SuperHub Documentation

Release 0.1

Javier Bejar

febrero 26, 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.`**`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.`**`transferApplicationData`**`(`*application*`)`

Trasfers data from

Param application:

CHAPTER

THREE

STDATA MODULE

3.1 STData

Description SuperHub STData class

Representation for Spatio Temporal data, basically latitude, longitude and time events with the user that generated the event

Performs different processings to the data matrix

Authors bejar

Version 1.0

File Data

Created on 18/02/2014 10:09

class `SuperHub.STData.`**`STData`**(*path*, *application*)

Class for a superhub dataset:

Parameters

- **path** – Sets the path of the file
- **application** – Sets the application of the dataset

`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

Parameters

- **mxhh** (*int*) – initial position of the heavy hitters list
- **mnhh** (*int*) – final position of the heavy hitters list

Returns list with the list of users ordered (desc) by number of events

Return type list

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

Generates an scale x scale accumulated plot of the events

Parameters

- **scale** (*int*) – Scale of the spatial discretization
- **distrib** (*bool*) – If returns the frequency or the accumulated events

daily_table()

Computes the accumulated events by day for the data table

Returns A list with the accumulated number of events for each day of the week

get_dataset()

Returns the numpy array that represents the dataset

Returns numpy array with the data

hourly_table()

Computes the accumulated events by hour for the data table

Returns A list with the accumulated number of events for each hour of the day

monthly_table()

Computes the accumulated events by month

Returns A list with the accumulated number of events for each mont of the year

read_data()

Loads the data from the csv file

select_data_users(*users*)

Selects only the events from the list of users

Parameters **users** (*list*) – List of users to select

Returns Returns a new object with the selected users

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

Parameters

- **mxhh** (*int*) – initial position of the heavy hitters list
- **mnhh** (*int*) – final position of the heavy hitters list

Retuns A list of the most active users in the indicated range

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

`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 (prevalence)
- Accumulated events per hour
- Accumulated events per weekday
- Accumulated events per month

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 data: STData

Param scale: Discretization scale

Param timeres: Time resolution in number of segments from the 24h period

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

r

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

class SuperHub.Transactions.**DailyDiscretizedTransactions**(*data*, *scale=100*, *timeres=4*)

Bases: SuperHub.Transactions.DailyTransactions

Class for the daily discretized transactions

Parameters

- **data** – STData
- **scale** – Space distretization
- **timeres** – Time distretization

generate_data_matrix(*minloc=20*, *mode='af'*)

Generates a sparse data matrix from the transactions

Parameters

- **minloc** (*int*) – Minimun number of locations for a user
- **mode** (*string*) –
  - af = location absolute frequency (total number of times)
  - nf = location normalized frequency for the user (divided by all user locations)
  - bin = presence/non presence of the location

  if the mode includes 'idf' the td x idf value is computed

Returns csc sparse numpy array representing the user locations

Return type csc sparse matrix

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

Returns Dictionary of daily transactions

`colapse_count`()

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

Returns A list with the count for all users of the times he has been in a place

`save`(*rfile*)

Saves the daily transactions in a file

Parameters **rfile** – File for the output. The function closes the file

`serialize`()

Transforms the transactions from dictionaries to lists

Returns Returns a list representation of the transactions

`users_daily_length`()

Computes the list of lengths of the daily transactions for all users

Returns A list with the count of events of each the users for each day

`users_prevalence`()

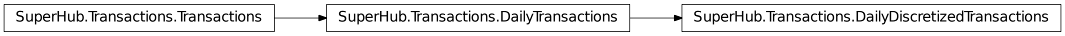
Computes the number of daily transactions for all users

Used to compute user prevalence histograms

Returns list with the count of tractactions for each user

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

Class for the user transactions :param data: STData


SuperHub.Transactions.Transactions
SuperHub.Transactions.DailyTransactions
SuperHub.Transactions.DailyDiscretizedTransactions

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

`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

`SuperHub.Util.`**`item_to_column`**`(`*item*, *scale*`)`

Transforms an item to a column nuber given the scale of the discretization an item is a string with the format posx#posy#time

@param item: @param scale: @return:

CHAPTER

NINE

INDICES AND TABLES

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

PYTHON MODULE INDEX

INDEX