2014 IEEE International Conference on Data Mining Workshop

Estimating online user location distribution without GPS location

Yusheng Xie Yu Cheng Ankit Agrawal Alok Choudhary

EECS Department, Northwestern University
2145 Sheridan Rd., Evanston, IL USA

{yxi389,ych133,ankitag,choudhar}@eecs.northwestern.edu

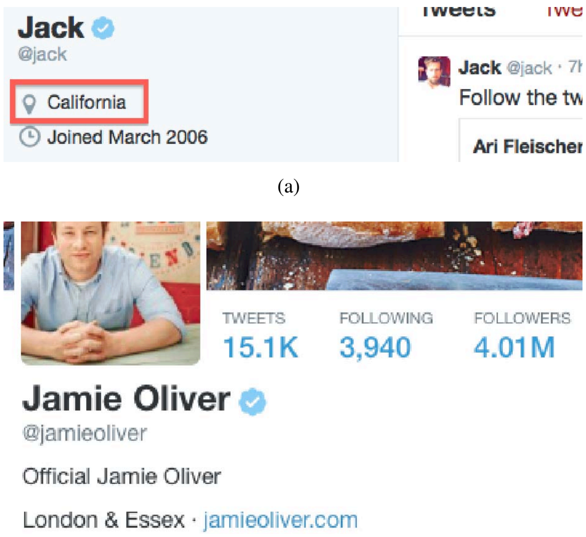
***Abstract*—We focus on the problem of offline user location estimation using online information, particularly for the application of TV segment advertising. Unlike previous works, the proposed method does not assume GPS information, but works with loosely structured information such as English location description. We propose to use a neural language model to capture the semantic similarity among the location descriptions. The language model can help reduce the otherwise expensive geolocating service lookups by internally resolving similar areas, neighborhoods, etc. onto the same description. We also propose a metric for comparing geodemographic histograms. This metric considers the demographic gap between the online world and the offline world. In the experiments section, we demonstrate the recall and accuracy of our language-based, GPS-free user location distribution estimation. In addition, we illustrate the effectiveness of the proposed distribution estimation metric.**

I. Introduction & Related Works

Geo-location is one of a few ties that bridge the online world and the offline world. Twitter is among the most popular social networks where users share things happening near them. In general, a user has two options to specify geolocation on Twitter:

- Type in the location information (not a drop down menu) that associates with his/her user account. For example, in Figure 1, Jack Dorsey specifies *California* and Jamie Oliver specifies *London & Essex*. Because user can type in anything, fictitious or nonsensical locations are quite common. For example, *Neverland* or *Hogwarts*, we find out, are among the most popular addresses.
- Attach GPS coordinates to the tweets a user sends out. When the user sends a tweet from a smart phone, s/he can choose to do this for each tweet. Roughly 1% Twitter users (estimated in our dataset) will enable GPS on their tweets. On the other hand, over 20% Twitter users will enter a location description[1].

If our goal is to estimate Twitter adoption by region (i.e., Twitter demographics), the GPS-tagged tweets cannot be directly aggregated statistically. The goal is really to assign to primary location/region to each Twitter account instead of each tweet. And their sparse availability makes it even more difficult.



(a)

(b)

Fig. 1. Example of how Twitter users can specify geo locations at account level. (a) Jack Dorsey simply entered *California* (b) Jamie Oliver entered two locations.

Previously, researchers[1], [2] use GPS location data from social networks (Twitter, Instagram) to predict user's future locations. In [2], the authors use friendship ties (i.e., following/follower status in Twitter) to predict realtime geographical affinity and reconstructs missing friendship ties from realtime geographical affinity. The core assumption is that real time location is a strong offline signal that correlates to online friendship. This assumption itself is quite reasonable. The easiest way to argue for it is the *Friday nights scenario*. On Friday nights, friends would go to dinners and clubs together and tweet about the gathering from the same location. Because they are real world friends who hang out together, they are very likely to follow each other on Twitter as an online extension of their offline friendship. This idea from [2] is simple and works well. The similar concept is taken to further refinement in [1], where the authors propose the distinction between *hometown* and *travel*. They argue that the real time location prediction should be handled differently when it comes to movements within one's hometown region versus long distance

[1] http://www.businessinsider.com/twitter-user-statistics-2014-4

978-1-4799-4274-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDMW.2014.30

IEEE computer society

travels. It is intuitive that the previously mentioned *Friday nights scenario* is more applicable to hometown location predictions but less so for long distance travel predictions. If one travels long distance regularly (e.g., a binational), s/he is very likely to have different social circles with basically no overlap. Recognizing such cases, the model in [1] uses dynamic Bayesian networks to allow flexibility.

Realtime location prediction from online social ties and offline GPS information is an interesting problem, but not the one we are solving in this study. One of the great strengths of online social networks is their ability to modernize existing laborious offline pipeline processes. The prominent example of this aspect is Google's work[3] on using search queries to track influenza epidemics in both temporal and spatial dimensions (the up to date results are published at http://www.google.org/flutrends/). Using large volumes of user search queries, Google is able to predict flu trends faster and more accurately than United States Centers for Disease Control and Prevention (CDC). There is no surprise that Google performs better than CDC on that front. Spatial data collection from an offline pipeline introduces stages of delays. Typically, CDC has regional outposts throughout the country, each of which is responsible for collecting flu data from the local hospitals and reporting back to CDC. Having gathered all regional data points, CDC finally performs analysis and prediction on the flu trends. This pipeline process is slower and probably less accurate than what Google performs with its search logs.

Another laborious offline pipeline process (with great business interests) is media audience measurement (MAM). MAM measures the number of people in each geographical region's audience in relation to media consumers (e.g., radio listenership, television viewership, newspaper/magazine readership). MAM is critical for offline advertising for two reason. First MAM covers the biggest offline advertising channels: TV and print. Second, advertisers rely on MAM information to make ad inventory purchase decisions. For example, Porsche (luxury, sports carmaker) often runs double-page advertisements on medicine magazines to target affluent medical personnel; Burger King (24 hour fast food chain) likes to purchase ad inventory from late night TV shows (presumably for attracting hungry customers who stay up late and can't find an open restaurant at late hours). Both TV and print are planned, high-reach, periodic media, so committing to purchase their expensive ad inventory requires careful, cost-aware decision making. Ad buyer uses MAM to make the decision of ad buying. Without loss of generality, we focus on TV ad buying using TV MAM.

TV MAM has existed long before Internet, so it is no surprise that the existing offline legacy solution is blamed for being slow and inaccurate in this day and age. The most widely and currently used TV MAM in North America are Nielsen ratings developed by the Nielsen company. Nielsen's TV ratings were developed in 1950s using methods originally intended for radio ratings in 1930s. The technology uses Set Meters, which are small physical devices. These devices are mailed to individual households and installed there; they communicate with Nielsen nightly through telephone line and report back to Nielsen the household's viewing habits (e.g., time series of channel switches). In the year 2013, there are around 31,000 such devices (or equivalent)[2] throughout the US while it is estimated that US has over 115 million TV households[3].

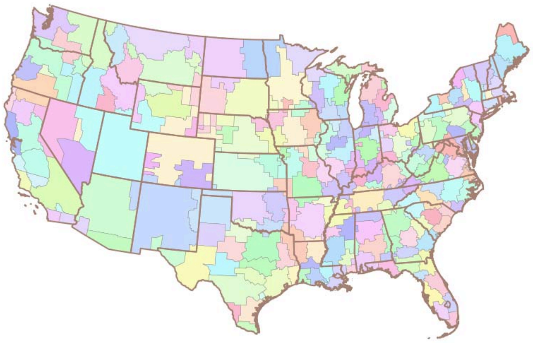


Fig. 2. Map of DMA topology in continental US

The demographics interesting to TV ad buyers include gender, age, income, race and region. Region, being the least sensitive information, is often the most widely and reliably collected demographics. In this study, we will focus on TV region demographics (or simply geodemographics).

Geodemographics is a staple in national census. A census would use different topologies ranging from granular to coarse In United States, often used topologies (in the decreasing order of granularity) are zip codes, cities, counties, and states. But TV media market uses a special topology called designated market area (DMA). DMA regions are initially defined based on the reachability of the signals from major TV or radio stations. As of 2013, there are 210 DMA regions. In terms of topological granularity, DMA falls in between counties and states. Figure 2 shows the DMA topology over state boundaries on a US map (lower 48 states)[4].

It is difficult to estimate the overall offline geodemographic information from Twitter because of three reasons.

- First, the 1% Twitter users who enable geo tagging are not proportionately distributed to the actual popular demographics or the Twitter user demographics. To illustrate this problem, suppose that New York City has a population of 8 million and further suppose 2 million of the New Yorkers have an active Twitter account. In contrast, suppose Minneapolis has a population of 400 thousand, of whom 50 thousand have an active Twitter

[2] http://www.nielsen.com/us/en/insights/news/2012/celebrating-25-years-of-the-nielsen-people-meter.html

[3] http://variety.com/2013/tv/news/nielsen-us-household-number-estimate-increase-1200471668/

[4] Picture originally from Wikipedia http://en.wikipedia.org/wiki/Media_market

account. Doing so we simply assume that New York City has a higher population as well as a higher per capita active Twitter account. We need a mechanism to adjust for this gap between online and offline populations.

- Second, the disproportionate distributions introduce inconsistent statistic confidence values in the estimated demographics among different regions. For example, we may observe 10 thousand Twitter users from New York in our sample and only 30 Twitter users in a small city (e.g., Pullman, WA). Smaller sample sizes in less populous regions will result in low confidence and high variance in their Twitter demographic estimations.
- Third, tourist attractions usually have a lot of tweets with GPS enabled, most of which are sent by their visitors but not their residents. This situation also makes the demographic estimation harder as it introduces more variance to the already sparse GPS information.

II. Problem Formulation

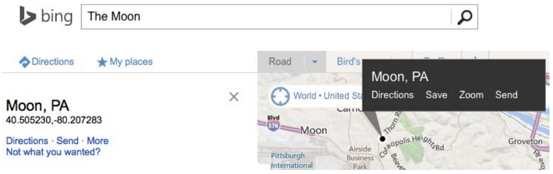
We assume several sets of Twitter users. Each set of Twitter users correspond to a demographic we are interested in (e.g., it can be the group of Twitter users who are interested in a TV show or an actor). The first part of the problem is to reliably resolve the demographics from just profile information without GPS into histogram distributions. The second part is to establish meaningful metrics for cross-comparing the demographics.

III. Resolve description to latitude, longitude

The ideal situation is to use the average GPS coordinates embedded in user's tweets to estimate this user's residential location. But less 1% Twitter users (estimated in our dataset) will enable geo tagging on their tweets and the percentage is even lower for the users who enough record geo tags for the estimated location to be robust. For such reason, we decide to develop a methodology that does not rely on embedded GPS coordinates.

The location description string, on the other hand, is a far more commonly adopted on Twitter user base. Over 80% of the users in our dataset specify some non-empty string in his/her location description. Given a string of location description, the obvious solution for parsing it is to send every string to a cloud resolving service such as the Google Maps Geocoding API[5] or the Microsoft Bing Maps Location Query API[6]. However, there are two problems with this solution.

- Such services from Google or Microsoft has an imposing business cost. Both services require business account subscription for sustained use of these API calls. It takes considerable cost and time (due to API throttling) to process 10s of millions of location strings.
- Such services assume well formed and sensible location string queries. They take the string input quite seriously, which is not how many Twitter users have in mind when they put in the location string. For example, some humorous Twitter users enter "The Moon" as their locations. And Figure 3 is what happens when we tell Bing to find us where "The Moon" is on its map. The embarrassment can neither be blamed on Twitter user's attitude nor Bing's algorithm; it is simply due to different expectations between the one who enters the text and the one who parses it.



The Moon

Moon, PA
40.505230,-80.207283

Fig. 3. Example of failed Twitter location string look up.

To address the above two weaknesses, we have designed a combination of techniques. First, we ran an experiment to see the histogram distribution of the most frequent locations strings entered by Twitter users. Initially, we thought that the location distribution would be somewhat power law like (i.e., several hundred top locations would cover the majority of the user base), but this assumption turned out to be quite inaccurate. We find out that the location distribution is actually very long-tailed. The top 100 thousand most frequent locations string cover about 10% of total users in our dataset.

A. Neural language models

To cope with the long tail of location strings, we resort to the semantics embedded in the the location descriptions. For example, the semantic difference between *Boston* and *Massachusetts* is similar to the difference between *Lowell* and *Massachusetts*. So by tracking the semantic differences, we hope to approximate descriptions from the tail distribution (e.g., Lowell, Massachusetts) with descriptions from the head distribution (e.g., Boston, Massachusetts).

The key assumption in this approach is to train a reliable language model that accurately represents the semantics (and syntaxes). The language model we use is a neural language model[4], [5], [6] called "word2vec". word2vec takes a unlabeled text corpus, and produces a high dimensional numeric vector for each word and each bigram. Those vector representations can capture semantics quite well as long as the vector dimension used is high (we use 200 in practice) and the training corpus is sufficient (millions of words for tens of thousands of vocabulary).

word2vec is essentially a single hidden layer neural network. For each word, it is trained on the words around it. The implementation has several tricks (e.g., hierarchical softmax[4], negative sampling[5]) to make its performance practical. In this paper, we use Google's open source implementation[7] of word2vec with only minor modifications.

[5] https://developers.google.com/maps/documentation/geocoding/

[6] http://msdn.microsoft.com/en-us/library/ff701714.aspx

[7] https://code.google.com/p/word2vec/

IV. Mapping Latitude, longitude to DMA topology

Given a set of $\langle \text{latitude}, \text{longitude} \rangle$ coordinates, the goal is to map each of them to one and exactly one of the 210 DMA regions. Mathematically speaking, the problem is equivalent to membership assignment on a first-order Voronoi diagram of 210-compartments (each DMA region being one compartment). There are quite a few fast algorithms for performing this task[7]. In our experiment, we first build a k-d tree[8] K that indexes the $\langle \text{latitude}, \text{longitude} \rangle$ centroids of the 210 DMA regions. Then for each pair p of $\langle \text{latitude}, \text{longitude} \rangle$ coordinates , we query K to find the nearest DMA region to p under the Haversine distance approximation[9]:

$$\text{distance}(\langle \phi_1, \lambda_1 \rangle, \langle \phi_2, \lambda_2 \rangle) = 2r \arcsin\left(\sqrt{\sin^2 \frac{\phi_2 - \phi_1}{2} + \cos\phi_1 \cos\phi_2 \sin^2 \frac{\lambda_2 - \lambda_1}{2}}\right) \tag{1}$$

where r is the average Earth radius, $\langle \phi_1, \lambda_1 \rangle, \langle \phi_2, \lambda_2 \rangle$ are the $\langle \text{latitude}, \text{longitude} \rangle$ coordinates of a DMA region centroid and the estimated coordinates from description string.

For each set of Twitter users, we can resolve the location description from each of its users. This operation transforms a set of Twitter users into a set of DMA region codes. In other words, this set of Twitter users can be visualized as a categorical histogram over all DMA regions. Figure 4 is an example result of this process.

Algorithm 1 describes the conversion of each user set into a histogram distribution. First(line 1), it trains a neural language model based on location text corpus d using word2vec implementation[8]. So l holds a list of the 500d vectors that represent the words appear in d. Since l and dr are both spatial data points (items in l are 500 dimensional vectors in hyperspace and items in dr are 2 dimensional points on surface), we index both l and dr in k-d trees(line 2 & 3). To build a histogram for each set T_i, we need to iterate over all its users in T_i and parse their location description string. We first look up the strings among the frequent locations in ls. If the current user's location description string is not in ls, we try to look up this string in the language model hyperspace and find its nearest location synonym (based on inter-vector L2 distance) that exists in ls (as seen in the while loop starting line 13). We try the top 3 nearest synonyms, if none of them matches any entry in ls, we skip to the next user. Once the algorithm gets a location $\langle \text{latitude}, \text{longitude} \rangle$ from ls, we query its nearest neighbor in dkd, the k-d tree indexing all DMA regions' centroids, and increase the count in the histogram accordingly (line 21 & 22). When searching for nearest neighbors in dkd, we can use the Haversine formula to determine precise distances. And it will work with conventional k-d tree search algorithm because the Havesine distance is, like Lp distances, monotonic in any dimension. In practice, when we perform the algorithm concerning only the lower 48 states in the US, we find that $L2$ distance works quite well as an approximation of the Haversine distance and is much easier to implement.

Algorithm 1: Resolve algorithm

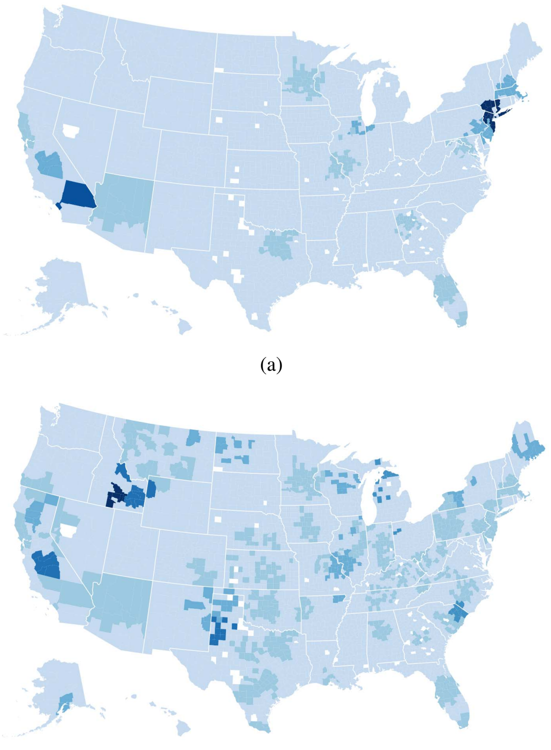
```
Input: $T_1, T_2, \ldots, T_I$, Twitter interest sets; each set is a
       set of user,
   $d$, generated document for training the language model,
   $dr$, DMA region codes with $\langle \text{latitude}, \text{longitude} \rangle$ of each
   region's centroid,
   $ls$, a dictionary where location strings as key and
   $\langle \text{latitude}, \text{longitude} \rangle$ as value, only the most frequently
   used description strings are sent to remote geolocating
   service for resolving $\langle \text{latitude}, \text{longitude} \rangle$.
Output: $h$, list of all $I$ DMA region-level histograms for
        $T_1, T_2, \ldots, T_I$
1  $l \leftarrow$ word2vec($d$,dim=500)
2  $lkd \leftarrow$ new kdtree($l$)
3  $dkd \leftarrow$ new kdtree($dr$)
4  $h \leftarrow$ new list$\langle$histogram$\rangle$()
5  for $i \in \{1, \ldots, I\}$ do
6      $h[i] \leftarrow$new histogram()
7      for $u \in T_i$ do
8          $s \leftarrow u$.locationDescString
9          if $s \in ls.keys()$ then
10             $g \leftarrow ls[s]$
11         else
12             $i \leftarrow 0$
13             while $i < 3$ and
               $t \leftarrow lkd.ithNNsearch(s, i, metric=L2\text{-}norm)$ do
14                 $i \leftarrow i + 1$
15                 if $t \in ls.keys()$ then
16                     $g \leftarrow ls[t]$
17                     break
18                 end
19             end
20         end
21         if $g$ is not null then
22             $dc \leftarrow dkd$.NNsearch($g$,metric=Haversine)
23             increase $h[i][dc]$ count
24         end
25     end
26 end
27 return $h$
```

V. Quantitative comparison of demographics

Our efforts so far have lead to categorical histograms over DMA regions. Histograms shown in Figure 4 can only be visually consumed unless we develop systematic metrics for quantitatively comparing these histograms.

Map comparison on categorical histogram level is an important topic in Geographic Information System (GIS) and has a long history of investigation that can be dated before the silicon age[10]. However, our problem stands out due to a few unique challenges and requires a novel measuring solution.

[8] https://code.google.com/p/word2vec/



(a)

(b)

Fig. 4. Histogram distributions of a set of Twitter users over DMA regions. Deeper color means higher value. (a) Raw counts of Twitter users in each DMA region (b) Counts normalized by each region's population.

- We have a large number of sets of Twitter users. Twitter itself is an interest network. It is conceivable to have 1 million (or more) sets (usually with overlapping memberships) to represent the diversified user interests on Twitter. To answer practical queries like "top N interest sets that have the most/least similar demographics with a given set", fast pairwise comparison of the histograms of those Twitter user sets becomes a requirement. Preferably even better, the proposed metric should be able to incorporate with k-d tree[8] scheme. This would enable relatively fast top-N look-ups without precomputing all pairs of histogram similarities.
- many existing map comparison metrics[10], [11] are binary metrics. It means that the metric will tell you whether the two map distributions are identical or not. We need a continuous measure instead of a binary one.

Existing popular solutions[10], [11] for map histogram comparison are mostly based on statistical measures of inter-rater, categorically-valued agreement (e.g., Cohen's kappa coefficient)[12]. However, when we compare two Twitter interest sets here, the user locations that generate the geographic histograms are not the same users in both sets. As a result, kappa statistic (or similar statistics) cannot apply to compare the histograms in our case.

What other options do we have then? Jensen-Shannon divergence (JSD) is a popular statistical method of measuring the similarity between two probability distributions. JSD is based on the better known Kullback-Leibler (K-L) divergence. Unlike K-L divergence, JSD is symmetrically defined and always returns a positive, finitely bounded value. Without loss of generality, consider two discreet probability mass functions P and Q. And let $M = \frac{P+Q}{2}$. The Jensen-Shannon divergence of P and Q can be defined as:

$$\text{JSD}(P,Q) = \text{JSD}(Q,P) = \sum_i \ln\left(\frac{P(i)}{M(i)}\right) P(i) + \sum_i \ln\left(\frac{Q(i)}{M(i)}\right) Q(i). \tag{2}$$

The Jensen-Shannon Similarity (JSS) is simply 1-JSD. Both JSS and JDD are between 0 and 1. Since the definition of JSS on two distributions is a drop-in replacement for, say, more standard L^p grid distances. It is straightforward to embed JSS as the metric in a k-d tree or a LSH that indexes the histograms of the Twitter interest sets.

A. *Correcting Twitter bias*

The demographics on social network does not represent the actual demographics. For example, 20-24 is the most populous age group among American Facebook users. In fact, there are more Americans in either age group 45-49 or 50-54 than 20-24[9]. It is not surprising for Facebook to have a skewed user demographics biased toward younger population because of its link to the Internet. Other social networks have different kinds of biases towards a particular demographic. Pinterest and Tumblr have noticeably more female users[10][11].

Twitter, like many other social networks, has biases in geodemographics. In the US, people on the east/west coast tend to be more involved on Twitter than people from the inland. We have to consider this geographical bias in formulating our comparison metrics for the demographics because it might lead to inaccurate comparison results, which we shall illustrate below with a toy example(which is fully illustrated in Figure 5). Suppose we have three interest groups of users on Twitter: users interested in the TV show *Silicon Valley (SV)*[12], those interested in the TV show *Mad Men (MM)*[13], and those interested in the news media *Wall Street Journal (WSJ)*. And suppose WSJ is interested in selling more subscriptions by targeting to the people interested in those two TV shows. Geodemographics is an integral part in deciding whether to target MM or SV. For the sake of simplicity, assume there are only two bins in our geodemographic distributions: (west coast, east coast). Now suppose east coast is more densely populated than the west coast at the distribution: $(0.45, 0.55)$. Since Twitter was born in San Francisco, the general Twitter population is, say, biased towards the west: $(0.6, 0.4)$. SV is

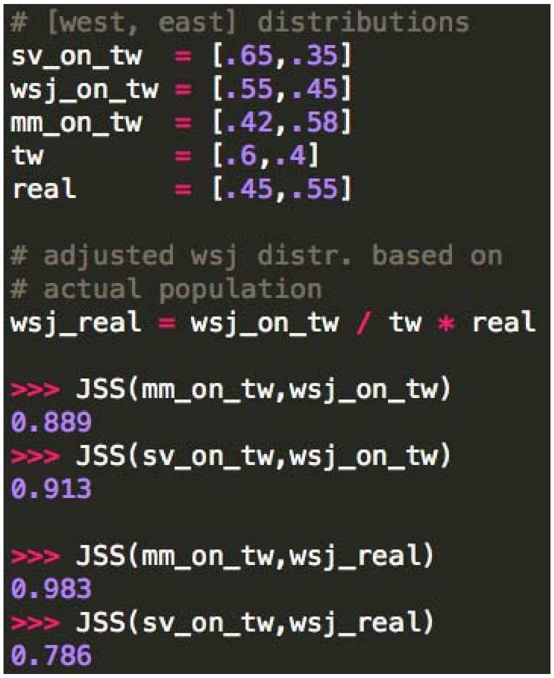
[9] http://en.wikipedia.org/wiki/Demographics_of_the_United_States

[10] http://techcrunch.com/2014/07/24/women-use-pinterest-but-they-dont-run-it/

[11] http://www.emarketer.com/Article/Whos-Using-Tumblr/1008608

[12] http://www.hbo.com/silicon-valley

[13] http://www.amctv.com/shows/mad-men

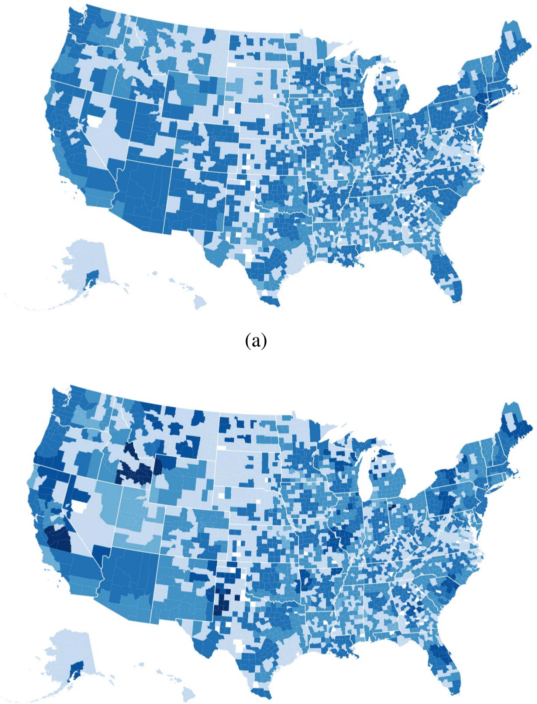


```
# [west, east] distributions
sv_on_tw  = [.65,.35]
wsj_on_tw = [.55,.45]
mm_on_tw  = [.42,.58]
tw        = [.6,.4]
real      = [.45,.55]

# adjusted wsj distr. based on
# actual population
wsj_real = wsj_on_tw / tw * real

>>> JSS(mm_on_tw,wsj_on_tw)
0.889
>>> JSS(sv_on_tw,wsj_on_tw)
0.913

>>> JSS(mm_on_tw,wsj_real)
0.983
>>> JSS(sv_on_tw,wsj_real)
0.786
```

Fig. 5. Toy example of the skewed distributions on Twitter. The calculation shows that WSJ is closer to Silicon Valley than it is to Mad Men, based on on-Twitter distributions. Using real population, WSJ is closer to Mad Men than it is to Silicon Valley, based on on-Twitter distributions. JSS$(.,.)$ in this figure symbolizes $1-$ JSD divergence. So higher means more similar.

a show about startups in California, so it also gets a west-biased distribution among its Twitter fans: $(0.65, 0.35)$. MM is a show about a fictitious New York advertising agency so it has a east-biased distribution: $(0.45, 0.55)$, but it is a relatively small bias towards the east because Twitter, as we assume, is biased towards the west. Finally WSJ, based in New York, has a distribution $(0.55, 0.45)$. WSJ is intentionally assumed to be west-biased to reflect what a biased Twitter distribution can do to an otherwise east-biased interest. The problem occurs when we try to calculate the JSS (i.e., 1- JSD) based on the on-Twitter distributions. The calculation shows that WSJ is closer to Silicon Valley than it is to Mad Men (0.889 vs. 0.913). But once we adjust the WSJ distribution based on actual population, the calculation shows that WSJ is closer to Mad Men instead (0.983 vs. 0.786). We adjust WSJ but not for the TV show because WSJ is targeting the TV shows' Twitter audience not their actual viewership.

Almost every scientific questions beckons "how" and "why". The Toy example in Figure 5 illustrates the "how".But why is it important for us to adjust Twitter's bias? The comparison between Figure 4 and Figure 6 illustrate how dramatic the bias can be. In the un-adjusted version (Figure 4), metropolitan areas like Seattle, Dallas are not well represented due to the bias. We believe this adjustment is critical in our methodology and makes it more robust.



(a)

(b)

Fig. 6. Histogram distributions same as in Figure 4, adjusted for the Twitter demographic bias. (a) Raw counts of Twitter users in each DMA region (b) Counts normalized by each region's population.

VI. Experiments

Our experiments need to evaluate two different claims we have made.

- First, a set of experiments should illustrate how accurately our proposed system estimates users' geographical distribution given an interest on Twitter.
- Second, another set of experiments should demonstrate how reasonable and effective our demographic comparison metric is.

A. The Data

To best evaluate our propositions, we use real Twitter data in our experiments. Table I summarizes the characteristics in our dataset. From August to November 2013, we download tweets related to 241 particular TV programs that are broadcasting (or have broadcasted) in the US. We identify those tweets through "#hashtags" related to each TV program. A TV program could have multiple #hashtags. For example, the AMC series *Breaking Bad* relates itself to two #hashtags: #BreakingBad and #GoodByeBreakingBad. We manually collect over 300 #hashtags.

When we apply our tagging system to this dataset of 7.3 million Twitter users, we find out that only 883 thousand (about 12%) users are tagged as US-based by our system. These US-based accounts are heavily concentrated in metropolitan areas

TABLE I
SUMMARY DESCRIPTION OF OUR DATASET; THE LAST FOUR ENTRIES ARE ESTIMATED STATISTICS.

Collection start time	2013 August 1st
Collection end time	2013 November 31st
Tweets origins	All around the world
Tweets collected	Over 100 million
Unique twitter users	7,346,392
Users w/ GPS tracking	$< 1\%$
Interest sets involved	241
Interest origins	American TV programs, CPG[14] brands, service providers
Users w/ US location string	883,715
Users in New York area	82,985
Users in Los Angeles area	71,688
Users in Dallas area	19,682

(New York area has over 9% of all US-based accounts in our sample while Los Angeles ares has over 8% of that share).

B. *Tagging recall & accuracy*

Tagging 12% of all users from the dataset is considerably better than what one can do with GPS information, which is less than 1% in our data. Our algorithm can provide a much larger geo-tagged sample than simply using GPS information, but this advantage is conditional on the accuracy of the algorithmic tagging.

It is difficult to measure the accuracy of the algorithmic tagging in a straightforward way. Previously released Twitter datasets[13] do not tag the tweets geographically. And it is impractical for us to manually tag all the twitter users or even a convincingly large sample of them.

Since our recall is already significantly better than using only GPS information, we focus on designing an evaluation for accuracy. The idea is to use a sampling process to select the users for manually resolving its location. We want to independently sample each bin from the histogram whose accuracy is to be evaluated. Within each histogram bin, we select users for verification like how active learning sample selection[14]. The upcoming selections depend on the description string of the previously selected users in that bin and whether our system tags them correctly into this bin. We stop the selection process when the calculated running accuracy achieves certain statistical convergence. Table II summarizes a few accuracy tests. Some of the most populous areas in the US are used. In each row of Table II, we list the canonical location description, the DMA region it should be assigned to, the number of users tagged by our algorithm, the number of tagged users we have judged by human, the number of misclassified locations (deemed by human judges), and the popular location strings on which the algorithm makes mistake. In general the accuracy is consistent and robust.

What we find to be interesting is in the mistakes the algorithm has made. There are two major sources of mistakes. First is ambiguity. For example, Los Angeles, CA vs. Los Angeles, Chile and Dallas, TX vs. Dallas, PA. Another major source is users' humorous intention. We find it interesting that many users specify "future New Yorker" in their location description. "Metropolitan Detention Center(MDC) Los Angeles" is also a popular location according to the users on Twitter. MDC is a Federal prison in downtown Los Angeles.

C. *Effective or not: JSS-based comparison*

Given that our tagging algorithm does a reasonably good job approximating the true underlying geographic user distribution for each Twitter interest, we should evaluate the effectiveness of the proposed JSS-based comparison metric.

Defining a metric is a subjective matter. For example, the H-index[15] in academic citation is proposed to capture both productivity and impact in a single metric. But it is impossible to say if an index assigned to a particular author is *wrong*. Instead, what we are most interested in is whether a metric is *likely* and *appropriate*. We take the original idea from locality sensitive hashing and adapt it to our domain: for two Twitter interests to have high affinity ("affinity" defined below), having similar location distributions is likely to be necessary; on the other hand, if two interests have dissimilar location distributions, it is likely to be sufficient to dismiss their affinity. Based on this logic, we perform two kinds of checks:

- We extract Twitter interest pairs with the highest affinity score, and check whether those pairs indeed have similar location distributions (i.e., JSS between the histograms).
- We extract the Twitter interest pairs that have dissimilar location distributions (i.e., low JSS between the histograms), and check whether indeed they have low affinity.

affinity between a Twitter interest pair (A, B) can be loosely defined as the likelihood for a user to be interested in A given that s/he is interested in B or vice versa. Detailed definitions, variations, and its relation to frequent item set mining can be found in this recent work[16]. Expecting positive evidences supporting the two described checks, we summarize our findings in Figure 7. The blue crosses in Figure 7 correspond to the first of the two checks mentioned above: high affinity score indeed promises high JSS. The red circles in Figure 7 correspond to the second check: low JSS indeed results in low affinity score. This result is by no means hard proof but we think it is strong evidence that the proposed JSS is a meaningful, well-defined metric to use when comparing location distribution histogram.

VII. ACKNOWLEDGMENTS

This work is supported in part by the following grants: NSF awards CCF-0833131, CNS-0830927, IIS-0905205, CCF-0938000, CCF-1029166, ACI-1144061, IIS-1343639, and CCF-1409601; DOE awards DE-FG02-08ER25848, DE-SC0001283, DE-SC0005309, DESC0005340, and DESC0007456; AFOSR award FA9550-12-1-0458.

VIII. CONCLUSION AND FUTURE WORK

In this work, we describe on the problem of offline user location estimation using online information. We are particularly interested in applications of TV segment advertising. Unlike

TABLE II
TAGGING ACCURACY AT POPULAR LOCATIONS AND THEIR FREQUENTLY MADE MISTAKES.

Location	DMA region	Tagged	Judged	Misclassified	Frequently made mistakes
New York, NY	NEW YORK	82985	100	12	*new york not nyc. upstate New York. future New Yorker*
Los Angeles, CA	LOS ANGELES	71688	100	8	*mdc los angeles. los angeles chile*
Chicago, IL	CHICAGO	27388	50	4	*somewhere close to chicago*
Boston, MA	BOSTON (MANCHESTER)	26074	50	7	*wishin i lived in boston. boston strong*
Kansas, MO	KANSAS CITY	23657	50	1	*Arkansas City KS*
Dallas, TX	DALLAS-FT. WORTH	19682	50	1	*Dallas PA*
Buffalo, NY	BUFFALO	4601	30	1	*buffalo wings*

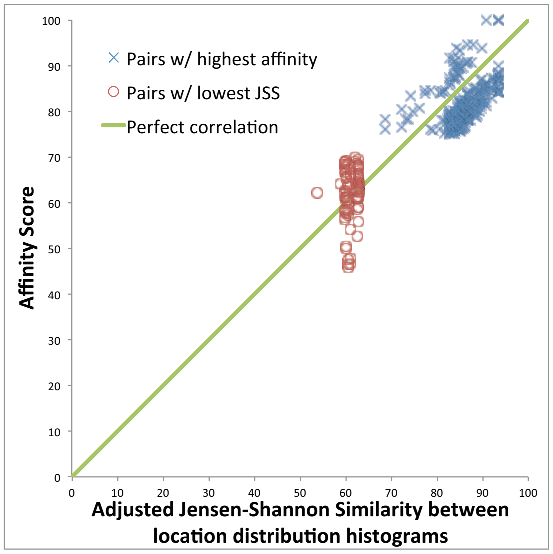


Fig. 7. By contrasting with affinity scores, we illustrate the usefulness of the proposed adjusted Jensen-Shannon Similarity (1-JSD) between location distribution histograms. Both affinity score and JSS are normalized between 0 and 100.

previous works, we propose a geo-tagging method that does not require GPS information from users. The tagger works with loosely structured information such as English location description. To digest large amount of unlabeled text-based location description, we propose a neural language model to capture the semantic similarity among the location descriptions. The language model can help reduce the otherwise expensive geolocating service lookups by internally resolving similar areas, neighborhoods, etc. onto the same description. To make the location distribution histogram estimations more robust, we illustrate a simple, effective way to adjust for the bias introduced by Twitter demographics. We also propose a metric for comparing geodemographic histograms (Jensen-Shannon Similarity). In the experiments section, we demonstrate the recall and accuracy of our language-based, GPS-free user location distribution estimation. In addition, we illustrate the effectiveness of the proposed distribution estimation metric by performing two heuristic checks. In the future, we plan to expand the application beyond the US to a global scale since Twitter and other social network services are global services.

REFERENCES

[1] J. Zhang, H. Zhao, and Y. Xie, “Follow you from your photos,” in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, Aug 2013, pp. 985–992.

[2] A. Sadilek, H. Kautz, and J. P. Bigham, “Finding your friends and following them to where you are,” in *WSDM ’12*.

[3] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant, “Detecting influenza epidemics using search engine query data,” *Nature*, vol. 457, pp. 1012–1014, 2009.

[4] A. Mnih and G. E. Hinton, “A scalable hierarchical distributed language model,” in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1081–1088.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[6] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[7] F. Aurenhammer, “Voronoi diagrams&mdash;a survey of a fundamental geometric data structure,” *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, Sep. 1991.

[8] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975. [Online]. Available: http://doi.acm.org/10.1145/361002.361007

[9] G. Van Brummelen, *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. Princeton University Press, 2012. [Online]. Available: http://books.google.com/books?id=mYz7QIt3vQoC

[10] G. Foody, “Thematic map comparison,” in *Photogrammetric Engineering and Remote Sensing*, vol. 70, no. 5, 2004, pp. 627–633.

[11] R. Levine, K. Yorita, M. Walsh, and M. Reynolds, “A method for statistically comparing spatial distribution maps,” in *Int J Health Geogr.*, 2009.

[12] N. C. Smeeton, “Early history of the kappa statistic,” pp. 795–795, 1985.

[13] Y. Xie, Z. Chen, Y. Cheng, K. Zhang, A. Agrawal, W.-K. Liao, and A. Choudhary, “Detecting and tracking disease outbreaks by mining social media data,” in *IJCAI’13*. AAAI Press.

[14] K. Zhang, Y. Xie, Y. Yang, A. Sun, H. Liu, and A. Choudhary, “Incorporating conditional random fields and active learning to improve sentiment identification,” *Neural Networks*, vol. 58, no. 0, pp. 60 – 67, 2014.

[15] J. E. Hirsch, “An index to quantify an individual’s scientific research output,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 46, pp. 16 569–16 572, 2005.

[16] Y. Xie, D. Palsetia, G. Trajcevski, A. Agrawal, and A. Choudhary, “Silverback: Scalable association mining for temporal data in columnar probabilistic databases,” in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, March 2014, pp. 1072–1083.