Package 'HPdgraph'

March 25, 2014

Type Package

Title Distributed algorithms for graph analytics

Version 0.5.0

Date 2014-01-22

Author Arash Fard

Maintainer Arash Fard <afard@vertica.com>

Depends R (>= 3.0.1), distributedR

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

License GPL (>= 2)

R topics documented:

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

Description

HPdgraph provides a few 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:	0.4.0
Date:	2014-01-22

Main Functions:

- hpdpagerank: It is a distributed function for computing pagerank vector of a graph.
- hpdwhich.max: It finds and returns the index of the maximum value stored in a darray.

Author(s)

Arash Fard <afard@vertica.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 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) which contains the adjacency matrix of the graph. A sparse darray is highly recommended for the sake of efficiency. The darray should be column-wise partitioned. It should be noticed that values of this darray will be altered after running hpdpagerank function.
`niter`	the maximum number of iterations
`eps`	the calculation is considered as 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 it is NULL, a constant value of 1/N will be used where N is the number of vertices. This darray should have a single row and the number of its columns should be equal to the number of vertices. Its number of partitions should be the same as dgraph.
`weights`	Optional edge weights (of type darray). When it is NULL, a constant value of 1 will be used. The dimensions, sparsity, and partitioning of this darray should be the same as dgrapg.
`trace`	when this argument is TRUE, intermediate steps of the progress are displayed.
`na_action`	it indicates what should happen when the dgraph contains missed values. Values of NA, NaN, and Inf in the adjacency matrix are treated as missed values. There are three options for this argument 'pass', 'exclude', and 'fail'. The default value is 'pass' which means the missed value will not be checked. When 'exclude' is selected, any edge with missed value will be replaced with zero. When 'fail' is selected, the function will stop in the case of any missed value in the input adjacency matrix.

Value

hpdpagerank returns a darray which contains the PageRank vector.

Author(s)

Arash Fard <afard@vertica.com>

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] <- 1; graph[2,3] <- 1; graph[3,1] <- 1; graph[3,2] <- 1;
   graph[3,4] <- 1; graph[4,5] <- 1; graph[4,6] <- 1; graph[5,4] <- 1;
   graph[5,6] <- 1; graph[6,4] <- 1

   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)

Arash Fard <afard@vertica.com>

Examples

```
 ## Not run:

    library(HPdgraph)
    distributedR_start()

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

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

## End(Not run)
```

Index