

Using Google Trends, Gaussian Mixture Models and DBSCAN for the Estimation of Twitter User Home Location

Paola Zola[1(✉)], Paulo Cortez[2], and Maurizio Tesconi[1]

[1] Institute for Informatics and Telematics (IIT) of the National Research Council of Italy (CNR), Pisa, Italy
{paola.zola,maurizio.tesconi}@iit.cnr.it
[2] ALGORITMI Centre, Department of Information Systems, University of Minho, 4804-533 Guimarães, Portugal
pcortez@dsi.uminho.pt

Abstract. In this work we propose a novel approach to estimate the home location of Twitter users. Given a list of Twitter users, we extract their timelines (up to 3,200) using the Twitter Application Programming Interface (API) service. We use Google Trends to obtain a list of cities in which the nouns of a specific Twitter user are more popular. Then, based on word popularity, we sample the geographical coordinates (latitude, longitude) over all the world surface. Finally, the Gaussian Mixture Model and the DBSCAN clustering algorithms are implemented to estimate the users' geographic coordinates. The results are evaluated using the mean and median error computed on the Haversine distance. Competitive findings are achieved when compared with a baseline approach that estimated the users' location given the Google Trends city mode.

Keywords: Geostatistics · Gaussian Mixture Models · DBSCAN · Twitter · Google Trends

1 Introduction

Inferring Twitter users home location is a growing interest research topic given its importance for many real-world applications. For example, knowing the location of people is fundamental in event detection studies, recommendation systems, health care support systems, friendship network analysis and so on [3,4].

Assign a location to Twitter users based on their message contents is a non-trivial task. In fact, only a few percentage of tweet is geotagged [9] and users often do not add reliable information to their profile information. Research studies addressing the geolocation problem either apply unsupervised (i.e., based on text location dictionaries) or supervised approaches (i.e., models trained on a sample of geotagged tweets).

Differently, in this work, we propose a novel unsupervised learning tool aiming to infer Twitter users home location at coordinate-level that is based on Google

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12253, pp. 526–534, 2020.
https://doi.org/10.1007/978-3-030-58814-4_38

Trends engine. The proposed approach exploits our previous work [11], where Google Trends (GT) was used to assign the home Country to Twitter accounts. The algorithm proposed in [11] firstly collected the users timelines to extract all nouns. The nouns collection, for each account, was then fetched to GT in order to determine the user's most probable country. In this paper we extend the GT applicability to a more fine grain location estimation level. In particular, we aim to assign to each Twitter account her home location in terms of geographic coordinates (latitude, longitude). The experiment are conducted on a dataset composed by 2,880 Twitter accounts with a verified location, and, for each user we consider all historical messages (up to 3,200 tweets). For each account, we extract the GT cities distribution passing, as keyword, the nouns (generic and proper) hold in the tweets collection. Having the cities distribution, we convert it on a synthetic world surface based on city polygon areas that respects the GT distribution. Then, the geographic sampled data points are used to fit two clustering algorithms: Gaussian Mixture Models (GMM) and the Density Based Spatial Clustering of Applications with Noise (DBSCAN).

The results are evaluated using mean and median absolute error computed on the Haversine distance from the ground truth users home location. The paper is organized as follows: Sect. 2 an overview of previous work about Twitter geolocation is proposed, then in Sect. 3 the proposed approach is explained and in Sect. 4 discusses the obtained results. Finally, Sect. 5 concludes the paper.

2 Overview on Twitter User Location Estimation

Approaches for social media users geolocation might be distinguish in two classes: word distribution (WD) and social network (SN) based. The first group (WD) analyzes the texts content identifying location indicative words (LIW) and/or using specific dictionaries for location entity recognition [8,9]. However, as argued in [8], focusing only on LIW and specific place mentions can discard important information linked to generic nouns. Users geolocation methods based on friendship social networks (SN) exploits users social interactions in order to infer the unknown location. A hybrid approach that merges WD and SN has been implemented in [2].

Focusing on WD approaches, which includes also our proposed method, initial studies mainly apply traditional natural language processing techniques [8], while recent researches adopt machine learning techniques [10]. Another feature to be considered is the location estimation granularity. In fact, previous works either focus on specific regions/countries [2,8] providing a street or coordinate level estimation or considered the entire world surface but predicting only the users' country [10,11]. Differently, in this work we apply clustering algorithms to assign to each user a probability distribution over the whole world map in order to estimate an unique point associated with the user home location.

3 Proposed Approach

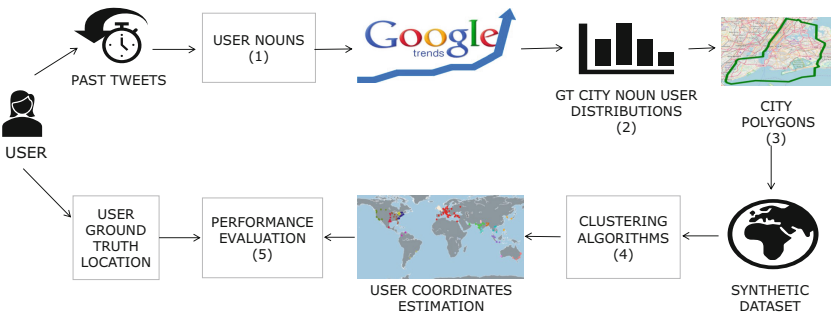


Fig. 1. Proposed approach

The proposed approach, which follows the one adopted in [11], is depicted in Fig. 1. The proposed algorithm is composed by five steps and it uses only tweet nouns assuming that they are the most representative part of speech able to identify different countries [8]. For user u, the proposed approach works by first identifying the sequence of all nouns $\mathbf{n}_u = \langle n_1, n_2, ..., n_{l_u} \rangle$ (phase (1) in Fig. 1). To obtain $\mathbf{n}_u$, the tweets are first preprocessed by transforming the text to lowercase and removing English stop words. For each noun $n_i \in \mathbf{n}_u$, a GT query is executed by using the `Pytrends` Python module. The GT query results for noun n_i is a sequence with integers confidence scores for the most frequent cities C that have typed the specific noun $n_i \in \mathbf{n}_u$ (phase (2) in Fig. 1).

Denoting C_u as the city distribution for a given user given all her nouns, we first locate these cities on the world surface, generating then samples of GT geographic data points that are used to fit a clustering model, with the final user location being estimated as the centroid of the largest cluster. To perform this operation, we computed the polygon areas $p(c)$ for all cities $c \in C$. The $p(c)$ is defined as a two-dimensional polygon where the edges correspond to the physical borders of the city c. The aim is to uniformly sample a certain number of two-dimensional points (latitude and longitude) in the city polygon representing the GT integer confidence scores generated for each city (phase (3) in Fig. 1). To obtain the cities polygons we downloaded the data from OpenStreetMap website[1]. If the specific $i-th$ city is not available on OpenStreetMap dataset we derived, as a proxy of the city polygon, the city surface built around the city centre coordinates. Given the coordinates of the $i-th$ city center, the circumference is computed considering a the radius equal to $r = \sqrt{\frac{A}{\pi}}$ where A correspond to the city area. The city centre coordinates (latitude and longitude)

[1] http://www.openstreetmap.org.

are derived from Wikidata[2]. The developed Python library for the city polygon definition is freely available on GitHub[3].

Once all city polygons are computed, we sample a finite number of points from each city polygons such that the GT data distribution is respected (e.g., there will be more points inside New York if the associated GT noun global scores are higher). The obtained GT noun city sampled geographic data points are then used to fit a clustering model (one for each user). Finally, the user location is estimated as the centroid of the largest cluster (phase (4) in Fig. 1).

In this paper we consider the GMM and the DBSCAN algorithms. The optimal number of clusters for the GMM is evaluated with the Bayesian Information Criteria (BIC). Moreover, we experimented different values associated with the maximum number of cluster in the model ranging in {5,10,20,30,50}, in order to evaluate how the algorithm performance changes given this parameter. For the DBSCAN algorithm we used the default parameters of `sklearn` Python module.

3.1 Clustering Algorithms

Gaussian Mixture Models (GMM) is a probabilistic model to represent the distribution of a population as a linear combination of Gaussian [1]. Usually, each Gaussian k in the GMM is called component. If each component in the mixture is interpreted as a cluster of the dataset then the GMM can be seen as a unsupervised learning algorithm for classification. Let's define K as the number of components in the GMM, α_k as the weight of the kth component, $\boldsymbol{\mu}_k$ and Σ_k the location and the scale parameter of the component k, then the GMM model for the random variable X has the following distribution:

$$p(X) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(X|\boldsymbol{\mu}_k, \Sigma_k) \tag{1}$$

where $\sum_{k=1}^{K} \alpha_k = 1$.

The estimation is performed with the Expectation Maximization Algorithm (EM), an iterative procedure which alternates the computation of the expectation of the likelihood (E-step) with the maximization of the expectation of the likelihood (M-step) up to convergence. The only parameter that needs to be fixed in a GMM is the number of clusters K. In this work, we select the best K value according to the Bayesian Information Criterion (BIC) as performed in [2].

DBSCAN (Density Based Spatial Clustering of Applications with Noise) is a density based algorithm to cluster data in the presence of noise. It is particularly good in discovering clusters of arbitrary shape [6]. The intuition behind the DBSCAN is that relevant clusters should have a higher point density when compared with noise. Thus, isolated points are considered noise, while regions with

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page.
[3] https://github.com/CostRagno/geopolygon.

an high density of points are considered clusters. Let's define $\boldsymbol{x}_i$, a threshold ϵ, D_i as the region of size ϵ centred in $\boldsymbol{x}_i$ and $dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as a distance measure between two points. Then, the Eps-neighbourhood N_ϵ of $\boldsymbol{x}_i$ results:

$$N_\epsilon(\boldsymbol{x}_i) = \{\boldsymbol{x}_j \in D_i | dist(\boldsymbol{x}_i, \boldsymbol{x}_j) < \epsilon\}. \tag{2}$$

The DBSCAN algorithm is based on four main concepts:

- the point $\boldsymbol{x}_i$ is a core point if $\#N_\epsilon(\boldsymbol{x}_i) > N$, where N is the minimum number of points to consider a point as core point;
- the point $\boldsymbol{x}_j$ is directly density-reachable from $\boldsymbol{x}_i$ if point $\boldsymbol{x}_j \in N_\epsilon(\boldsymbol{x}_i)$ and $\#N_\epsilon(\boldsymbol{x}_i) > N$;
- the point $\boldsymbol{x}_j$ is density-reachable if there is a chain of points $\boldsymbol{x}_j, \boldsymbol{x}_{j+1} \ldots, \boldsymbol{x}_i$ such that $\boldsymbol{x}_{j+1}$ is directly-density reachable from $\boldsymbol{x}_j$;
- all points not reachable are considered noise.

Note that if the point $\boldsymbol{x}_i$ is reachable but $\#N_\epsilon(\boldsymbol{x}_i) < N$, then it is called a border point. The DBSCAN evaluates all the points in a dataset and it considers as part of a cluster all the reachable.

3.2 Evaluation

We assume a pure unsupervised learning approach, with no ground truth location data being used by the clustering methods during the estimation of the user location. Thus, the ground truth data is only used as an external attribute during the evaluation phase, for validation purposes, allowing to compute absolute error distances (phase (5) in Fig. 1). In particular, we adopt the Mean Absolute Error (MAE) and the Median Absolute Error (MdAE) on the Haversine distance measures expressed in Kilometres as proposed in [5]. The Haversine formula measures the great-circle distance between two points on a sphere given their longitudes and latitudes. The great-circle distance is the shortest distance between two points on the surface. Moreover, for comparison purposes, we computed as a baseline method the simple city mode for each user, given her GT cities distribution, extracting the city center coordinates.

4 Data and Results

4.1 Data

The dataset was collected between January and February 2018 and it is composed by 2,880 Twitter Users with a verified home location. The information regarding the ground truth country are based on a double check system that matched the metadata information (the address provided by the user in her Twitter account) and the analysis of location indicative words (LIW) given the historical tweets for each account [11]. The original dataset, freely available on GitHub[4], is composed by 3,298 accounts but, in this work, we keep only the

[4] https://github.com/paolazola/Twitter-country-geolocation.

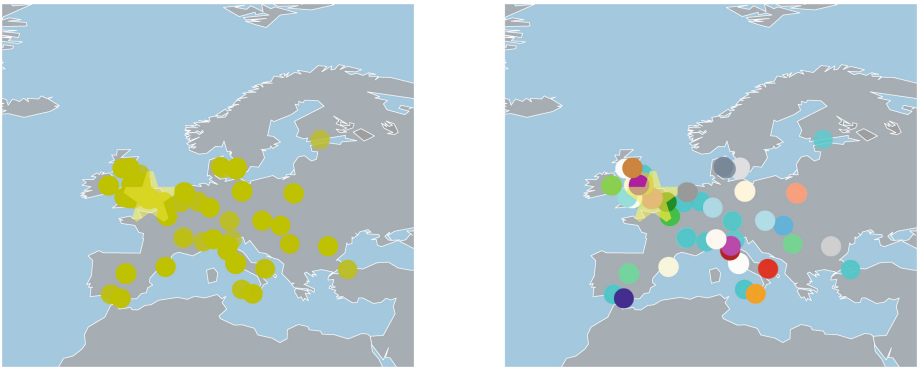
accounts still available removing the suspended and deleted ones. For each Twitter user we then collected her timeline up to 3,200 according to the Twitter API restriction.

4.2 Experiment Results

We performed the GT query search for all 2,880 users according to the nouns extracted from their timelines. For each user, the performed queries resulted in a city distribution from which we sampled random data respecting the GT queries distribution. Then, the iterative procedure described in Sect. 3 is performed for GMM, with maximum number of clusters within the set {5,10,20,30,50}, and for the DBSCAN algorithm, until a unique location (pair of latitude and longitude) is extracted. Table 1 reports the MAE and MdAE values, expressed in kilometres (km), computed between the user home location estimated with different configurations of GMM and DBSCAN with respect to the ground truth. Moreover, Table 1 also shows the MAE and MdAE values obtained by the baseline method. From the results in Table 1 it is possible to notice two main aspects. The first one is that the best configuration of the GMM is the one with maximum number of cluster equal to 10. The second one is that if the MAE is considered, the GMM ($k = 10$) is the best model, while the best MdAE is achieved by DBSCAN. However, both models (GMM with $k = 10$ and DBSCAN) overperform the baseline for both MAE and MdAE distance measures. In particular, the MdAE of DBSCAN is $1,000$ km less than the MdAE of the baseline, denoting the competitive performance of the proposed approach. We also present an example of the user home location estimation for the GMM ($k = 10$) and the DBSCAN (Fig. 2). Figure 2 reports an example of corrected estimated user; the real user location is London and both GMM (k = 10) and DBSCAN derive the final estimation point with latitude $51.507, 51.500$ and longitude $-0.127, -0.144$ respectively for GMM (k = 10) and DBSCAN.

Table 1. Models results in terms of MAE and MdAE. The bold notation underlines the best models. The k in the GMM corresponds to the maximum number of clusters.

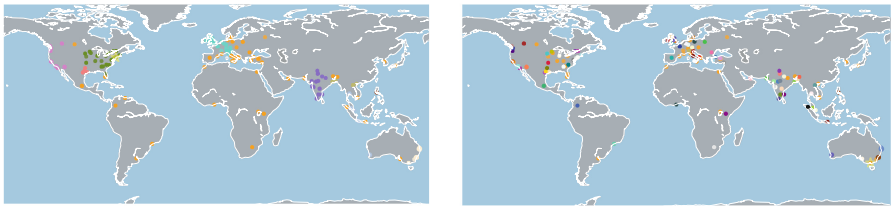
Model	MAE (km)	MdAE (km)
GMM (k = 5)	4582.34	2035.39
GMM (k = 10)	**4359.05**	1759.40
GMM (k = 20)	4796.14	2206.07
GMM (k = 30)	4806.32	2230.69
GMM (k = 50)	5143.23	3503.95
DBSCAN	4484.16	**1737.58**
Baseline	5326.58	2770.89



(a) GMM (k=10) **(b)** DBSCAN

Fig. 2. GMM (k = 10) and DBSCAN home location estimation for User#1. The different colours represent different clusters while the yellow star represents the true user location. In this case User#1 is from London and both DBSCAN and GMM (k = 10) predicted the correct location. (Color figure online)

Moreover, we have analyzed the distribution of the Haversine distance between the true location and the estimated points, revealing that some users are correctly located with an error of just few meters, while for others the estimation error is huge. In particular, we noticed that the largest location estimation errors are frequent among anglophone countries. In particular, Australian users are often estimated as if belonging to the United States. We highlight that this phenomenon, difficulty of anglophone country identification, is well known in the geolocation research field [7]. The estimation errors that involves Australia/New Zealand and United States might be related with the low population density in the Oceania continent, which results in a lower number of Australian cities from GT queries. Regarding this phenomenon, we noticed that DBSCAN performed better in clustering anglophone users. We report an example of this in Fig. 3, where the GMM ($k = 10$) predicted the user home location of User#2 in Chicago, United States while her home location is in Adelaide, Australia. For the same user, the DBSCAN model correctly predict the user home location in Adelaide, Australia. While interesting results were achieved for DBSCAN (in terms of MdAE values), we note that the algorithm was set with the default `sklearn` Python module parameters: ϵ equal to 0.5 and N equal to 5. Nevertheless, a deeper analysis of the DBSCAN hyperparameters will be analyzed in future works.



(a) GMM (k=10) **(b)** DBSCAN

Fig. 3. GMM ($k = 10$) and DBSCAN home location estimation for User#2. The different colours represent different clusters while the yellow star represents the true user location. In this case User#2 is from Adelaide. The GMM ($k = 10$) incorrectly predicted the user home location in United States while the DBSCAN predicted the correct location. (Color figure online)

5 Conclusions

Inferring Twitter user geolocation using only written texts is a challenging task that is fundamental for many social media analytics (e.g., event detection analysis, recommendation systems). Indeed, several works have addressed this task (e.g., [5]), but often these studies focus on a small world region, requiring geotagged labels to train supervised learning methods or rigid geographic dictionaries (LIW). In this work, we propose a novel method to infer Twitter users location considering the world surface using a pure textual based approach. The proposed approach is based on Google Trends (GT) city distributions associated with Twitter user nouns, which are used to fit clustering algorithms, namely GMM and DBSCAN. The aim of this work is to offer a valid and flexible method for users geolocation that can be applied to any language and do not rely on simple LIW. Competitive results were achieved by the clustering methods when compared with a baseline one (GT city mode) that lead the approach to be useful to determine users location for several applications such as marketing campaign and recommendation systems. Nevertheless, further research is needed in order to improve accuracy and reduce anglophone country mismatches (e.g., between Australia and United States of America). In future work, we intend to improve the quality of the proposed geolocation estimation methods by: studying the effect of the tuning of DBSCAN hyperparameters; considering other clustering algorithms (e.g., k-means); and weighting the GT noun city distribution data points with the city population density.

References

1. Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics **49**, 803–821 (1993)
2. Bakerman, J., Pazdernik, K., Wilson, A., Fairchild, G., Bahran, R.: Twitter geolocation: a hybrid approach. ACM Trans. Knowl. Discov. Data (TKDD) **12**(3), 34 (2018). https://doi.org/10.1145/3178112

3. Burton, S.H., Tanner, K.W., Giraud-Carrier, C.G., West, J.H., Barnes, M.D.: "Right time, right place" health communication on Twitter: value and accuracy of location information. J. Med. Internet Res. **14**(6), 156 (2012)
4. Earle, P., Guy, M., Buckmaster, R., Ostrum, C., Horvath, S., Vaughan, A.: OMG earthquake! Can Twitter improve earthquake response? Seismol. Res. Lett. **81**(2), 246–251 (2010)
5. Eisenstein, J., O'Connor, B., Smith, N.A., Xing, E.P.: A latent variable model for geographic lexical variation. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 1277–1287 (2010)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. KDD **96**(34), 226–231 (1996)
7. Han, B., Cook, P., Baldwin, T.: Geolocation prediction in social media data by finding location indicative words. In: Proceedings of COLING 2012, pp. 1045–1062 (2012)
8. Middleton, S.E., Middleton, L., Modafferi, S.: Real-time crisis mapping of natural disasters using social media. IEEE Intell. Syst. **29**(2), 9–17 (2013). https://doi.org/10.1109/MIS.2013.126
9. Schulz, A., Hadjakos, A., Paulheim, H., Nachtwey, J., Möhlhäuser, M.: A multi-indicator approach for geolocalization of Tweets. In: Seventh International AAAI Conference on Weblogs and Social Media (2013)
10. Zubiaga, A., Voss, A., Procter, R., Liakata, M., Wang, B., Tsakalidis, A.: Towards real-time, country-level location classification of worldwide Tweets. IEEE Trans. Knowl. Data Eng. **29**(9), 2053–2066 (2017). https://doi.org/10.1109/TKDE.2017.2698463
11. Zola, P., Cortez, P., Carpita, M.: Twitter user geolocation using web country noun searches. Decis. Support Syst. **120**, 50–59 (2019). https://doi.org/10.1016/j.dss.2019.03.006