K Center Clustering Implementation and Evaluation With Regards to Identifying User Location Based on Tweet History

Roberto Hong Xu Kuang
Computer Science
California State Polytechnic
University, Pomona
rxu@cpp.edu

Abstract

By using an unsupervised of machine learning, we attempt to identify the location of a user based on the most recent history of the users Tweets. First, by connecting directly to the Twitter API, we are able to collect the data that we need. Then, using a natural language processing tool, the text of each tweet is passed to a gazetteer to identify a location that the tweet is referring to. If the location of a user can be successfully re-trieved in this way, then it opens up the possibilities where advertisements and further data collection can be better suited for each user. This paper will show how to reliably collect and cluster the locations of tweets using an unsupervised learning method known as K-Center clustering. Results will also show how plausible this method can be by comparing the predicted location to the location found in the User profile.

1 Introduction

The amount of content being generated online by users is increasing at an exponential rate as time goes on. As more and more users flock to web services like Facebook, Reddit, and Twitter, it is crucial to develop a method to properly gather data without supervision. With user generated content, it is very common for them to post information online with regards to their current location, status, or even future events. By just being able to accurately measure the location, it becomes a very valuable tool when it comes to marketing and advertising to the consumer (Beckland, 2011). In fact, many applications today rely on location in order to be popular, such as Yelp and FourSquare (Leiteritz, 2010).

Currently, besides requesting the user to grant access to their current location, businesses have to rely on using IP addresses to get the location of the user. While this has been the standard for very long, it is almost impossible to do that now, with VPNs and dynamic IPs being used to mask the IPs of users (How do I Determine...,).

Twitter currently has two possible ways to share the location of a user: geotagging the location of the user for each tweet and the description found on each user profile. However, geotag is disabled unless the user opts to use the service (FAQs about adding location...,). This, along with the fact that many users do not put their most accurate location into their description or location field on the profile, limits the available information to gather from users.

We attempt to solve this problem by identifying the location of the user based on just references made in their tweets.This is done by extracting the locations found in the tweets. We must assume that the user will post places related to where he or she lives.

The rest of this paper will cover just how to properly extract and parse the tweetsof a particular user. Then, it will be explained on how to use a gazetterr, the geographical dictionary. Once the coordinates are calculated, the K-Center clusering algorithm will be explained in detail.

2 Related Work

As Twitter is becoming such a gold mine for data, it is not surprising that the idea of getting information of users based on tweets is not novel. There have been other experiments conducted where the location has been successfully extracted and compared to (Katragadda, Jin, and Raghavan, 2014).

2.1 Natural Language Processing

There are many ways of parsing the data and the way that I have chosen it is using the Ark-NLP that was developed by students of Carnegie Mellon University and Toyota Technological Institute

in Chicago (Owupoti et al, 2013). Ark-NLP was created specifically for parsing data from Tweets, and in the paper they talk about how they created a part of speech tagger along with a tokenizer. This is extremely useful because they have collected data from Twitter and trained the data in order to properly parse data, running about 800/tweets a second (Owupoti et al, 2013).

2.2 K-Center Clustering

K Center clustering has been researched and explained by several authors already. It has been said that K Center is NP-Hard, meaning that there is no efficient way to get the correct answer (Dasgupta]2013,). However, there is an algorithm developed that will be explained in Section 3 that comes close.

3 Experiment

This section will cover several subsections. First will be the collection of data and then properly parsing the data. Second will be the description of the K-Center Algorithm. Last will be the results and explanation of the output.

3.1 Data Collection

In order to access the data of an individual user, we must use the Twitter API. The API is the RESTful kind, so an internet connection is required to properly retrieve data of any kind. Due to the rate limit calls of the API, it is very time consuming to collect a large amount of data at a time. Therefore, the solution used was to only gather data of just one individual at a time, while storing the data after being used.

Once the tweets have been retrieved, it is necessary to use a natural language processor to tag the tweet. Using ARK-NLP, this becomes a trivial task. Once the tweet has been completely tagged, we must distinguish just what data is needed and what can be tossed away.

Due to the fact that we only require names of locations, we can discard everything besides nouns, and we can form what is called an n-gram. Once a whole list of n-grams are collected, the next step will be to run them through a gazetteer.

3.2 Gazetteer

A gazetteer is a geographical dictionary that contains a list of names and their proper coordinates given in latitude and longitude.For this paper, the online gazetteer called DBPedia is used.

DBPedia is an open source data collection service that automatically grabs data from WikiPedia, and returns it in an RDF format. While there are other gazetteers available, such as the U.S. Census, DBPedia has an online endpoint that can be used to query data. This is ideal because it allows users to use their service without managing a server or database.

By using the n-grams generated from the NLP process, the locations are sent to DBPedia and we expect a latitude and longitude in return. This allows us to plot every single location in a euclidean space.

3.3 Distance

Every single coordinate has a latitude and longitude, and this is plotted on a spherical surface, much like the Earth. However, this means that in order to get the distance between two points, it is not as simple as getting the euclidean distance.

Due to the fact that the Earth is spherical, a different formula is required. There are two main formulas. One is the haversine formula. The other is the spherical law of cosines.

The haversine formula is a bit more complicated than the other, due to the fact that it was invented when most computer data was in float format, giving less precision. With the Spherical law of Cosines, it takes advantage of the fact that there is now 64 byte double precision, so it is easier to implement.

1. Given R radius of earth in either meters or miles.

2. Convert latitude and longitude of both points from degrees to radians.

3. lat1 = latitude of first point, lat2 = latitude of second point

dLat = differene between two latitudes

dLong = difference between two longitudes

3a. Haversine:

a = sin(dLat/2) * sin(dLat/2) + cos(lat1) * cos(lat2) * sin(dLong/2) * sin(dLong/2)

c = 2 * asin(square root of (a))

3b. Spherical Law of Cosines:

a = sin(lat1) * sin(lat2) + cos(lat1) * cos(lat2) * cos(dLat)

c = acos(a)

4. Total distance = c * R

3.4 K-Center Clustering

We first consider location identification has a clustering problem where the goal is to cluster all ge-

olocations together so that the largest cluster will be designed as the home cluster. Then, the center of that cluster will be calculated and returned as the predicted home of the user.

Similar to the K-Means clustering algorith, this K-Center has many of the same benefits where it is robust, but also requires a set cluster before being used.

Now, while there are many ways of clustering for K Center, the most well known and efficient one seems to be one called "Farthest First". This means that after the first point is selected, the farthest point is considered to be the center of the next cluster. The following is the psuedo code that will go over the K Center Clustering.

If there is a group of points P, we need to find K points from P and the smallest radius R where every single P is within the radius of at least one K center. To do this:

1. Randomly pick a geolocation x to be assigned as the first cluster center.

2. Get the point with farthest distance from the center of all clusters.

3. Repeat for K points.

4. Calculate the radius where every single point is within the radius of the center of a cluster.

5. Assign every geolocation within the R distance to the each cluster.

6. Predict the center of the cluster with the most amount of geolocations as the predicted home of the user.

3.5 Results

Once the predicted location is calculated, it needs to be compared to see just how accurate the location is. However, this is very hard to do so due to the fact that it is rare for the geotag feature to be enabled for a user. This means that there is no real concrete location from a user.

The only possible solution that was created was to manually compare the location from the description or location field from the profile. However, this is stil not the best solution as there are profiles with completely no information.

Judging the results of several users, it has become apparent that the accuracy is very unpredictable currently. This is due to several factors.

1. The gazetteer is not as accurate as hoped. There are many acronyms in the ngram data and using the gazetteer to look up an acronym did not seem to work out. Also, ther are sometimes multiple locations with the same ngram in their name, and it is hard to pick the correct location.

2. The first cluster is picked randomly, so it is rare for each test to get the same clusters. This means that the test must be ran multiple times, where sometimes it is accurate within 200 miles, and sometimes it is off by 2000 miles.

4 Conclusion

This paper is a small step in correctly geolocating a user based on their generated content. While the results are not very satisfactory in terms of accuracy, there is a lot of basic ground work that has been started, and there is a great possibility that this work can be used as a stepping stone to another experiment.

With more time and research, the algorithms and data collection methods can be refined and the results will become more stable. However, as long as Twitter does not force geotagging, there will still never be a concrete location to be defined as the home of a user. No matter how stable the predicted home location is, it will still require manual comparison. If this was changed, then it would be much easier to test accuracy.

In the future, it is hopeful that by learning more about the NLP and gazetteer, data can be parsed more accurately, thus allowing more accurate results. There are also many ways of doing a K-Center clustering, and there are countless other ways of getting the location of a user. By testing all of these algorithms in the future, the predicted location of the user can become as accurate as possible.

5 Appendix

This project was meant to be a two person project, however my partner decided to abandon the class and project. We had split up the work evenly, but the term was more than half way over before my partner told me the news. Therefore, much of the code and data is very inefficiently written and rushed. Originally, my partner was to handle the parsing of the data, including using the gazetteer. This is the number one problem of the data of the project currently. Aside from that, my partner was also supposed to implement her own algorithm, and this would have allowed me to create a comparison of accuracies.

While I also had my own part, due to the fact that I had to take on the rest of my partner's duties,

everything became rushed.

5.1 GUI

A graphical user interface was created for this project. It is very simple to use.

The top of the GUI features a running counter of the current rate limit for the Twitter API. As each button is pressed, it will update this counter appropriately.

The user must enter a twitter username in the text area, and hit the submit button.

This will cause the program to load a little bit, in order to query the Twitter API and create the appropriate files. After this is done, the two text areas will load the appropriate data. The left will show the raw data plus it's tag from the NLP. The rigt area will show the ngrams that was created by stripping the rest of the tweets.

After that, the user can hit the "Next" button, which will run the gazetteer. However, this will cause the program to stall for a very long time. For some reason, querying DBPedia takes a very long time, and there is some sort of hidden rate limit. However, once it is done loading, it will fill the text areas with content.

The left area will now show the locations of all the ngrams, and the right area will show 2 things. The first line shows the predicted location of the user. The next entries will be the possible locations found within a 0.05 latitude and longitude radius.

The bottom of the GUI will show the user's description and location field. This is used for manual comparison of the accuracy of predicted location.

5.2 Steps to Run Code

In order to run the project:

1. Compile all java files to class files.

2. Be sure to include the jars located in the dependencies folder.

3. The file 'model.20120919' is required for ARK-NLP, so put it in the same folder as the project.

4. The main function is located in TwitterRest.java.

References

How do I determine the physical location of an IP address? . http://www.computerhope.com/issues/ch001044.htm

FAQs about adding location to your Tweets . https://support.twitter.com/articles/78525-faqs-about-adding-location-to-your-tweet

Jamie Beckland. 2011. *How to Use Geolocation in your Marketing Initiatives* . http://www.socialmediaexaminer.com/how-to-use-geolocation-in-your-marketing-initi

Raphael Leiteritz. 2010 *The Importance of Geolocation Services*, http://google-latlong.blogspot.com/2010/04/importance-of-geolocation-services.html

O. Owoputi, B O'Connor, C. Dyer, K. Gimpel, N. Schneider, N. Smith 2013 *Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters* http://www.ark.cs.cmu.edu/TweetNLP/owoputi+etal.naacl13.pdf

Sanjoy Dasgupta 2013 *Clustering in metric spaces* http://cseweb.ucsd.edu/~dasgupta/291-geom/kcenter.pdf

Satya Katragadda, Miao Jin, Vijay Raghavan 2014 *An Unsupervised Approach to Identify Location based on the Content of User's Tweet History* http://www.cacs.louisiana.edu/~mjin/publication/AMT14_LocationAnalysis.pdf