Jointly analysis of geospatial and friendship of Gowalla data

15 March 2017

Juan Benjumea

# Introduction

Gowalla is a location-based social networking website where users share their locations by checking-in. The friendship network is undirected and was collected using their public API, and consists of 196591 nodes and 950327 edges. It has a total of 6,442,890 check-ins of these users over the period of Feb. 2009 - Oct. 2010. The aim of the project is to use Data Science tools to analyse and build a recommendation system for both the geospatial checkins and the social network links in that database. As a part of the project I will tag a users as a locals or tourists (who mostly visit monuments, stay in hotels ...) and adapt responses given that tag. I will use the 17400 Paris check-ins from 1366 users and I will join Gowalla location data with Google Places to separate the type of venues reported Gowalla's users.

# Datasets description

Multiple sources of information were gather and used in the developing of this project; the datasets includes the Stanford's SNAP Gowalla data [11], queries to Google Maps API and vectorial geodata from OpenStreetMaps.

The [Gowalla](https://snap.stanford.edu/data/loc-gowalla.html) is an anonymized and clean dataset collected from February 2009 to October 2010 from the Gowalla's startup to capture human mobility in a location-based social network (LBSN). In Gowalla LBSN, people were able to **check-in** at places or **spots** that they visited near to their local vicinity; Check-ins were collected from a mobile application or through a mobile website, the incentive for users was to get advantages in the places they check-in[[1]](#footnote-1). SNAP's Gowalla data is split in two datasets: a) a collection of more than 6.4 million of individual check-ins[[2]](#footnote-2) and b) an undirected graph[[3]](#footnote-3) with 196591 nodes and 950327 edges [11]

For this project a subset of the check-in in Gowalla data was extracted; the subset corresponds a selection of data from Paris, extracting all datapoints in a radius of 30km from the position 48.86°N, 2.35°E (Google’s Paris location). The selected Gowalla data contains for Paris consists in 17496 check-ins between September 2009 and October 2010 [[see github](https://github.com/jubenjum/dssp5-proj/blob/master/data/loc-gowalla_totalCheckins_Paris.txt)]. The undirected graph for Gowalla is not used in this Data Science pipeline given the limits on the selected data as a low number of connections for within Paris, searching time, and other constrains.

The data from Gowalla was enriched using information fetched from Google Places using their web-service API (function [nearbysearch](https://developers.google.com/places/web-service/search)). I developed scripts to do multiple web-scrapping and parsing Google's JSON geodata. The data of Gowalla for Paris has 4178 different check-ins locations (different latitude, longitude pairs), for each one of those locations I searched and fetched all the Google Places within 100m radius of that position, in total I got a total of 19089 individual Google Places.

Other data used in the project is OpenStreetMap are vectors in GEOJSON format for the Paris districts and the Seine River, data that is used on the visualization of Paris geodata.

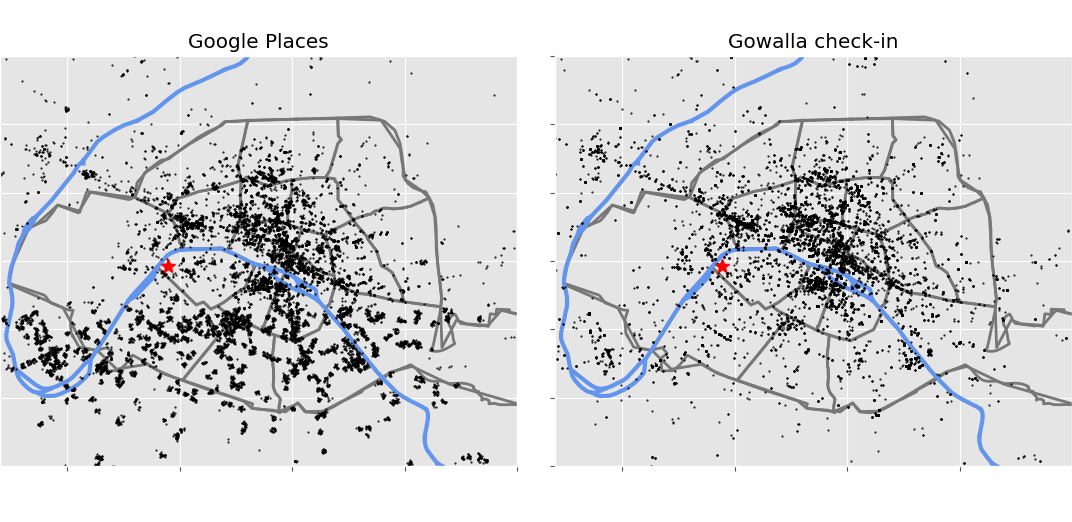


Figure 1: Map of selected datasets, left are the returned POI from Google Places, right are the Gowalla check-ins in Paris, the red start is the location of Eiffel Tower

# Preprocessing

Raw data was given in multiples formats depending of the source; Gowalla format consist on fixed width the columns, that contains: user identification number, the UTC time of the check-in, the location in decimal latitude and longitude, the column is the a unique spot id. For these datapoints the only needed modifications done on the data is on the UTC time, that time was corrected using a day saving time function, this dataset was already clean for research by SNAP, then no extra modifications were done on this data. Gowalla’s selected data was converted to a csv file to do the data processing.

To obtain the Google Places data, for all unique locations in Paris from Gowalla I made requests to for all places around 100m these points in Google using their web-service and requesting all types of places (hotel, monuments, zoo …). Google Places web-service returns a JSON file for each query, as I have 4178 queries and Google set a quota of 2500 queries/day per developer key, I got the information in two batches. A selection of the JSON fields for this data were selected, see Feature engineering section for details on the selection.

Given the proximity of Gowalla check-in locations and the radius of query on Google Sites, I got duplicated places from Google; I searched and removed all that duplicated data from the parsed data using Google data hash code. The total number of POI in Google Places is 30286 POIs, and 19089 POIs after cleaning.

From the map Figure [1](#fig:1), it can be see that there is a fraction of points in Gowalla that are not present on Google, also the density of Google is higher, that can happened because:

* The Gowalla spots doesn't exist anymore, closed restaurants, shops etc.
* Check-ins in private properties (e.g. apartment, offices)
* Google Places POI is gather around 100m of GoWalla's data, though there is a higher density of information.

## Analysis of Gowalla data for Paris

From the analysis of 17496 Gowalla check-in locations, the places that are most visited are in Table 1. The first place corresponds to is the CDG Airport and analysing the data it was found that check-in places at the airport correspond to car rental agencies, that could be due to promotional coupons or reductions when using the application and doing those check-ins. The other 9 major check-ins corresponds to places near-to or in site seeing, which is around 10% of all the check-ins in Paris, meaning that a proportion of data is provided by tourists. However there are two places that are not exclusive from tourists that are Gare du Nord and Mntreuil:

|  |  |
| --- | --- |
| **# Check-ins** | **Venue** |
| 402 | CDG Airport |
| 198 | Louvre |
| 194 | Pont des Arts |
| 171 | Eiffel Tower |
| 267 | BNF/François-Mitterrand |
| 114 | Gare du Nord (local people?) |
| 106 | Notre Dame |
| 100 | A place near to BNF |
| 93 | Arc de Triomphe |
| 91 | Montreuil (local people?) |

Table 1: The top 10 in Paris from Gowalla crossing information with Google Places

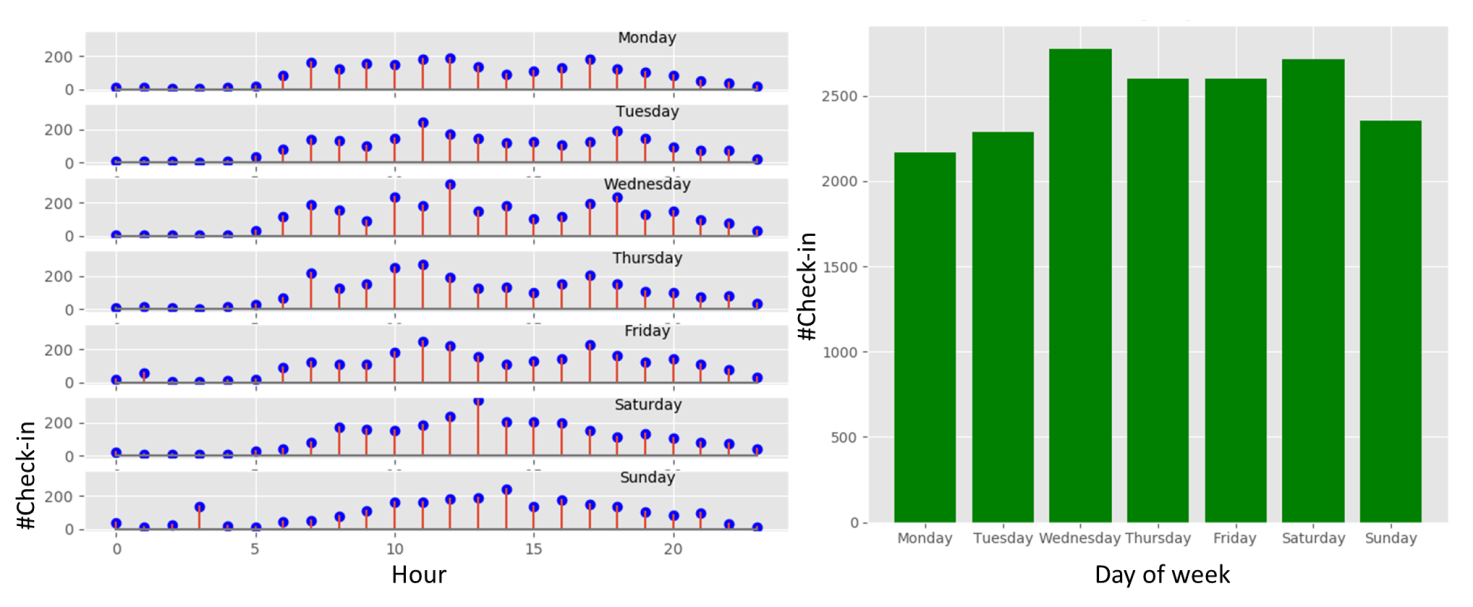


Figure 2: Temporal distribution of Gowalla check-ins in Paris, left the hourly check-ins, right the distribution by day of the week

The temporal distributions of Gowalla check-ins in Paris is shown in Figure 2. The week distribution of data is almost constant (right), a small reduction in the number of check-ins on Monday and Tuesday and maybe due to that museums are close. For the hours of the day it’s clear that there is a pattern, most of the check-ins are done during day time (5am to 8pm), and pick around midday, and from Monday to Friday at 7-8am and 5-6pm, that can be a contribution in check-ins from local people.

# Feature Engineering

Two parts were necessary to complete with this project from the engineering point of view, the first is the selection of tools required to process the data in an efficient way given the size and complexity of the processed data, and the second is the feature selection from the data on itself.

## Software packages and used Tools

One part of the entire Data Science pipeline used to produce the results in this report is done with standard Linux command (bash, awk, sed, cut …), the size of the data and the number of samples allows making a fast exploration of data, and do searching, cleaning and querying efficiently using those command line tools. For the Data Science part I used the packages that we learned on the DSSP5 at Ecole Polythenique, for the developing part I used python and the following packages:

* **Matplotlib**: to make figures and maps.
* **Scikit-learn**: the non-supervised classification with KNN
* **FAISS**: a new KNN implementation from Facebook AI group [10], which allow to project features in low dimensions and it does searches in high dimension of data, also uses CUDA/CPU to accelerate the computations.
* **Pandas**: as a container for all data into dataframes.

## Feature selection

From Gowalla’s dataset its was selected only the check-in data for Paris, the data was transformed to a pandas dataframe, t time stamps were corrected to take into account day time saving.

Google Places returns JSON containers with the structure described at their webpage[[4]](#footnote-4), for this DSSP pipeline used only a subset of Google data and the selected is:

* **geometry/location**: latitude and longitude, numerical values
* **id**: location Google’s hash as a 10 character string
* **name**: name of the place, a string
* **type**: a string with one of Google's supported see Table A1 or [types](https://developers.google.com/places/supported_types)

Google classify their places in 130 types, the full list of all supported types are listed on the Table A1 and Google’s developer webpage. All Google parsed values are keep on dataframes with the fields: latitude, longitude, place\_id (Google’s hash code), name of venue[[5]](#footnote-5) and

the place can contain multiple types, for example Eiffel Tower is in Google places is: *restaurant, shop, amusement park* and *establishment*, because there are those venues in that site seeing, and each single place have a different signature; these Google string were translated on a categorical vector of dimension d=130, containing 1 when feature is present the and 0 wen absent,

that gives for each Google Place location a categorical variable that I call spot signature in the code. These spot signatures are used to build a user signature, that is the

A map of all location of Gowalla check-ins and Google Places are in the Figure 1.

# Methods used for classification

# Analysis and interpretations of the results

difference can be explained on recurrent visit of same people to the same places or different people visiting the same places.

[github repo](https://github.com/jubenjum/dssp5-proj)

Faiss handles collections of vectors of a fixed dimensionality d, typically a few 10s to 100s. These collections can be stored in matrices. We assume row-major storage, ie. the j'th component of vector number is stored in row , column of the matrix. Faiss uses only 32-bit floating point matrices.

# Potential further work

# References

[1] G. Ference, M. Ye, and W.-C. Lee, “Location recommendation for out-of-town users in location-based social networks,” in *Proceedings of the 22Nd aCM international conference on information & knowledge management*, 2013, pp. 721–726 [Online]. Available: <http://doi.acm.org/10.1145/2505515.2505637>

[2] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, “Time-aware point-of-interest recommendation,” in *Proceedings of the 36th international aCM sIGIR conference on research and development in information retrieval*, 2013, pp. 363–372 [Online]. Available: <http://doi.acm.org/10.1145/2484028.2484030>

[3] H. Wang, Z. Li, and W.-C. Lee, “PGT: Measuring mobility relationship using personal, global and temporal factors,” in *Data mining (iCDM), 2014 iEEE international conference on*, 2014, pp. 570–579.

[4] S. M. Rahimi and X. Wang, “Location recommendation based on periodicity of human activities and location categories,” in *Advances in knowledge discovery and data mining: 17th pacific-asia conference, pAKDD 2013, gold coast, australia, april 14-17, 2013, proceedings, part iI*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 377–389 [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-37456-2_32>

[5] A. Narayanan and V. Shmatikov, “De-anonymizing social networks,” in *Security and privacy, 2009 30th iEEE symposium on*, 2009, pp. 173–187.

[6] J.-D. Zhang and C.-Y. Chow, “GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations,” in *Proceedings of the 38th international aCM sIGIR conference on research and development in information retrieval*, 2015, pp. 443–452 [Online]. Available: <http://doi.acm.org/10.1145/2766462.2767711>

[7] Q. Yuan, W. Zhang, C. Zhang, X. Geng, G. Cong, and J. Han, “PRED: Periodic region detection for mobility modeling of social media users,” 2016.

[8] L. Rossi and M. Musolesi, “It’s the way you check-in: Identifying users in location-based social networks,” in *Proceedings of the second aCM conference on online social networks*, 2014, pp. 215–226 [Online]. Available: <http://doi.acm.org/10.1145/2660460.2660485>

[9] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, “A random walk around the city: New venue recommendation in location-based social networks,” in *2012 international conference on privacy, security, risk and trust and 2012 international confernece on social computing*, 2012, pp. 144–153.

[10] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with gPUs,” *CoRR*, vol. abs/1702.08734, 2017 [Online]. Available: <http://arxiv.org/abs/1702.08734>

[11] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: User movement in location-based social networks,” in *Proceedings of the 17th aCM sIGKDD international conference on knowledge discovery and data mining*, 2011, pp. 1082–1090 [Online]. Available: <http://doi.acm.org/10.1145/2020408.2020579>

# Appendices

|  |  |  |  |
| --- | --- | --- | --- |
| accounting | pet\_store | embassy | taxi\_stand |
| cemetery | storage | light\_rail\_station | campground |
| gym | amusement\_park | political | funeral\_home |
| natural\_feature | courthouse | sublocality\_level\_3 | meal\_takeaway |
| rv\_park | insurance\_agency | beauty\_salon | premise |
| administrative\_area\_level\_1 | pharmacy | establishment | train\_station |
| church | store | liquor\_store | car\_dealer |
| hair\_care | aquarium | postal\_code | furniture\_store |
| neighborhood | dentist | sublocality\_level\_4 | mosque |
| school | intersection | bicycle\_store | real\_estate\_agency |
| administrative\_area\_level\_2 | physiotherapist | finance | transit\_station |
| city\_hall | street\_address | local\_government\_office | car\_rental |
| hardware\_store | art\_gallery | postal\_code\_prefix | gas\_station |
| night\_club | department\_store | sublocality\_level\_5 | movie\_rental |
| shoe\_store | jewelry\_store | book\_store | restaurant |
| administrative\_area\_level\_3 | place\_of\_worship | fire\_station | travel\_agency |
| clothing\_store | street\_number | locality | car\_repair |
| health | atm | postal\_code\_suffix | general\_contractor |
| painter | doctor | subpremise | movie\_theater |
| shopping\_mall | laundry | bowling\_alley | roofing\_contractor |
| administrative\_area\_level\_4 | plumber | floor | university |
| colloquial\_area | sublocality | locksmith | car\_wash |
| hindu\_temple | bakery | postal\_town | geocode |
| park | electrician | subway\_station | moving\_company |
| spa | lawyer | bus\_station | room |
| administrative\_area\_level\_5 | point\_of\_interest | florist | veterinary\_care |
| convenience\_store | sublocality\_level\_1 | lodging | casino |
| home\_goods\_store | bank | post\_box | grocery\_or\_supermarket |
| parking | electronics\_store | synagogue | museum |
| stadium | library | cafe | route |
| airport | police | food | zoo |
| country | sublocality\_level\_2 | meal\_delivery |  |
| hospital | bar | post\_office |  |

Table A1. List of all types returned by Google Place API

1. https://en.wikipedia.org/wiki/Gowalla [↑](#footnote-ref-1)
2. the columns are 1-user\_id, 2-UTC\_Time, 3-latitude, 4-longitude and 5-spot\_id; 1,3,4 are directly used on the DSSP's workflow; 2 is corrected for day saving time, however all variables are keep on the dataframes [↑](#footnote-ref-2)
3. Contains in the columns user\_id friend\_id. [↑](#footnote-ref-3)
4. https://developers.google.com/places/web-service/search [↑](#footnote-ref-4)
5. UTF-8 string UTF-8 with multiple language description of each place [↑](#footnote-ref-5)