Geo-location Prediction on Twitter

Patel Mohammed Farees

Abstract: Micro-blogging web services, such as Tumblr or Twitter, has provided a great platform to exchange, communicate, and access current affairs. Today, it is important to understand the behavior and location of a user in order to provide important services such as geo-aware advertisement, recommendation and event detection.

I am trying to address the problem of geographical location prediction of user based on the textual content of their tweets. More specifically, my prediction is finding the location of a tweet at different granularities such as continents, countries, cities and corresponding location specific GPS coordinates along with their predicted confidence intervals.

In this report, I will be discussing an ensemble based clustering approach which is a combination of three supervised machine learning algorithms. I have used K-means clustering, support vector machines, and cosine similarity to predict with the geolocation of a respective tweet obtained from Twitter streams. My results were found to be promising and performed well on the new streaming tweets.

Patel Mohammed Farees
Syracuse University,
mpatel06@syr.edu

Introduction

The ever increasing use of cell phone devices to access social media, has given the social media platform so much power that data extracted from these platforms is an attractive real-time resource for data analysis and predictive modelling.

People on these platforms post and update statuses, follow and discuss events within a local and global spectrum, comment and share their daily lives with friends, and even publicize individual expression on local political and social trends.

Since these contents are very informative, it can be used for a wide range of applications such as providing location based recommendations (Ye et al., 2010), personalized advertisements, monitoring of fire (Paul et al., 2014), flue-outbreaks (Power et al., 2013) and event detection. Additionally, predicting geolocation of tweets can help disaster response managers to direct necessary resources to effectively coordinate aid. Similarly, by learning user behavior and their dynamic location, advertisers can tailor advertisements for a specific user and location.

Most of the social media platforms provide users with the option to geo-locate their posts but, Jurgens in one of his paper reported that less than 3% of all Twitter posts are geotagged. (Jurgens et al., 2015). This definitely has a very detrimental impact on location-specific applications. Hence, my report investigates and improves the text-based geolocation prediction for Twitter. Text-based geolocation will predict the user's location based on the content of their tweets. My approach will result in predicting user's continent, country, city, and physical coordinates i.e. latitude and longitude pairs based on the content of the tweets.

This report will propose an ensemble clustering approach, a 3-step technique for predicting the fine-grained location of tweets. In clustering step, similar country-specific tweets are grouped in a cluster. This will help us transform a multi-target prediction task into a multi-classification task. In classification step, a corpus of tweets belonging to each cluster is provided along with its clustering assignment as a class variable. This will help us deal with the country-level prediction of a user's tweet. Once we have the prediction of a country, inferring the continents is a simple task. The next step is focused on predicting the physical coordinates of a tweet. And lastly, we leverage the concept of cosine similarity for finding the top five similar tweets for a given tweet. Lastly, the average of geocodes of the five

most similar tweets to a test tweet will be the geolocation of the given tweet. The result demonstrates that my approach leads to promising results in predicting the geo-location of the tweets. The remainder of the paper will describe the approach and results in detail.

Prior Work

For a better comparability of my approach, I focus on the content-based geo-location prediction approaches described in many papers.

Eisenstein et al (2010) focusses on combining user's tweets into a big corpus that will represent a document which can be used to predict a user's geolocation. He proposed a generative model that aims at predicting the user's geographical location based solely on the content of the tweets. The lexical model is based on two sources of variation -- topic and geographical regions. The model uses the correlations between global and local topics. In fact, the local topics are derived from the global topics. The idea is to analyze these lexical variations by topics and corresponding geographic locations, and their interaction is influenced by regional indicators that are associated to location indicative words. (Bo and Baldwin, 2012; Han et al, 2014). Finally, based on the lexical interactions, the model divides the geographical space into groups based on topics and infers the geolocation of a user.

Another approach by Wing and Baldridge (2011) focused on constructing a discrete grid representation of the earth's surface to automatically identify the location of a document based only on its text. Their approach finds a grid cell whose word distribution has the smallest KL (Kullback Leibler) divergence (Zhai and Lafferty, 2001). In each grid cell, the model calculates a word distribution. Then the similarity between a test document's words and that of each cell is computed using Naive Bayes. The predicted location is the center of the most similar cell.

In my paper, I have taken the above work into consideration and focused on predicting geolocation by developing an ensemble clustering approach. Unlike many previous approaches that implemented different kinds of language models, I have used supervised machine learning algorithms approach that includes K-means clustering (K-means) and support vector machines (SVM) along with similarity measure such as cosine similarity to effectively predict the geographical location of users based on their textual tweets.

Proposed Approach / Algorithm

The geolocation prediction that I want to address here is to predict a user's geolocation based on the textual content of their tweets. But instead of exploiting every tweet, I decided to combine all user's tweets corresponding to a country into one single representative document. Then this document is used as the input and the users' country as the output of the system. I have "m" and "v" documents in the training and test data respectively. Moreover, we have n features or tokens describing each document. Each document is represented with a country and its corresponding geolocation. Finally, we estimate the error by calculating the Euclidean distance between the actual and predicted geolocation.

$$\sum_{i=1}^{v} dist(f(X_i^{test}), Y_i^{test})$$

where,

X^{train}, X^{test} : *Training and testing set is of the size* $\mathrm{R}^{m \times n}, \mathrm{R}^{v \times n}$

Y^{train}, Y^{test} : *Training and testing class label (country) of the size* $\mathrm{R}^{m \times 1}, \mathrm{R}^{v \times 1}$

My approach directly captures user's appearance by clustering their geographical location. From the observation of user's geographical appearance, we can learn from the training samples to identify a geolocation distribution. Then for a new data point, we can estimate to which location it belongs. Finally, we can estimate the geolocation by finding the most similar training data points to that new data point in the same cluster.

My analysis started with extracting real-time tweets through the Twitter streaming API. After having spent one week extracting tweets, I aggregated around one million tweets. Secondly, the textual content of the tweets is transformed to obtain a clean version of the text. I have gathered tweets from across the world, these tweets may comprise of different languages and hence for applying any transformation to the textual content, we need to be cautious about dealing with multi-language data. These transformations include removing user handle, punctuations, emoji and smileys. Further the stop words are removed from the corpus, textual content is converted to a lower case and derived words are reduced to their word stem (base or root).

Secondly I divide 80% of my data into training and the rest 20% into testing set. All the tweets corresponding to a specific country are first

combined into one corpus or a document. Since we are working with textual data, it is important to convert the textual content into numeric vectors that can be used for further analysis. Later documents are converted from sparse vectors of tokens into sparse vectors of bag-of-words representation with term frequency - inverse document frequency (TF-IDF) weights (Sparck Jones, 1972; Robertson, 2004). In this way, we discard language grammar structure, token's order, and part-of-speech. The basic idea behind performing this transformation is that the frequency with which a token appears in a document could indicate the extent that the document pertains to that token. The TF-IDF score reflects how important a token is to a document. The more common a token is to many documents; the more penalization it gets. The training (X^{train}) and test (X^{test}) sets' TF-IDF weights are computed separately to their own tokens' scores. (Sparck Jones, 1972; Robertson, 2004).

The tweets in the testing set is not combined based on countries. Tweets in the testing set are individual tweets and are predicted individually. Now once we are done performing the data pre-processing on the training and testing set, we can follow the below approach for test prediction.

The approach can be explained in three following steps:

K-means clustering

The basic idea in the first step is to transform a multi-target prediction (y_lat, y_long) task into a multi-class (y_country) classification task. In order to find regions of interest, we cluster the documents in the training set using K-means clustering that dynamically captures the geographical location distribution. At the end of this step, we have a cluster assignment vector c,

$$\mathbf{c} \in \{1, \ldots, K\}^m$$

where "m" is the number of countries and the i^{th} element c_i contains the cluster assigned to the i^{th} instance in a training set i.e. a country. This means that each country will have its own cluster. Later, the number of clusters (K) can be chosen optimally through cross validation of the training set.

Support vector machines

Now that we have identified clusters, we need to learn a model on X^{train} and c in order to map the test instances to those clusters. For that reason, we use a classifier which has c as the target and X^{train} as the predictors domain.

From now on, the task of geolocation prediction can be treated as a multi-class classification problem. The SVM with L2 regularization (Joachims, 1998) is trained on the dataset associated with corresponding clusters c. The SVM is chosen because it is one of the state-of-the-art algorithms for text classification.

Cosine Similarities

Once we have estimated to which cluster c_i a test instance X_i^{test} should belong, there are a couple of strategies for predicting the geolocation. Now once we have predicted the country, our next objective is to find the city and the geolocation of that tweet.

For predicting the city, we go back to the original training set that has all the individual tweets of various countries and then extract tweets pertaining to a specific cluster (country). Now, we calculate the cosine similarity between the test tweet and all those tweets that are extracted from the training set that belongs to the predicted country. Finally, the city label of the training tweet that is most similar to the test tweet gets assigned to the test tweet.

Similarly, the geolocation can be predicted by either taking the mean of geolocation of all tweets pertaining to a predicted city and specific cluster (country) or taking the median of all tweets pertaining to a predicted city and specific cluster (country).

Those aforementioned steps yield the pseudocode (see algorithm 1).

Algorithm 1: The Ensemble Clustering algorithm

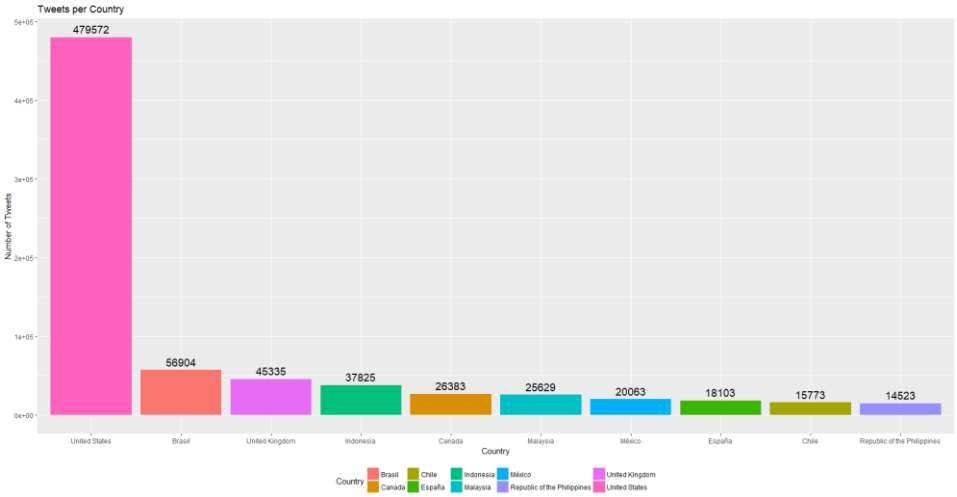
INPUT: X_o^{train}, X^{train}, X^{test}, Y^{train}, cost s, number of clusters K

{**Step 1**: K-means clustering}
c ← K-means(Y^{train}; K)
{**Step 2**: SVM}
g ← SVM(X^{train}; s; c)
$Y_{country}$ ← g
$Y_{continent}$ ← transformations.cn_to_ccn($Y_{country}$)
{**Step 3**: Cosine similarity}
for i = 1 . . . v do
    c_i ← $g(X_i^{test})$
    Y_{city} , Y_{lat} , Y_{long} ← Cosine.Similarity(X_i^{test} , {(X_o^{train}) | $g(X_j^{train}) = c_i$ })
end
return $\hat{y}_i$

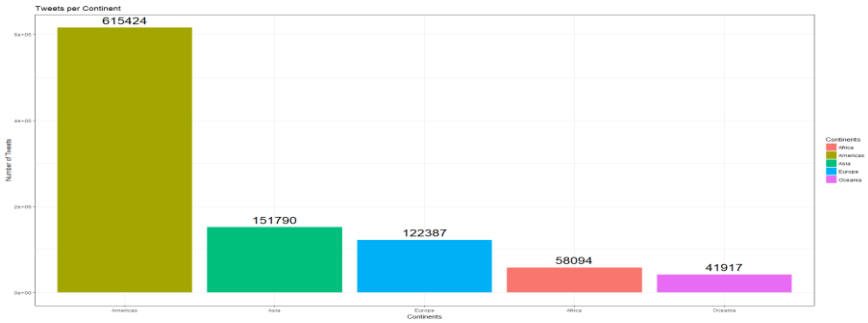
Results and Findings

In this section I will discuss how I calculated the value of K using cross validation and showcase some basic statistics about the data. Further, I discuss an evaluation metric to measure how good is the method described in this report.

Our dataset has one million rows and below is the distribution of tweets based on continents and countries. I have displayed the top ten countries based on their volume of tweets. We can see approximately, 50% of the total tweets comes from the US. Similarly, the Americas continent contribute 50% to the overall volume of tweets. From this observation, we can infer that users tend to appear in a handful geographical locations. This is also observed and illustrated by Hong et al (2012) in one of his paper.



(a) Fig: Tweets per Country

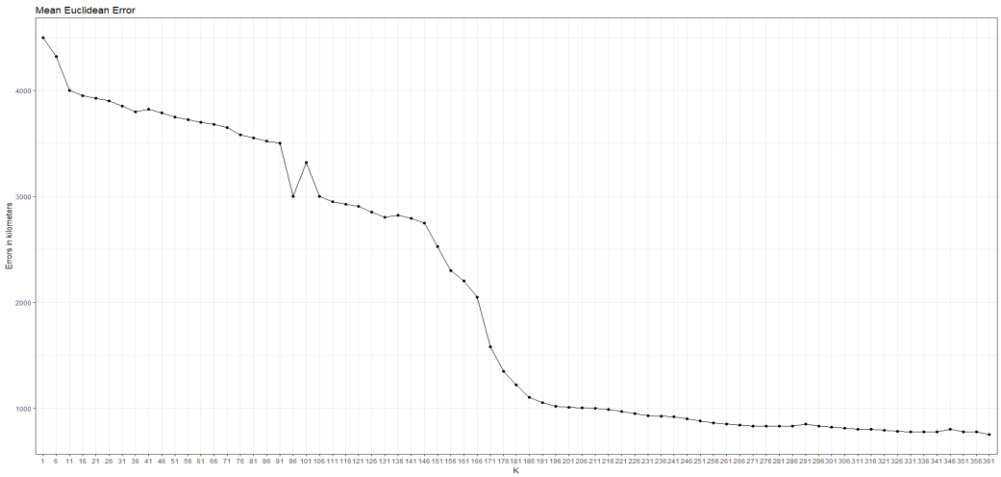


(b) Fig: Tweets per Continent

The evaluation metric that I used to evaluate my algorithm is based on Euclidean distance. Given two points on the earth's surface represented by their latitude and longitude coordinates, the easiest way to calculate the physical distance between them is the Euclidean distance. That means we treat the earth's surface as a flat plane in two dimensional space. Hong et al (2012) report this metric in their work

$$d_E(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{(\hat{y}^{lat} - y^{lat})^2 + (\hat{y}^{lon} - y^{lon})^2}$$

I ran a 10-fold cross validation and defined a grid for the value K (Number of clusters). Running cross validation over the grid. I got the following result:



(c) Fig: Cross-validation Mean Squared Error

We can see when K =201, the mean squared error drop to its minimum i.e. 860. At this value of K, my model performed at its best. Below are the results of my model.

Prediction variable	Accuracy rate
$Y_{continent}$	87%
$Y_{country}$	64%
Y_{city}	48%
Y_{lat} , Y_{long}	31%

The results of my analysis have proved that it is effective to predict the user's geolocation by directly exploiting the location distribution instead of developing complex language models. The performance of my model is very close to the

performance of some intricate lexical models proposed by Eisenstein et al (2011) and Hong et al (2011) . Another advantage of the K-means clustering is that it takes less computation time because the number of clusters is much smaller and it dynamically adapts to the location distribution despite of which dataset is evaluated.

Conclusion

In this report, I have introduced an easy-to-implement clustering based ensemble approach to deal with the geolocation prediction in Twitter streams. I have chosen a method that does not rely on complex lexical or language models. Instead, at first it transforms the target space to learn a clustering assignment. Secondly, it learns to map the training set to the respective clustering assignments. And finally, it makes prediction by leveraging the cosine similarity. The approach is experimentally straightforward and effective. My analysis on one million tweets and its geolocation task shows that the proposed approach is effective and accurate at predicting continent and country with corresponding confidence interval.

For future work, I plan to make further experiments to improve my approach. There are some considerations worth analyzing. At first, I would like to conduct an all-in-one model for geolocation prediction in general. It means that instead of combining different separated algorithms, we can build a single model that captures the users' geolocation.

References

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. the Journal of machine Learning research 3:993–1022

Bo H, Baldwin PCT (2012) Geolocation prediction in social media data by finding location indicative words. Proceedings of COLING 2012: Technical Papers pp 1045–1062

Chandra S, Khan L, Muhaya FB (2011) Estimating twitter user location us-ing social interactions–a content based approach. In: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on So-cial Computing (SocialCom), 2011 IEEE Third International Conference on, IEEE, pp 838–843

Decker BL (1986) World geodetic system 1984. Tech. rep., DTIC Document Eisenstein J, O'Connor B, Smith NA, Xing EP (2010) A latent variable model for geographic lexical variation. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Com-putational Linguistics, pp 1277–1287