SuperHub Documentation

Release

Javier Bejar

febrero 20, 2014

CONTENTS

Contents:

CHAPTER

ONE

CONSTANTS MODULE

1.1 Constants

Description SuperHub constants,

The coordinates of the region of interest and the path to the data files And the information of the mongo database

Authors bejar

Version 1.0

CHAPTER

TWO

DB MODULE

2.1 DB

Description SuperHub data functions

Exports data from database to csv file

Loads data from csv file

Performs different processings to the data matrix

Authors bejar

Version 1.0

SuperHub.DB.**getApplicationData**(*application*)
Get the data events from the database and saves it in a csv file

Param application

Param cpath

Param square

SuperHub.DB.**getApplicationData2**()

SuperHub.DB.**getApplicationDataOne**(*application*)

Param application:

SuperHub.DB.**getLApplicationData**(*lapplication*)
Retrieves data from a lists of Social applications Saves an individual file for each application and a file with all the data

Parameters **lapplication** –

SuperHub.DB.**saveDataResult**(*data*, *fname*)
Save data results in a file

Param data:

Param fname:

SuperHub.DB.**transferApplicationData**(*application*)
Trasfers data from

Param application:

CHAPTER

THREE

DATA MODULE

3.1 Data

Description SuperHub Data class

Performs different processings to the data matrix

Authors bejar

Version 1.0

File: Data

Created on 18/02/2014 10:09

@author: bejar

class SuperHub.Data.**Data**(*path*, *application*)

Class for a superhub dataset

dataset = numpy array application = Name of the data file cpath = Path of the data file mxhh = maximum position of the heavy hitters list mnhh = minimum position of the heavy hitters list lhh = list of users ordered by the number of elements in the dataset

application = None

compute_heavy_hitters(*mxhh*, *mnhh*)

Computes the list of the number of events and returns a list with the users between the positions mxhh and mnhh in the descendent order

If the list heavy hitters have already been computed it is reused

Param data:

Param mxhh:

Param mnhh:

Returns list with the list of users

contingency(*scale*, *distrib=True*)

Generates an scale x scale accumulated plot of the events

Param data:

Param scale:

Param distrib:

daily_table()
Computes the accumulated events by day for the data table

Returns A daily table

dataset = None

datasethh = None

get_dataset()
Returns the numpy array that represents the dataset @return:

hourly_table()
Computes the accumulated events by hour for the data table

Returns An hourly table

lhh = None

mnhh = None

monthly_table()
Computes the accumulated events by month

@return: A montly rable

mxhh = None

read_data()
Loads the data from the csv file

Param application:

Returns

select_data_users(*users*)
Selects only the events from the list of users Returns a new object with the selected users

Param users: List of users to select

Returns

select_heavy_hitters(*mxhh*, *mnhh*)
Deletes all the events that are not from the heavy hitters Returns a new data object only with the heavy hitters

@param mxhh: @param mnhh: @return: A list of the most active users in the indicated range

wpath = None

CHAPTER

FOUR

DESCRIPTIVE MODULE

4.1 Descriptive

Description SuperHub Descriptive data functions

Functions for computing descriptive statistics from the dataset

For now mainly histograms

Authors bejar

Version 1.0

File: Descriptive

Created on 20/02/2014 15:23

@author: bejar

SuperHub.Descriptive.**data_histograms**(*application*, *lhh=None*)

Generate histograms for different characteristics of the data Outputs the data used to generate the histograms

- Number of daily events
- Number of days of users
- Accumulated events per hour
- Accumulated ecents per weekday

Param application:

Param lhh:

SuperHub.Descriptive.**plot_accumulated_events**(*data*, *distrib=True*, *scale=100*)

Plots the accumulated geographical events in the selected area to the specified scale

Param application: name of the data file

Param distrib: whether the PDF or the absolute numbers are plotted

Param scale: scale of the discretization

SuperHub.Descriptive.**user_events_histogram**(*data*, *scale=100*, *timeres=4*)

Histogram of the number of places-time a user has been

Param scale:

Param application:

Param mxhh:

Param mnhh:

CHAPTER

FIVE

PLOTS MODULE

5.1 Plot

Description Different plots of the data

Authors bejar

Version 1.0

SuperHub.Plots.**daily_histogram**(*data*)
Plot of events accumulated by week day

Param application:

Param mxhh:

Param mnhh:

SuperHub.Plots.**hourly_histogram**(*data*)
Plots of events accumulated by hours

SuperHub.Plots.**montly_histogram**(*data*)
Plots the events accumulated by month

@param application: @param mxhh: @param mnhh: @return:

SuperHub.Plots.**plotHisto**(*data*, *bins*)
Plots a histogram

Param data:

Param bins:

SuperHub.Plots.**saveHisto**(*data*, *bins*, *fname*)
Saves a histogram

Param data:

Param bins:

Param fname:

SuperHub.Plots.**savePlot**(*axis*, *data*, *fname*)
Saves a plot of the data using the values of axis

Param data:

Param num:

Param fname:

CHAPTER

SIX

ROUTES MODULE

6.1 Routes

Description Routes

Routines that compute routes

Authors bejar

Version 1.0

File: Routes

Created on 20/02/2014 15:17

@author: bejar

SuperHub.Routes.**transaction_routes**(*data*, *nfile*, *scale=100*, *supp=30*, *timeres=4.0*, *colapsed=False*)

Generates a diagram of the routes obtained by the frequent itemsets fp-growth algorithm

Param dataclean:

Param application:

Param mxhh:

Param mnhh:

Param scale:

Param supp:

Param timeres:

SuperHub.Routes.**transaction_routes_many**(*data*, *lhh=None*, *lscale=None*, *supp=30*, *ltimeres=None*, *colapsed=False*)

Computes the diagrams of frequent routes for a list of parameters

Param application:

Param lhh:

Param lscale:

Param supp:

Param ltimeres:

CHAPTER

SEVEN

TRANSACTIONS MODULE

7.1 Transactions

Description Transactions,

Class for transactions processing

Authors bejar

Version 1.0

Created on 18/02/2014 10:59

@author: bejar

class SuperHub.Transactions.**DailyDiscretizedTransactions**(*data*, *scale=100*, *timeres=4.0*)

Bases: SuperHub.Transactions.DailyTransactions

Class for the daily discretized transactions

class SuperHub.Transactions.**DailyTransactions**(*data*)

Bases: SuperHub.Transactions.Transactions

Class for the daily transactions

colapse()

Colapses the transactions of a user on a set with all the different items in the transactions (basically where has been and when (considering the discretization used) during the period of time covered by the transactions

Param trans: Dictionary of user/time transactions

Returns Dictionary of daily transactions

colapse_count()

Colapsed the transactions of a user on a dictionary with all the different items in the transctions, counting how many times the user has been at that time at that place (considering the discretization used) @return:

save(*rfile*)

Saves the daily transactions in a file

Param nfile:

Param application:

Param mxhh:

Param mnhh:

Param scale:

serialize()
Transforms the transactions from dictionaries to lists

Param trans:

Returns

users_daily_length()
Computes the list of lengths of the daily transactions for all users

users_prevalence()
Computes the number of daily transactions for all users

class SuperHub.Transactions.**Transactions**(*data*)
Class for the user transactions

application = None

usertrans = None

wpath = None

CHAPTER

EIGHT

UTIL MODULE

8.1 Util

Description Util

Different Auxiliary functions used for different purposes

Authors bejar

Version 1.0

File: Util

Created on 20/02/2014 14:12

@author: bejar

SuperHub.Util.**diff_items**(*seq*)

Number of different geo point in a sequence

Param seq:

Returns

SuperHub.Util.**item_key_sort**(*v*)

auxiliary function for sorting geo-time events

Param v:

Returns

CHAPTER

NINE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

INDEX

T

U

W