Package 'HPdgraph'

January 20, 2015

Type Package

Title Distributed algorithms for graph analytics

Version 1.0.0

Date 2015-01-16

Author HP Vertica Analytics Team

Maintainer HP Vertica Analytics Team <distributedRTeam@external.groups.hp.com>

Depends R (>= 3.0.0), distributedR, MatrixHelper

Description

Distributed algorithms for graph analysis. Written using HP Vertica Distributed R package.

License GPL (>= 2) | file LICENSE

R topics documented:

HPdgraph-package 1
hpdpagerank 2
hpdwhich.max 3

Index **5**

`HPdgraph-package` *Distributed algorithms for graph analytics*

Description

HPdgraph provides distributed algorithms for graph analytics. It is written based on the infrastructure created in HP Labs for distributed computing in R.

Details

Package:	HPdgraph
Type:	Package
Version:	1.0.0
Date:	2015-01-16

Main Functions:

- hpdpagerank: compute pagerank of a graph in a distributed fashion.
- hpdwhich.max: returns the index of the maximum value stored in a darray.

Author(s)

HP Vertica Analytics Team <distributedRTeam@external.groups.hp.com>

References

1. Using R for Iterative and Incremental Processing. Shivaram Venkataraman, Indrajit Roy, Alvin AuYoung, Rob Schreiber. HotCloud 2012, Boston, USA.

---

hpdpagerank *Distributed PageRank*

Description

hpdpagerank function is a distributed implementation of the PageRank algorithm.

Usage

```
hpdpagerank(dgraph, niter = 1000, eps = 0.001, damping=0.85,
            personalized=NULL, weights=NULL, trace=FALSE,
            na_action = c("pass","exclude","fail"))
```

Arguments

dgraph	a darray (dense or sparse) that contains the adjacency matrix of the graph. A sparse darray is strongly recommended for memory efficiency. The darray should be column-wise partitioned.
niter	maximum number of iterations
eps	the calculation is considered complete if the difference of PageRank values between iterations change less than this value for every vertex.
damping	the damping factor
personalized	optional personalization vector (of type darray). When NULL, a constant value of 1/N will be used where N is the number of vertices. This darray should be dense and have a single row. The number of its columns should be equal to the number of vertices. Number of partitions should be the same as dgraph.
weights	optional edge weights (of type darray). When NULL, a constant value of 1 will be used. The dimensions, sparsity, and partitioning of this darray should be the same as dgraph.
trace	when TRUE, intermediate steps of the progress are displayed.
na_action	indicates what should happen when dgraph contains missing values. Values of NA, NaN, and Inf in the adjacency matrix are treated as missing values. Three options for this argument are 'pass', 'exclude', and 'fail'. The default value is 'pass' which means missing values will not be checked. When 'exclude' is selected, any edge with missing value will be replaced with zero. When 'fail' is selected, the function will stop in case of any missing value in the input adjacency matrix.

Value

hpdpagerank returns a darray which contains the PageRank vector.

Author(s)

HP Vertica Analytics Team

References

Sergey Brin and Larry Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Proceedings of the 7th World-Wide Web Conference, Brisbane, Australia, April 1998.

http://www-db.stanford.edu/~backrub/google.html

Examples

```
## Not run:

   library(HPdgraph)
   distributedR_start()

   graph <- matrix(0, 6,6)
   graph[2,1] <- 1L;graph[2,3] <- 1L;graph[3,1] <- 1L;graph[3,2] <- 1L;
   graph[3,4] <- 1L;graph[4,5] <- 1L;graph[4,6] <- 1L;graph[5,4] <- 1L;
   graph[5,6] <- 1L;graph[6,4] <- 1L

   dgraph <- as.darray(graph, c(6,3))
   pr <- hpdpagerank(dgraph)

## End(Not run)
```

---

hpdwhich.max *Distributed which.max*

Description

hpdwhich.max function is a distributed version of which.max function for a 1D-array which has darray as its input argument.

Usage

```
hpdwhich.max(PR, trace=FALSE)
```

Arguments

PR	a darray (dense or sparse). It must have only a single row.
trace	when this argument is TRUE, intermediate steps of the progress are displayed.

Details

This function finds and returns the index of the maximum value stored in a darray. The darray is assumed to have a single row which is similar to the pagerank vector returned by hpdpagerank. Therefore, it is suitable for finding the index of the page with the highest rank in the pagerank vector produced by hpdpagerank.

Value

it returns the index of the maximum value stored in a darray.

Author(s)

HP Vertica Analytics Team

Examples

```
 ## Not run:

    library(HPdgraph)
    distributedR_start()

    graph <- matrix(0, 6,6)
    graph[2,1] <- 1L;graph[2,3] <- 1L;graph[3,1] <- 1L;graph[3,2] <- 1L;
    graph[3,4] <- 1L;graph[4,5] <- 1L;graph[4,6] <- 1L;graph[5,4] <- 1L;
    graph[5,6] <- 1L;graph[6,4] <- 1L

    dgraph <- as.darray(graph, c(6,3))
    pr <- hpdpagerank(dgraph)
    hpdwhich.max(pr)

## End(Not run)
```

Index

∗Topic **Big Data Analytics**
HPdgraph-package, 1
hpdpagerank, 2
hpdwhich.max, 3
∗Topic **Distributed Graph Analytics**
HPdgraph-package, 1
∗Topic **Distributed R**
HPdgraph-package, 1
∗Topic **distributed R**
hpdpagerank, 2
hpdwhich.max, 3
∗Topic **distributed pagerank**
hpdpagerank, 2
hpdwhich.max, 3

HPdgraph *(HPdgraph-package)*, 1
HPdgraph-package, 1
hpdpagerank, 2
hpdwhich.max, 3