Visualization of Trending Topics' Distribution of Tweets in London and DBSCAN Cluster

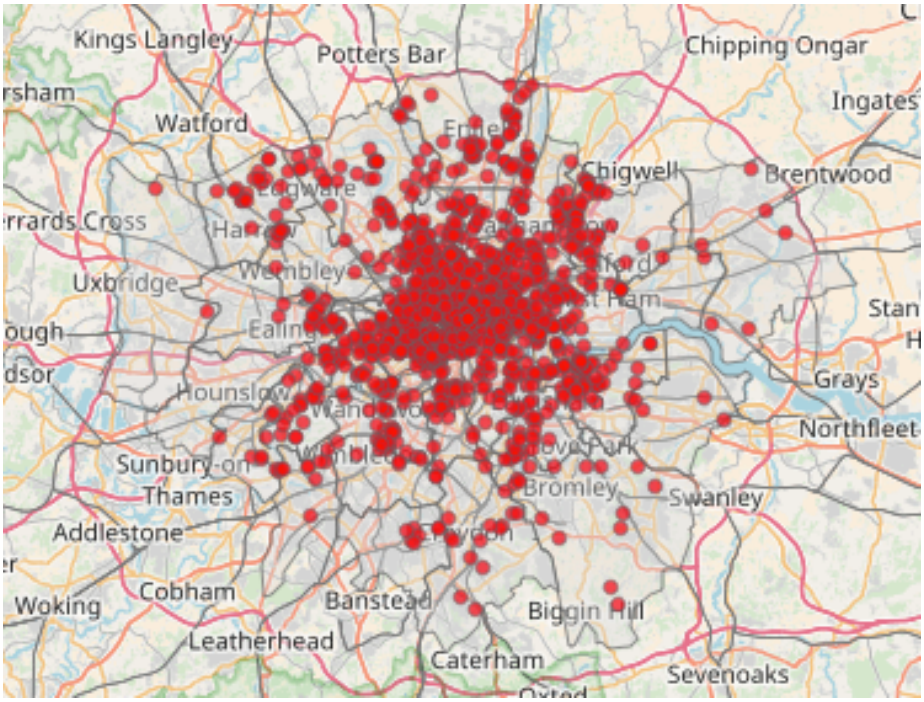


Figure1: Tweets about "new year fireworks of eve" distribute in London from Jan 1st To 2nd

Word Count - 1988
Main document -1348
Appendix -640

Summary

The twitter tool in the report is a tool aiming to visualize the recent distribution of trending topics in London. Interactive maps from “tmap” package is used to show the distribution in London or certain borough. The Ripley’s K is used to identity in what scale the distribution appears random or cluster and to be a reference setting epsilon of DBSCAN that represent the cluster of this distribution. The “Open Street Map” and “ggplot2” packages are used to show the results. The analysis is also applied in certain borough in London, too.

Introduction

Over the past ten years, the increase of usage of online social media(OSM) is unprecedented, which has played a pivotal role in society and economy (Arora et.al 2014). The universe usage is building a network including almost everything like sports, politics, philosophy, entertainment, shopping, transport and literature. In return, the big data will make people inseparable from it. Especially after the location information (geographical information) of OSM has been captured and analyzed universally, the network plays an unprecedentedly profound influence. For example, WECHAT that is an extremely popular OSM in China, is now not only serve public as a social media, but also a bank, supermarket, insurance company, taxi company, ordering software and so on, they usually say they want to make a closed-loop ecological environment for human （Meeker 2015）.

Significant events like football match, election, fireworks of new year’s fireworks are reflected on OSM, which is an interesting research source for many fields like public transport, sociology, GIS and so on. The tool aims to provide an approach to analyze timely distribution of trending topics in London and its boroughs for especially those researchers who do not hold specified GIS and code skills. Due to the limitation of data source that has to

be got from API of different social media companies that do not always allow public freely get too much data form their dataset, the tool specifies on analysis of data from Twitter.

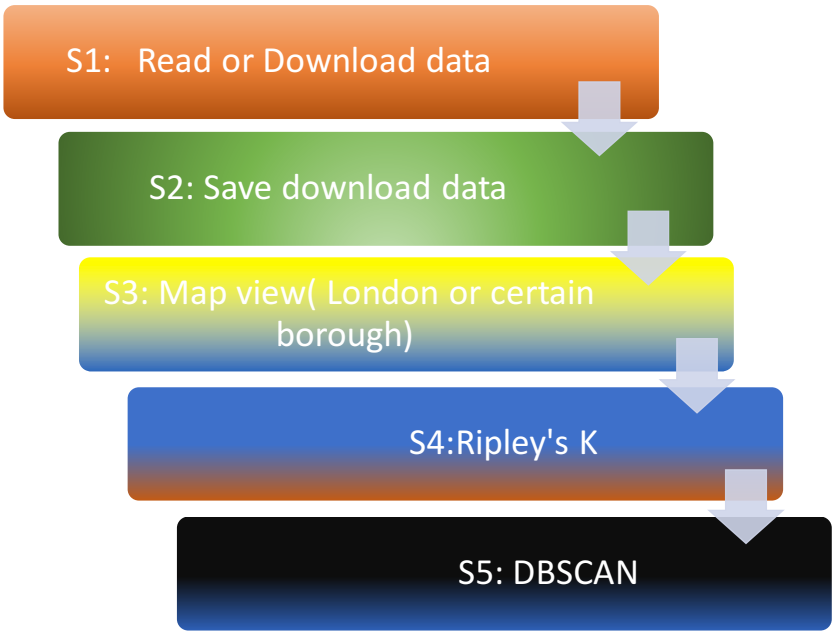
Input

The tool includes two kinds of input, the trending words the users want to analyze, and the preexisting trending topics saved on the file.

If the user input a new a new word, the tool can download data from API of twitter automatically, but the process will take a few minutes and there are a few limitations which will be explained in appendix1. If the user input preexisting words the data can be read from files for following analysis. No matter which one the users choose, the raw data is a table (data frame) including the text of chosen words and information of latitude and longitude, then the data can be converted to format of simple features (sf) and sp automatically.

Process and Methods

Figure2: The process of the tool.



Step1 and Step2

The process of the tool is shown in fig2. In the report, preexisting data about the new year's fireworks form Jan 1st to 4th 2018 is used as an example to show the process. Step1 is getting data, the input for users are the new trending words or preexisting words, and after the input the tool will get "sf" and "sp" including recent tweets about trending words till the time the user uses it. The tool just allows users to input one word to download data, but usually, the trending event includes more than one trending word (Thapa 2016), then the user can repeat Step1 and Step2 using different trending words, because Step2 is to save the download data as csv, so the user just need to stack the csv tables vertically then can get the data representing trending events.

Step3: Map View

Due to the limitation of the API for individual, the users just can get limited data with location information and usually will take a few minutes of longer, the data is just a very small sample of all tweets. So, the tool uses interactive map from "tmap" package to show the distribution of trending words. The size of dots is set bigger than common dots we use normally to show the density of words due to the shortage of sample.

Step4: Ripley's K

Ripley's K is used to show the distribution of the words and whether they cluster or not for any circle radius r. Meanwhile, the plot of Ripley's K is used to be a reference setting epsilon of DBSCAN. If the line of K is above the estimated line, the data appear to be clustered at this r radius, and if the line of K is below the estimated line, the distribution are dispersed.

Step5: DBSCAN

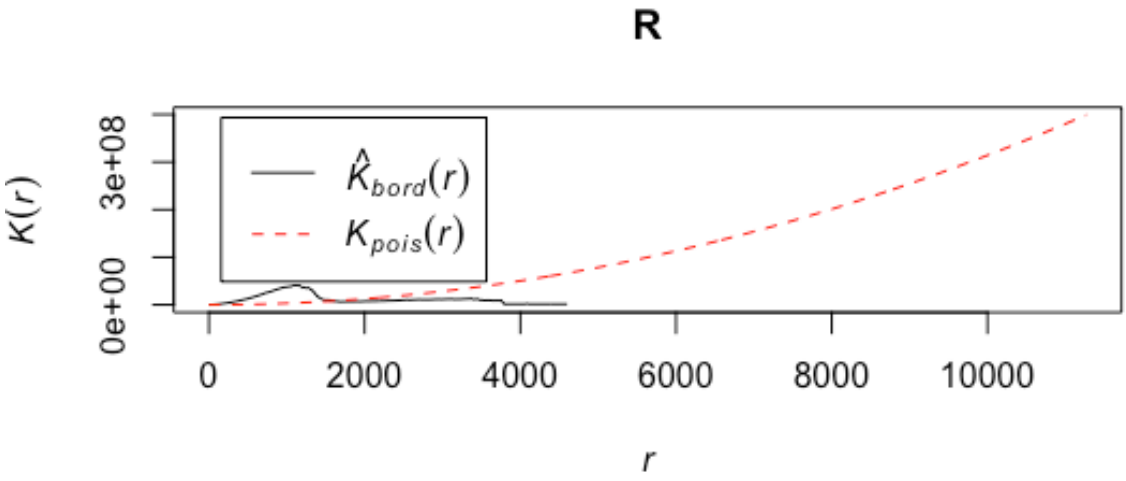
DBSCAN is used to detect clusters of words based on their density, the output can represent the cluster in London about the trending topics based on given epsilon, the size of neighborhood, and n, the minimum number of points to search for (Adam 2017). The output

is on base map of Open Street Map, which should be combined with the output of Step3, then can identify where the trending words cluster.

Results Discussion

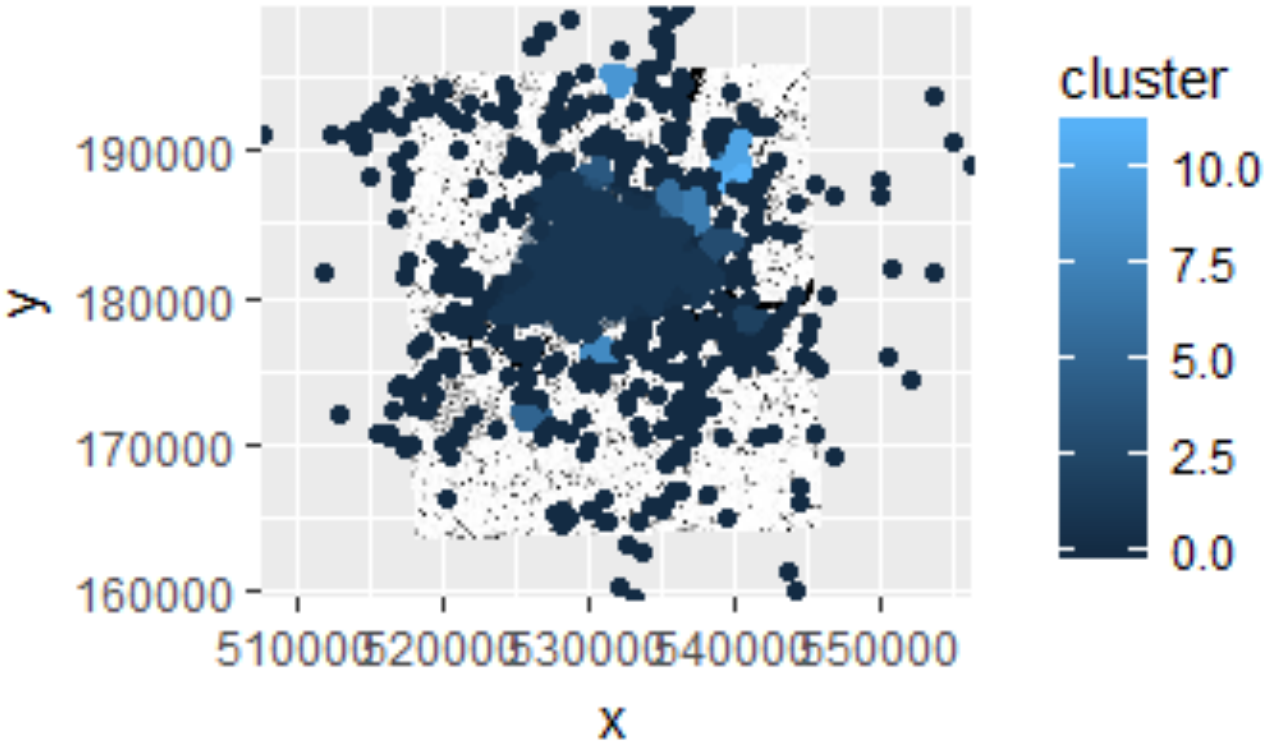
Figure3 is the result of Ripley'K about new year's fireworks in London, and Figure 1 is the points distribution of it.

Figure3: Ripley' K about new year' s fireworks in London



As shown in Fig 3, it can be figured out that until distances of around 1600 m, trending words are clustered, at around 1800m, the distribution show random and then dispersed between about 2000 and 4000 m. So, in Step5, the epsilon is set as 900m, and the n is set as 5. The result is shown in Figure 4.

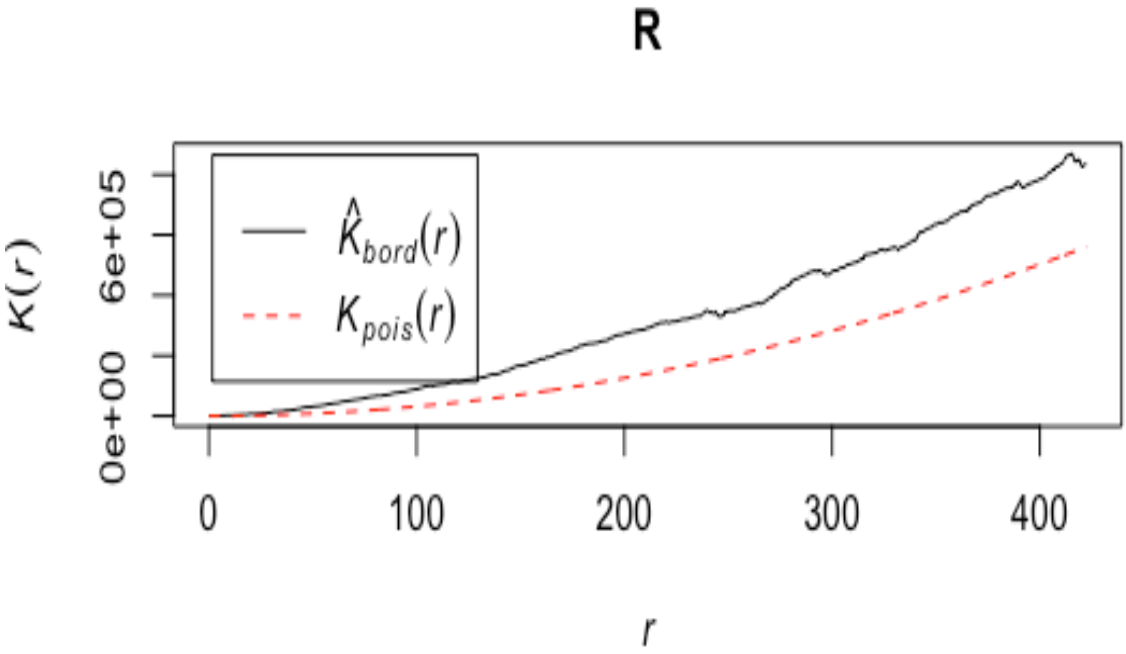
Figure4: DBSCAN about new year' s fireworks in London (R=800m, N=10)



The process can be done in certain borough as in London, the results about new year’s fireworks in City of London are shown in Figure5,6,7.

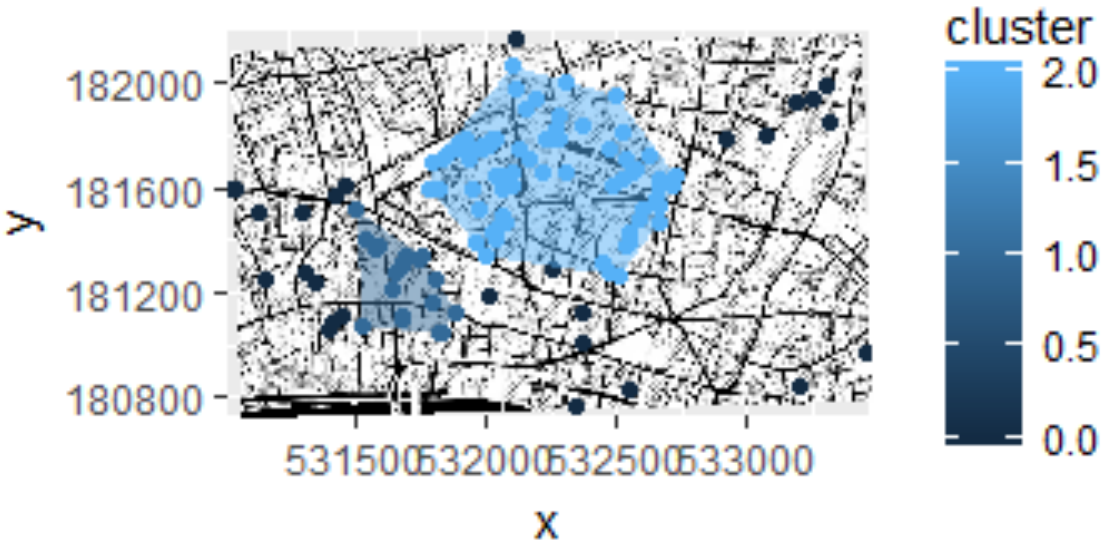
Figure5: Tweets about “new year fireworks of eve” distribute in City of London from Jan 1st to 2nd

Figure6: Ripley'K about new year's fireworks in City of London



Tweets Cluster

Figure7: DBSCAN about new year’s fireworks in City of London (R=200m, N=10)



Issues

As shown above, the limitation of API brings problems of the tool, the shortage of points with location information hamper the analysis, one of the solutions is to apply for many accounts of API, and search data in different PCs, then get the data together, which may make the research better. According to the practice, the data users get from API just a less than 1% sample of all points, and among the points, only 5% include location information.

As the increase of point count, the spatially bounded point clustering algorithm, proposed by Andrienko et al (2014) can be used, which can extract places from points (Ostermann et.al 2015). Then we can know where the people get together like in a stadium, hospital, or a park

instead of just know whether and where they cluster. The feasibility of the research is based on whether we can get the appropriate data or not.

Reference:

Arora, M., Gupta, R. and Kumaraguru, P., 2014. Indian Premier League (IPL), Cricket, Online Social Media. *arXiv preprint arXiv:1405.5009*.

Meeker, M., 2015. Internet trends 2015-code conference. *Glokalde*, *1*(3).

Thapa, L., 2016, July. Spatial-Temporal Analysis Of Social Media Data Related To Nepal Earthquake 2015. In *XXIII ISPRS Congress* (pp. 567-571).

Ostermann, F.O., Huang, H., Andrienko, G., Andrienko, N., Capineri, C., Farkas, K. and Purves, R.S., 2015. Extracting and comparing places using geo-social media. *ISPRS GEOSPATIAL WEEK 2015*, *2*(W5).

Andrienko, G., Andrienko, N., Schumann, H. and Tominski, C., 2014. Visualization of trajectory attributes in space–time cube and trajectory wall. In *Cartography from Pole to Pole* (pp. 157-163). Springer Berlin Heidelberg.

Michael W Kearney. 2016. Intro to rtweet: Collecting Twitter Data. https://cran.r-project.org/web/packages/rtweet/vignettes/intro.html.

APPENDIX 1: USER DOCUMENT

Prepare for run it.

1. Open the file “TWITTER.R” with R-Studio
2. The R had better be version 3.4.3 or over.
3. The following packages should be installed inadvance.
   Package(twitteR)
   Package(rtweet)
   Package(devtools)
   Package(bit64)
   Package(httr)
   Package(rjson)
   Package(data.table)
   Package(sf)
   Package(sp)
   Package(plyr)
   Package(spatstat)
   Package(sp)
   Package(rgeos)
   Package(maptools)
   Package(GISTools)
   Package(tmap)
   Package(sf)
   Package(geojsonio)
   Package(data.table)
   Package(sf)
   Package(OpenStreetMap)
   Package(Data.talble)
4. If the user wants to download data, the computer should be connected to the internet.
5. Run all code in “TWITTER.R” for once in advance.

Function

1. startprogram(times)

Function startprogrm(times) is to prepare for following work. Times is to record how much times the users use the function. The function should run before the following function.

For example:
at the first time the input is >- startprogrm(1)
The first time the user use the function, a web may ask you to input account name and password, I have applied one for the user.

#account:ucfnjwa@ucl.ac.uk
#password wj19931025

2. reade(a)

The function reade(a) is to read existing data, a is the file name that is in the same folder with the .R file.

For example:
now the file “fierworkeve.csv” is existing.
>- reade(“fierworkeve.csv”)

3. word(a)

The function is to search data from API of TWITTER, a is the trending word that is going to be searched. The process is not always so stable and will take a few minutes. And if the users want to get more data, there are two means, first, they can search data in days. In addition, applying for a few accounts of API and using different PCs is a good idea, but they have to edit the code in order to do it.

For example:
>-word(“fireworkeve”)

4. savemyresreach(a)

The function is to save the download data after word(a), the a is the file name you want to save and should end with .csv. The function just can work on one object once, the second time will fail.

For example:
>-savemyresreach(“fireworkeve.csv”)

The function 5-11 is operated based on the output of function 2 or function 3.

5. lookmapview(c,b)

The function is to show the interactive map from tmap package, c is the color of the dots, b is the title text.

For example:
>-lookmapview(“red”, “map of london”)

6. Ripleyk(times)

The function is to show Ripley’s K plot of London, times represent how much times the users use it.

Tweets Cluster

For example:
>- Ripleyk(1)

7. Cluster1(r,n)

The function is to do DBSCAN in London, r is the radius, n is the min number of points in a circle.

For example:
>-Cluster(800,5)

The function7,10 should be used in Windows, if it works in Mac, the java (JDK) should be installed in advance.

8. lookbor(a,b)

The function is to show interactive distribution map of a borough in London, a is the name of the borough, b is the title text.

For example:
>-Lookbor(“City of London”, “map of City of London”)
Function 9-11 are operated based on the output of function 8.

9. Ripleykborough(a)

The function is to show Ripley’s K plot of certain borough, a is the name of the borough.
For example:
>-Ripleykborough(“City of London”)

10. Clusterb(r,n,y1,x1,y2,x2)

The function is to do DBSCAN in certain borough, r is the radius, n is the min number of points in a circle, y1,x1,y2,x2 is the bbox of the borough that can be drawn from function11.

For example:
>-Clusterb(300,5, 51.3589310,-0.3101178,51.64527550,0.08932426)

11. boroughpre(a)

The function is to get y1,x1,y2,x2, the bbox of the borough, a is the name of the borough.

For example:
>-boroughpre(“City of London”)

the output is something like that in Fig8. Y1 = 51.5103500, x1=-0.1128281, y2=51.5229359, x2=-0.0782191.

Figure8: a sample of output in function11.

```
> boroughpre(1)
                  min         max
coords.x1 -0.1128281 -0.0782191
coords.x2 51.5103500 51.5229359
>
```