands) for each website. Defaults to the total pageviews in the matrix for each website (i.e., assumes z matrix represents all website pageviews) divided by 1000.

step A value to control the step size of the lambda grid (distance between budget points). Default is 0.05.

size A value to control the number of lambda values tried (number of budget points). Default is 100.

tol A value to control the convergence tolerance of the coordinate descent procedure. Default is 10^-3.

iters A value to control the number of iterations until algorithm should exit if convergence tolerance is not reached. Default is 200.
Value

bid: A matrix of bid values by website at each budget
spend: a matrix of total spend by website at each budget
budget: a vector of budget values
lambda: a vector of lambda values
a: a vector of a values (used to calculate shifted logistic curves and reach in reach.ELMSO function)

References


Examples

```r
z=matrix(round(abs(rnorm(5000,0,0.7))),1000,5)
CPM.avg=c(3,4,5,6,7)
tau.values=rep(1000,5) #Note tau here is in thousands of pageviews
allocation=ELMSO(z=z,CPM=CPM.avg,tau=tau.values)
allocation$bid
allocation$spend
allocation$budget
allocation$lambda
allocation$a
```

**ELMSO.fixed  Fixed ELMSO Function (fixed advertising costs, no cost curve)**

Description

This function allows you to allocate budget to a set of websites when cost is fixed at each website based on a matrix of pageviews for those sites.

Usage

```r
ELMSO.fixed(z, CPM, tau = NULL, step = 0.05, size = 100,
             tol = 10^-3, iters = 200)
```

Arguments

- **z**: An n by p matrix of pageviews
- **CPM**: A p-dimensional vector of the (fixed) CPM values at each website
- **tau**: A p-dimensional vector of total pageviews (in thousands) for each website. Defaults to the total pageviews in the matrix for each website (i.e., assumes z matrix represents all website pageviews) divided by 1000.
reach.ELMSO

Description

This function allows you to calculate reach achieved at a given budget value from the ELMSO output.

Usage

reach.ELMSO(bid, a, z)
Arguments

- **bid**: A p-dimensional vector of the bidded CPM at each website for a particular budget value.
- **a**: A p-dimensional vector of steepness values for the cost curves associated with each website.
- **z**: An n by p matrix of pageviews.

Value

A value between 0 and 1 specifying the reach achieved with the given budget allocation.

References


Examples

```r
z = matrix(round(abs(rnorm(5000, 0, 0.7))), 1000, 5)
CPM.avg = c(3, 4, 5, 6, 7)
tau.values = rep(100, 5) # Note tau here is in thousands of pageviews
allocation = ELMSO(z = z, CPM = CPM avg, tau = tau.values)
reach.ELMSO(allocation$bid[, 101], allocation$a, z)
```

reach.ELMSO fixed

Calculating Reach from Fixed ELMSO Function

Description

This function allows you to calculate reach achieved at a given budget value from the fixed ELMSO output.

Usage

```r
reach.ELMSO.fixed(CPM, w, z, tau = NULL)
```

Arguments

- **CPM**: A p-dimensional vector of the fixed CPM at each website for a particular budget value.
- **w**: A p-dimensional vector of amount spent at each website.
- **z**: An n by p matrix of pageviews.
- **tau**: A p-dimensional vector of total pageviews (in thousands) for each website. Defaults to the total pageviews in the matrix for each website (i.e., assumes z matrix represents all website pageviews) divided by 1000.
Value

A value between 0 and 1 specifying the reach achieved with the given budget allocation.

References


Examples

```r
z=matrix(round(abs(rnorm(5000,0,0.7))),1000,5)
CPM.fixed=c(3,4,5,6,7)
tau.values=rep(100,5) #Note tau here is in thousands of pageviews

allocation=ELMS0.fixed(z=z,CPM=CPM.fixed,tau=tau.values)
reach.ELMS0.fixed(CPM.fixed,allocation$spend[,101],z,tau.values)
```
Index

ELMSO, 2
ELMSO.fixed, 3
reach.ELMSO, 4
reach.ELMSO.fixed, 5