

Neighbourhood comparison for Uppsala or any other city with Foursquare

Introduction

In modern times, the regional planning has an increasing importance: globalization, migration flows, connectivity are examples of factors that changes the society continuously and this reflects in how our cities are built.

Foursquare, with its API, is a gold mine of information that can be used as basis to other studies.

This work aims to give a simple tool that suits the data provided by Foursquare to create a unique fingerprint of a certain area, such as a neighbourhood. This fingerprint can be used to compare the area with other ones, which can be anything.

A quick comparison among different zones gives information that can be suited as ground for regional planning, demographics and more.

Uppsala has been taken as an example since it has quite different neighbourhoods; however, the Python notebook can be easily modified to make other kinds of comparisons Examples: two commercial areas in two different cities; a Chinatown against a true Chinese city; low-value residential zones in different cities of the same nation. Professionals such as e. g. researchers and urbanists can suit this work as a tool for their inquiries.

Data

The main data is retrieved from Foursquare by means of its API: basically, the only thing that is needed is the number of venues of a certain type, in a given area.

Regarding the lists of locations, there are three modes:

1. List of neighbourhood names loaded from a file.
2. Uppsala case: a list of neighbourhoods in Uppsala is web-scraped.
3. Single comparison: two locations are compared according to hard-coded constants in the configuration parameters section.

For the Uppsala case, the swedish Wikipedia article about Uppsala's neighbourhoods is web-scraped: https://sv.wikipedia.org/wiki/Lista_%C3%B6ver_stadsdelar_i_Uppsala

Finally, the coordinates of each zone is retrieved by means of the geopy.geocoders library.

Methodology

First of all, the Python notebook contains a section called “Configuration parameters” which allows the user to run the notebook in three different modes, increasing the flexibility of the study. Here there is a copy of the notebook cell, with the comments explaining how each parameter affects the run:

```
Mode = 2 # 1 = list of neighbours loaded from file; 2 = Uppsala case (web-scraped neighbour list);
other = two single neighbourhoods comparison

Radius = 500 # Range in meters within the venues are searched (too small -> unrepresented venues, too
large -> overlapping venues)

N_clusters = 5 # Number of predefined clusters (for mode 1, it must be less or equal to 2)

# Exclusive parameters for Mode 1

Filename = "Test.csv" # Filename for the list of neighbourhoods (only for mode 1)

# Exclusive parameters for Mode 3

Input_location_A = "Gottsunda" # First neighbour to compare (only mode 3)

Input_location_B = "Fittja" # Second neighbour to compare (only mode 3)

Input_city_A = "Uppsala" # Name of the city where the first neighbour is located (only mode 3, can be
set to null if the location name is already univocal)

Input_city_B = "Stockholm" # Name of the city where the second neighbour is located (only mode 3, can
be set to null if the location name is already univocal)

# Foursquare account parameters

# PLEASE REPLACE THESE STRINGS WITH YOUR FOURSQUARE'S PERSONAL ACCOUNT

CLIENT_ID = '...'

CLIENT_SECRET = '...'

VERSION = '20180605'
```

The data retrieved from Foursquare is put into a list of data frames. Each of them contains a list of venue types and the count of how many of them are present in the area.

For the Uppsala case, as mentioned in Data, a web scraping library such as Beautiful Soup was used. In general, the Python notebook can be configured with another web-scraped page or even just another dataset containing a simple list of neighbourhood names. Nothing more is needed since the geopy.geocoders library can retrieve the coordinates needed by Foursquare with only the name.

Each neighbourhood becomes the input for Foursquare, which returns a list of venues. This list is used to create a data frame containing the type of venue and the quantity present in the neighbourhood. Below you can see two examples taken from the Uppsala case:

	Type	Ultuna
0	Pharmacy	1
1	Diner	1
2	Food & Drink Shop	1
3	Bus Station	1
4	Gym / Fitness Center	1

Figure 1 Venues in Ultuna, Uppsala

	Type	Sala backe
0	Bakery	2
1	Middle Eastern Restaurant	1
2	Grocery Store	1
3	Park	1
4	Pizza Place	1
5	Bus Stop	1
6	Fast Food Restaurant	1

Figure 2 Venues in Sala backe, Uppsala

Once all the locations have been scanned, the corresponding data frames are combined in a larger one by means of a full join operation. The full join assures that all the type of venue are taken in account for every zone. So, if Foursquare doesn't report any bakery in Ultuna, their presence in another location (like Sala Backe) will assure that the merged data will contain a complete list of venue types. This is what actually makes different neighbourhoods like Ultuna and Sala Backe comparable.

The merged data frame is a table where the first column is the venues type name and each other column contains the quantity of each type in a certain neighbourhood.

Calculating the correlation among each combination of neighbourhood gives an idea of how similar to each other they are. However, a clustering algorithm such as K-means produces more significant information.

In order to feed the merged data in the K-means algorithm, it necessary some preliminary transformations: 1) remove the categorical data (column "Type"); 2) transpose the matrix, since the goal is to create a cluster of neighbourhoods rather than venue types; 3) give to the new columns the name of the corresponding venue type.

Now the data is ready to be processed and then displayed. Again, an image taken from the Uppsala case:

	Cluster Labels	Seafood Restaurant	Coffee Shop	Bakery	Juice Bar	Beer Bar	Restaurant	American Restaurant	Scandinavian Restaurant	Pub	...	Hostel	Indian Restaurant	Gas Station	Skating Rink	Hockey Arena
Gottsunda	3	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Tunabackar	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Ulleråker	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Sala backe	3	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Kvarngärdet	4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	1.0	1.0	1.0	0.0	0.0

Figure 3 Merged data for neighbourhoods in Uppsala

Results

The kind of results largely depends by how the notebook was configured. In general, the quality of data from Foursquare varies enormously according to country and region. This has a huge impact on how the locations are “fingerprinted” by the script. Furthermore, the cluster algorithm is quite sensible towards two parameters especially: the number of clusters and the radius of search. The radius shouldn’t be too small, because it wouldn’t cover the location’s size; however, a too large radius might lead to overlapping areas, which basically means high correlation between two locations which actually aren’t so similar.

In the following page there are some results found with Uppsala’s neighbourhoods, using a radius of 500 meters and 5 clusters:

[illegible]

Figure 4 Correlation among Uppsala's neighborhoods

Neighbourhood	Cluster
Ärna	0
Årsta	0
Bergsbrunna	0
Boländerna	0
Brillinge	0
Eriksberg	0
Fålhagen	0
Fyrislund	0
Gamla Uppsala	0
Gränby	0
Håga	0
Hällby	0
Husbyborg	0
Kuggebro	0
Kungsängen	0
Kvarnbo	0
Librobäck	0
Löten	0
Luthagen	0
Nåntuna	0
Norby	0
Polacksbacken	0
Rickomberga	0
Sävja	0
Slavsta	0
Starbo	0
Stenhagen	0
Svartbäcken	0
Ultuna	0
Vaksala	0
Valsätra	0
Vårdsätra	0
Vilan	0
Fjärdingen	1
City	2
Berthåga	3
Ekeby	3
Flogsta	3
Gottsunda	3
Kåbo	3
Nyby	3
Sala backe	3
Sunnersta	3
Tunabackar	3
Ulleråker	3
Kvarngärdet	4

Figure 5 Cluster of neighbourhoods in Uppsala

1. Many neighbourhoods belong to cluster 0. This cluster corresponds to neighbourhoods where Foursquare reported only a few venues, which make them similar to each other.
2. Clusters 1 and 2 were assigned only to Fjärdingen and City. These are the most central neighbourhoods in Uppsala: they are small and very dense of venues.
3. There is a bunch of neighbourhoods which belong to cluster 3 and 4. Analysing them more in detail, it's possible to see that, in these neighbourhoods, pizza places are abundant, while absent elsewhere:

	Cluster Labels	Pizza Place
Husbyborg	0	0
Hällby	0	0
Librobäck	0	0
Luthagen	0	0
Rickomberga	0	0
Stenhagen	0	0
Eriksberg	0	0
Håga	0	0
Kvarnbo	0	0
Norby	0	0
Polacksbacken	0	0
Starbo	0	0
Ultuna	0	0
Valsätra	0	0
Vårdsätra	0	0
Bergsbrunna	0	0
Nåntuna	0	0
Sävja	0	0
Vilan	0	0
Boländerna	0	0
Fyrislund	0	0
Fålhagen	0	0
Kungsängen	0	0
Kuggebro	0	0
Slavsta	0	0
Vaksala	0	0
Årsta	0	0
Brillinge	0	0
Gamla Uppsala	0	0
Gränby	0	0
Löten	0	0
Svartbäcken	0	0
Ärna	0	0
Fjärdingen	1	0
City	2	0
Berthåga	3	2
Flogsta	3	2
Ekeby	3	2
Kåbo	3	1
Gottsunda	3	1
Sunnersta	3	1
Ulleråker	3	1
Sala backe	3	1
Nyby	3	1
Tunabackar	3	1
Kvarngärdet	4	1

Figure 6 Relationship between clusters and pizza places

Finally, just an example obtained with Mode 3, by comparing Sweden’s two largest cities: Stockholm and Göteborg. In this case, the radius was set to 2000 meters and the number of clusters to 2:

	Göteborg	Stockholm
Göteborg	1.000000	0.498873
Stockholm	0.498873	1.000000

Figure 7Correlation between Göteborg and Stockholm

In this mode, since there are only two locations for two clusters, the only interesting number is the correlation. It’s about 0.5, which means that the two city centres have a similar distribution of venues in the range of 2 km.

Discussion

Asdasd

Conclusion

Asdasd