A Content-Driven Framework for Geolocating Microblog Users

ZHIYUAN CHENG, JAMES CAVERLEE, and KYUMIN LEE, Texas A&M University

Highly dynamic real-time microblog systems have already published petabytes of real-time human sensor data in the form of status updates. However, the lack of user adoption of geo-based features per user or per post signals that the promise of microblog services as location-based sensing systems may have only limited reach and impact. Thus, in this article, we propose and evaluate a probabilistic framework for estimating a microblog user's location based purely on the content of the user's posts. Our framework can overcome the sparsity of geo-enabled features in these services and bring augmented scope and breadth to emerging location-based personalized information services. Three of the key features of the proposed approach are: (i) its reliance purely on publicly available content; (ii) a classification component for automatically identifying words in posts with a strong local geo-scope; and (iii) a lattice-based neighborhood smoothing model for refining a user's location estimate. On average we find that the location estimates converge quickly, placing 51% of users within 100 miles of their actual location.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database applications–Data mining; J.4 [**Computer Application**]: Social and Behavioral Sciences

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Microblog, Twitter, location-based estimation, spatial data mining, text mining

ACM Reference Format:
Cheng, Z., Caverlee, J., and Lee, K. 2013. A content-driven framework for geolocating microblog users. ACM Trans. Intell. Syst. Technol. 4, 1, Article 2 (January 2013), 27 pages.
DOI = 10.1145/2414425.2414427 http://doi.acm.org/10.1145/2414425.2414427

1. INTRODUCTION

Highly dynamic real-time microblog systems like Twitter, Plurk, and Sina Weibo have already published petabytes of real-time human sensor data in the form of status updates. Coupled with growing location-sharing social media services like Gowalla, Foursquare, and Google Latitude, we can see unprecedented access to the activities, actions, and trails of millions of people, with the promise of deeper and more insightful understanding of the emergent collective knowledge "wisdom of the crowds" embedded in these activities and actions.

As one of the most popular microblog services, Twitter has experienced an exponential explosion in its user base since its creation in 2006, reaching approximately 105 million users as of 2010. During its first four years of existence, Twitter has changed the world and the way we live, surprising the public and the research community [Johnson 2009; Miller 2010]. At its current rate, Twitter publishes approximately 50 million

This article is an expanded version of Cheng et al. [2010], which appeared in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 759–768.

Authors' addresses: Z. Cheng (corresponding author), J. Caverlee, and K. Lee, Department of Computer Science and Engineering, Texas A&M University, 403 H.R. Bright Building, College Station, TX 77843-3112; email: zcheng@cse.tamu.edu.

DOI 10.1145/2414425.2414427 http://doi.acm.org/10.1145/2414425.2414427

Table I. Categorization of Twitter User's Location Field

Category	Percentage	Example(s)
Coordinates	5%	"29.3712, −95.2104"
City-Level Locations	21%	"Los Angeles, CA", "New York City"
General / Nonsensical / Missing	74%	"California", "Wonderland", NULL

tweets per day (or about 600 tweets per second), totaling around 20 billion tweets in all [Cheema 2010]. Twitter has become one of the leading social networking sites with global impact [Huberman et al. 2008] and has been recognized in its early days as a potentially useful means to study the formation of communities [Java et al. 2007].

Microblog systems like Twitter contain a huge volume of content, diversified topics, and a wide user base, which in total provide significant opportunities for mining and exploring the real-time Web. Mining this people-centric sensor data promises new personalized information services, including local news summarized from tweets of nearby Twitter users [Yardi and Boyd 2010], the targeting of regional advertisements, spreading business information to local customers [Patrick and Kevin 2009], and novel location-based applications (e.g., Twitter-based earthquake detection, which can be faster than through traditional official channels [Sakaki et al. 2010]).

Unfortunately, microblog users have been slow to adopt geospatial features. Taking Twitter as an example, as listed in Table I, in a random sample of over 1 million Twitter users, we find that only 21% have listed a user's location as granular as a city name (e.g., Los Angeles, CA); only 5% have listed a location as granular as latitude/longitude coordinates (e.g., 29.3712, −95.2104); the rest are overly general (e.g., California), missing altogether, or nonsensical (e.g., Wonderland). In addition, Twitter began supporting per-tweet geo-tagging in August 2009. Unlike user location (which is a single location associated with a user and listed in each Twitter user's profile), this per-tweet geo-tagging promises extremely fine-tuned Twitter user tracking by associating each tweet with a latitude and longitude. Our sample shows, however, that fewer than 0.42% of all tweets actually use this functionality. Together, the lack of user adoption of geo-based features per user or per post signals that the promise of microblog services as location-based sensing systems may have only limited reach and impact.

To overcome this location sparsity problem, we propose that a reasonable framework to predict a microblog user's location should contain the following features: (i) the proposed framework should be generalizable across social media sites and future human-powered sensing systems; (ii) the framework should be robust in the presence of noise and the sparsity of spatial cues in a microblog user's posts; (iii) the framework should provide accurate and reliable location estimation; and (iv) the prediction framework should be based purely on the publicly available data from the user, with no need for proprietary data from system operators (e.g., backend database) or privacy-sensitive data from users (e.g., IP or user/pass).

With these guidelines in mind, in this manuscript, we propose a framework which is based purely on the content of the user's posts, even in the absence of any other geospatial cues. Our intuition is that a user's posts may encode some location-specific content, either specific place names or certain words or phrases more likely to be associated with certain locations than others (e.g., "howdy" for people from Texas). In this way, we can fill-the-gap for the large portion of microblog users lacking city-level granular location information. By augmenting the massive human-powered sensing capabilities of Twitter and related microblogging services with content-derived location information, this framework can overcome the sparsity of geo-enabled features in these services and bring augmented scope and breadth to emerging location-based personalized information services. This in turn could lead to even broader applications

Table II. Examples of Tweets

User	Tweet	Topic	Location Hint
User1	More like this, please. White House science fair: http://bit.ly/9bKI7h	Education	DC
	C++ celebrates 25th anniv of its first commercial release! #TAMU	C++	College Station
	@jelsas I read that as #applausability. I am clapping for your tweet.	Conversation	N/A
	Off to Chicago. Found a Papasito's in concourse E at IAH!	Travel	Chicago / Houston
User2	Shaq dmc. In the place to be. I been doin this here since 93.	Conversation	N/A
	I'm n da apple store. I almost got away a wit dat a new iphone.	Personal	N/A
	Vote for my boy rick fox on dancing wit da stars.	Conversation	N/A
User3	@Peter Dude, were you in San Francisco recently?	Conversation	San Francisco
	Got an email from a guy in Serbia asking for source code.	Personal	Serbia
	Really impressed by fans of the Aggies.	Conversation	TAMU / UC Davis

of social media in time-critical situations such as emergency management and tracking the diffusion of infectious diseases.

Effectively geolocating a microblog user based purely on the content of her posts is a difficult task, however.

—First, microblog status updates are inherently noisy, mixing a variety of daily interests (e.g., food, sports, daily chatting with friends). For example, as shown in Table II, User1 talks about education, C++, conversational topics, and travel. Are there clear location signals embedded in this mix of topics and interests that can be identified for locating a user?

—Second, microblog users often rely on shorthand and nonstandard vocabulary for informal communication, meaning that traditional gazetteer terms and proper place names (e.g., Eiffel Tower) may not be present in the content of the posts at all, making the task of determining which terms are location sensitive nontrivial. As we can see from User2's posts in Table II, User2 relies on informal language which may cause difficulty in analyzing the user's content.

—Third, even if we could isolate the location-sensitive attributes of a user's posts, a user may have interests that span multiple locations beyond her immediate home location, meaning that the content of her posts may be skewed toward words and phrase more consistent with outside locations. For example, New Yorkers may talk about NBA games in Los Angeles or the earthquake in Haiti. This can also be observed from User1 and User3 in Table II.

—Fourth, a user may have more than one associated location, for example, due to travel, meaning that content-driven location estimation may have difficulty in precisely identifying a user's location.

With these issues in mind, in this manuscript we propose and evaluate a probabilistic framework for estimating a microblog user's city-level location which satisfies all the requirements we mentioned. Taking only a user's publicly available content as the input data, the framework is generalizable across different microblogging sites, and other online social media Web sites. Experimentally, we select Twitter as an exemplar microblogging service over which to evaluate our framework. The proposed approach relies on three key features: (i) its data input of pure content, without any external data from users or Web-based knowledge bases; (ii) a classifier which identifies words in status updates with a local geographic scope; and (iii) a lattice-based neighborhood smoothing model for refining the estimated results. The system provides k estimated cities for each user with a descending order of possibility. On average, 51% of randomly sampled microblog users are placed within 100 miles of their actual location (based on an analysis of just 100s of posts). We find that increasing amounts of data (in the form of wider coverage of microblog users and their associated tweets) results in more precise location estimation, giving us confidence in the robustness and continued refinement of the approach.

The rest of this manuscript is organized as follows: Related work is in Section 2. Section 3 formalizes the problem of predicting a microblog user's geolocation and briefly describes the sampled dataset used in the experiments. In Section 4, our estimation algorithm and corresponding refinements are introduced. We present the experimental results in Section 5. Finally, conclusions and future work are discussed in Section 6.

2. RELATED WORK

Studying the geographical scope of online content has attracted attention by researchers in the last decade, including studies of blogs [Fink et al. 2009; Lin and Halavais 2004], Web pages [Amitay et al. 2004], search engine query logs [Backstrom et al. 2008; Yi et al. 2009], and even Web users [Hurst et al. 2007]. Prior work relevant to this manuscript can be categorized roughly into three groups based on the techniques used in geolocating: content analysis with terms in a gazetteer, content analysis with probabilistic language models, and inference via social relations.

Several studies try to estimate the location of Web content utilizing content analysis based on geo-related terms in a specialized external knowledge base (a gazetteer). Amitay et al. [2004], Fink et al. [2009], and Zong et al. [2005] extracted addresses, postal code, and other information listed in a geographical gazetteer from Web content to identify the associated geographical scope of Web pages and blogs.

Serdyukov et al. [2009] generate probabilistic language models based on the tags that photos are labeled with by Flickr users. Based on these models and Bayesian inference, they show how to estimate the location for a photo. In terms of the intention, their method is similar to our work. However, they use a GeoNames database to decide whether a user-submitted tag is a geo-related tag, which can overlook the spatial usefulness of words that may have a strong geo-scope (e.g., earthquake, casino, and so on). Separately, the work of Crandall et al. [2009] proposes an approach combining textual and visual features to place images on a map. They have restrictions in their task that their system focuses on which of ten landmarks in a given city is the scope of an image.

In the area of privacy inference, a few researchers have been studying how a user's private information may be inferred through an analysis of the user's social relations. Backstrom et al. [2010], Lindamood et al. [2009], and Hearthely et al. [2009] all share a similar assumption that users related in social networks usually share common attributes. These methods are orthogonal to our effort and could be used to augment the content-based approach taken in this manuscript by identifying common locations among a Twitter user's social network. Besides, several literatures address the interplay between distance and social tie strength. Given location trails from two users, both Cranshaw et al. [2010] and Zheng et al. [2011] propose metrics to measure similarity between two users given their location history, and predict friendship according to the similarities. McGee et al. [2011] investigates the relationship between the strength of the social tie between a pair of friends and the distance between the pair with a set of 6 million geo-coded Twitter users and their social relations. They observe a bimodal distribution in Twitter, with one peak around 10 miles from people who live nearby, and another peak around 2500 miles, which further validates Twitter's use as both a social network (with geographically nearby friends) and as a social media platform (with very distant connections). In addition, they also observe that users with stronger tie strength (reciprocal friendship) are more likely to live near each other than users with weak ties.

Recent work on detecting earthquakes with real-time Twitter data makes use of location information for tracking the flow of information across time and space [Sakaki et al.

2010]. Sakaki et al. consider each Twitter user as a sensor and apply Kalman filtering and particle filtering to estimate the center of the bursty earthquake. Their algorithm requires prior knowledge of where and when the earthquake is reported, emphasizing tracking instead of geolocating users. As a result, this and related methods could benefit from our efforts to assign locations to users for whom we have no location information.

As people care about the privacy issues of real-time microblog systems and location-sharing services, we do note that researchers are working in the opposite direction of trying to protect a user's location information and other sensitive information [Beresford and Stajano 2003; Kalnis et al. 2007; Freni et al. 2010]. Our work could be helpful for researchers in the domain of location-preserving data mining, and raise awareness of the privacy leakages and risks associated with posting location-relevant content to microblogging services.

3. PRELIMINARIES

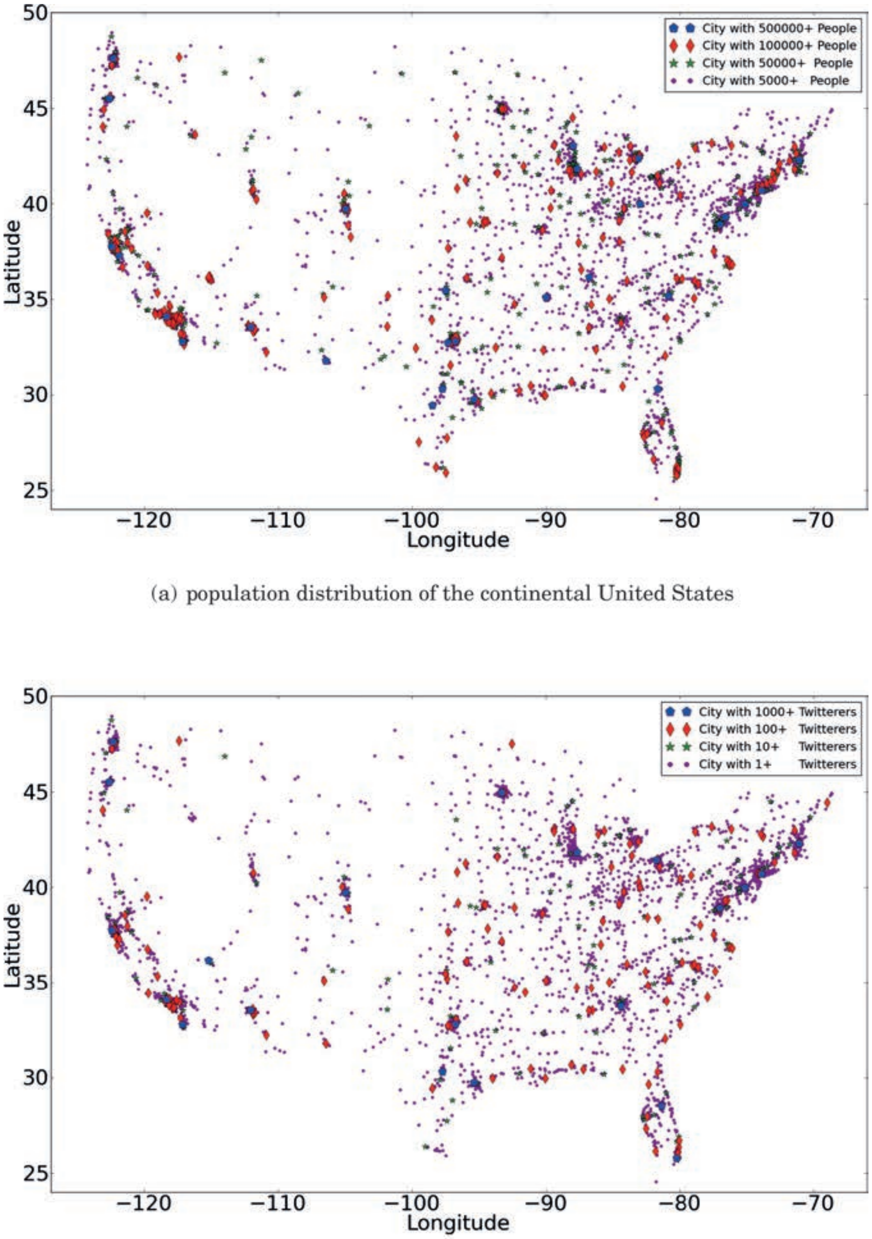
In this section, we briefly explain our dataset, formalize the research problem, and describe the experimental setup.

3.1. Location Sparsity on Twitter

To derive a representative sample of Twitter users, we employed two complementary crawling strategies: crawling through Twitter's public timeline API and crawling by breadth-first search through social edges to crawl each user's friends (following) and followers. The first strategy can be considered as random sampling from active Twitter users (whose tweets are selected for the public timeline), while the second strategy extracts a directed acyclic subgraph of the whole Twitter social graph, including less active Twitter users. We combine the two strategies to avoid bias in either one. Using the open-source library twitter4j [Yamamoto 2007] to access Twitter's open API [Twitter 2007] from September 2009 to January 2010, we collected a base dataset of 1,074,375 user profiles and 29,479,600 status updates.

Each user profile includes the capacity to list the user's name, location, a Web link, and a brief biography. We find that 72.05% of the profiles collected do list a nonempty location, including locations like "Hollywood, CA", "England", and "UT: 40.708046,-73.789259". However, we find that most of these user-submitted locations are overly general with a wide geographic scope (e.g., California, worldwide), missing altogether, or nonsensical (e.g., Wonderland, "CALI to FORNIA"). Specifically, we examine all locations listed in the 1,074,375 user profiles and find that just 223,418 (21% of the total) list a location as granular as a city name and that only 61,335 (5%) list a location as granular as a latitude/longitude coordinate. This absence of granular location information for the majority of Twitter users (74%) indicates the great potential in estimating or recommending location for a Twitter user.

For the rest of the article, we focus our study of Twitter user location estimation on users within the continental United States. Toward this purpose, we filter all listed locations that have a valid city-level label in the form of "cityName", "cityName, stateName", and "cityName, stateAbbreviation", where we consider all valid cities listed in the Census 2000 U.S. Gazetteer [USCensusBureau 2002] from the U.S. Census Bureau. Even when considering these data forms, there can still be ambiguity for cities listed using just "cityName", for example, there are three cities named Anderson, four cities named Arlington, and six cities called Madison. For these ambiguous cases, we only consider cities listed in the form "cityName, stateName", and "cityName, stateAbbreviation". After applying this filter, we find that there are 130,689 users (with 4,124,960 status updates), accounting for 12% of all sampled Twitter users. This sample of Twitter users is representative of the actual population of the United States as can be seen in Figure 1(a) and Figure 1(b).



(a) population distribution of the continental United States

(b) user distribution of sampled Twitter dataset

Fig. 1. Comparison between the actual U.S. population and the sample Twitter user population.

3.2. Problem Statement

Given the lack of granular location information for Twitter users, our goal is to estimate the location of a user based purely on the content of her tweets. Having a reasonable estimate of a user's location can enable content personalization (e.g., targeting advertisements based on the user's geographical scope, pushing related news stories, etc.), targeted public health Web mining (e.g., a Google Flu Trends-like system that analyzes tweets for regional health monitoring), and local emergency detection (e.g., detecting emergencies by monitoring tweets about earthquakes, fires, etc.). By focusing on the content of a user's Twitter stream, such an approach can avoid the need for private user information, IP address, or other sensitive data. With these goals in mind, we focus on city-level location estimation for a Twitter user, where the problem can be formalized as follows.

Location Estimation Problem. Given a set of tweets $S_{tweets}(u)$ posted by a Twitter user u, estimate a user's likelihood score of being located in city i: $s_{likelihood}(i|S_{tweets}(u))$, such that the city with maximum likelihood score $l_{est}(u)$ is the user's actual location $l_{act}(u)$.

As we have noted, location estimation based on tweet content is a difficult and challenging problem. Twitter status updates are inherently noisy, often relying on shorthand and nonstandard vocabulary. It is not obvious that there are clear location cues embedded in a user's tweets at all. A user may have interests which span multiple locations and a user may have more than one natural location.

3.3. Evaluation Setup and Metrics

Toward developing a content-based user location estimator, we next describe our evaluation setup and introduce four metrics to help us evaluate the quality of a proposed estimator.

Test data. In order to be fair in our evaluation of the quality of location estimation, we build a test set[1] that is separate from the 130,689 users previously identified (and that will be used for building our models for predicting user location). In particular, we extract a set of active users with 1000+ tweets who have listed their location in the form of latitude/longitude coordinates. Since these types of user-submitted locations are typically generated by smartphones, we assume these locations are correct and can be used as ground truth. We filter out spammers, promoters, and other automated-script-style Twitter accounts by supervised learning using features derived from Lee et al.'s work [Lee et al. 2010] on Twitter bot detection. For example, features such as the average posts per day, the ratio of number of following and number of followers, and the ratio of the number URLs in the 20 most recent posts are used in characterizing a user. After bot filtering, the test set will consist of primarily "regular" Twitter users for whom location estimation would be most valuable. Finally, we arrive at 5,119 test users and more than 5 million of their tweets. These test users are distributed across the continental United States similar to the distributions seen in Figure 1(a) and Figure 1(b).

Metrics. To evaluate the quality of a location estimator, we compare the estimated location of a user versus the actual city location (which we know based on the city corresponding to her latitude/longitude coordinates). The first metric we consider is the *error distance* which quantifies the distance in miles between the actual location of the user $l_{act}(u)$ and the estimated location $l_{est}(u)$. The *Error Distance* for user u is defined as

$$ErrDist(u) = d(l_{act}(u), l_{est}(u)).$$

To evaluate the overall performance of a content-based user location estimator, we further define the *average error distance* across all test users U.

$$AvgErrDist(U) = \frac{\sum_{u \in U} ErrDist(u)}{|U|}$$

[1] Data associated with this manuscript is available at http://infolab.tamu.edu/data/.

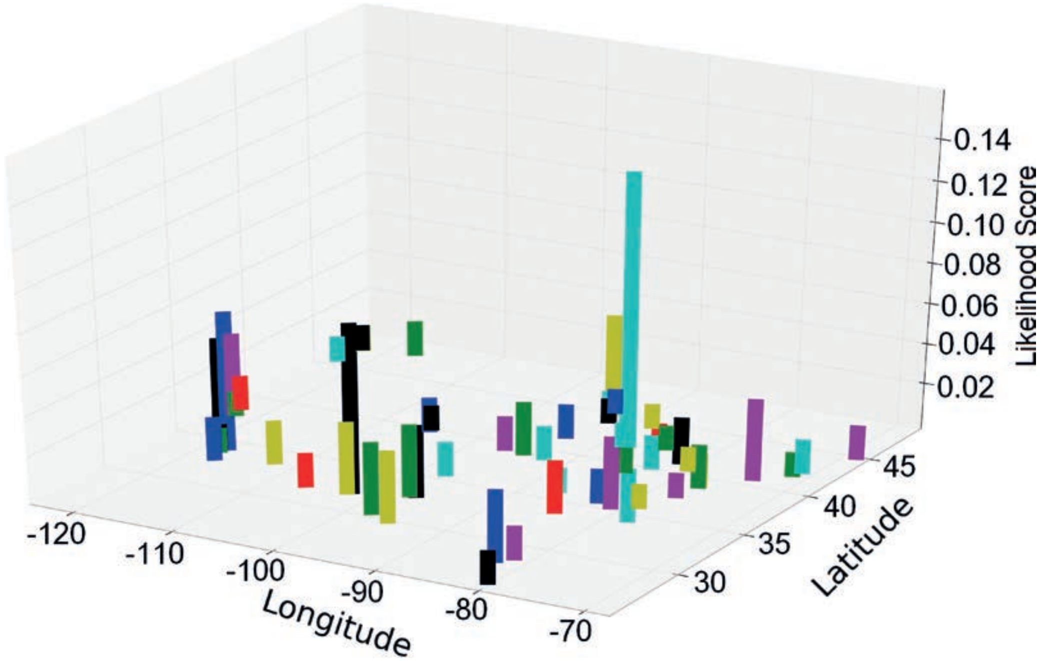


Fig. 2. City estimates for the term “rockets”.

A low *average error distance* means that the system can geolocate users close to their real location on average, but it does not give strong insight into the distribution of location estimation errors. Hence, the next metric—*accuracy*—considers the percentage of users with their error distance categorized in the range of 0–100 miles.

$$Accuracy(U) = \frac{|\{u|u \in U \wedge ErrDist(u) \leq 100\}|}{|U|}$$

Further, since the location estimator predicts k cities for each user in decreasing order of confidence, we define the *accuracy with k estimations (accuracy@k)* which applies the same accuracy metric, but over the city in the top-k with the least error distance to the actual location. In this way, the metric shows the capacity of an estimator to identify a good candidate city, even if the first prediction is in error.

4. CONTENT-BASED LOCATION ESTIMATION: OVERVIEW AND APPROACH

In this section, we begin with an overview of our baseline approach for content-based location estimation and then present two key optimizations for improving and refining the quality of location estimates.

Baseline Location Estimation. First, we can directly observe the actual distribution across cities for each word in the sampled dataset. Based on maximum likelihood estimation, the probabilistic distribution over cities for word w can be formalized as $p(i|w)$ which identifies for each word w the likelihood that it was issued by a user located in city i. For example, for the word “rockets”, we can see its city distribution in Figure 2 based on the tweets in the sampled dataset (with a large peak near Houston, home of NASA and the NBA basketball team Rockets).

Of course users from cities other than Houston may tweet the word “rockets”, so reliance on a single word or a single tweet will necessarily reveal very little information about the true location of a user. By aggregating across all words in tweets posted by a particular user, however, our intuition is that the location of the user will become clear. Given the bag of words $S_{words}(u)$ extracted from a user's tweets $S_{tweets}(u)$, we propose to estimate the likelihood score of the user being located in city i as

$$s_{likelihood}(i|S_{words}(u)) = \sum_{w \in S_{words}(u)} p(i|w).$$

$p(i|w)$ denotes the probability for word w to occur in city i. And it can be calculated by applying Bayes' rule $p(i|w) = \frac{p(w|i)*p(i)}{p(w)}$. In practice, we can directly get the probability of $p(i|w)$ by applying maximum likelihood estimation on the training data without using Bayes' rule. Given the probability to locate an individual word w at city i, we simply aggregate probabilities given the words that occurred in user u's bag of words $S_{words}(u)$ extracted from her tweets, and come up with the likelihood to locate user u in city i. In addition, the city with the highest likelihood across all cities is selected as the estimated location $l_{est}(u)$ for user u, formally defined as

$$l_{est}(u) = i_{mle} = \arg\max_{i \in S_c} s_{likelihood}(i|S_{words}(u)),$$

where S_c refers to all the cities considered in our dataset. This location estimator is further formalized in Algorithm 1.

ALGORITHM 1: Content-Based User Location Estimation

```
Input:
$tweets$: List of n tweets from a Twitter user $u$
$cityList$: Cities in continental US with 5k+ people
$distributions$: Probabilistic distributions for words
$k$: Number of estimations for each user
Output:
$estimatedCities$: Top K estimations
$words = preProcess(tweets)$
for $city$ in $cityList$ do
    $likelihood\_score[city] \leftarrow 0$
    for $word$ in $words$ do
        $likelihood\_score[city] += distributions[word][city] * word.count$
    end for
end for
$estimatedCities = sort(likelihood\_score, cityList, k)$
return $estimatedCities$
```

Initial Results. Using this baseline approach, we estimated the location of all users in our test set using per-city word distributions estimated from the 130,689 users shown in Figure 1(b). For each user, we parsed her location and status updates (4,124,960 in all). In parsing the tweets, we eliminate all occurrences of a standard list of 319 stop-words, as well as screen names (which start with @), hyperlinks, and punctuation in the tweets. Instead of using stemming, we use the Jaccard coefficient to check whether a newly encountered word is a variation of a previously encountered word. The Jaccard coefficient is particularly helpful in handling informal content like in tweets, for example, by treating "awesoome" and "awesooome" as the word "awesome". In generating the word distributions, we only consider words that occur at least 50 times in order to build comparatively accurate models. Thus, 25,987 per-city word distributions are generated from a base set of 481,209 distinct words.

Disappointingly, only 10.12% of the 5,119 users in the test set are geo-located within 100 miles to their real locations and the AvgErrDist is 1,773 miles, meaning that such a baseline content-based location estimator provides little value. On inspection, we discovered two key problems: (i) most words are distributed consistently with the population across different cities, meaning that most words provide very little power at distinguishing the location of a user; and (ii) most cities, especially with a small population, have a sparse set of words in their tweets, meaning that the per-city word distributions for these cities are underspecified leading to large estimation errors.

In the rest of this section, we address these two problems in turn in hopes of developing a more valuable and refined location estimator. Concretely, we pursue two directions.

—*Identifying Local Words in Tweets.* Is there a subset of words which have a more compact geographical scope compared to other words in the dataset? And can these “local” words be discovered from the content of tweets? By removing noise words and nonlocal words, we may be able to isolate words that can distinguish users located in one city versus another.

—*Overcoming Tweet Sparsity.* In what way can we overcome the location sparsity of words in tweets? By exploring approaches for smoothing the distributions of words, can we improve the quality of user location estimation by assigning nonzero probability for words to be issued from cities in which we have no word observations?

4.1. Identifying Local Words in Tweets

Our first challenge is to filter the set of words considered by the location estimation algorithm (Algorithm 1) to consider primarily words that are essentially “local”. By considering all words in the location estimator, we saw how the performance suffers due to the inclusion of noise words that do not convey a strong sense of location (e.g., “august”, “peace”, “world”). By observation and intuition, some words or phrases have a more compact geographical scope. For example, “howdy” which is a typical greeting word in Texas may give the estimator a hint that the user is in or near Texas.

Toward the goal of improving user location estimation, we characterize the task of identifying local words as a decision problem. Given a word, we must decide if it is local or nonlocal. Since tweets are essentially informal communication, we find that relying on formally defined location names in a gazetteer is neither scalable nor provides sufficient coverage. That is, Twitter’s 140-character length restriction means that users may not write the full address or location name (e.g., “t-center” instead of “Houston Toyota Center”, home of the NBA Rockets team. Concretely, we propose to determine local words using a model-driven approach based on the observed geographical distribution of the words in tweets.

4.1.1. Determining Spatial Focus and Dispersion. Intuitively, a local word is one with a high local focus and a fast dispersion, that is it is very frequent at some central point (like say in Houston) and then drops off in use rapidly as we move away from the central point. Nonlocal words, on the other hand, may have many multiple central points with no clear dispersion (e.g., words like basketball). How do we assess the spatial focus and dispersion of words in tweets?

Recently Backstrom et al. introduced a model of spatial variation for analyzing the geographic distribution of terms in search engine query logs [Backstrom et al. 2008]. The authors propose a generative probabilistic model in which each query term has a geographic focus on a map (based on an analysis of the IP-address-derived locations of users issuing the query term). Around this center, the frequency shrinks as the distance from the center increases. Two parameters are assigned for each model, a constant C which identifies the frequency in the center, and an exponent α which controls the speed of how fast the frequency falls as the point goes further away from the center. The formula for the model is $Cd^{-\alpha}$ which means that the probability of the query issued from a place with a distance d from the center is approximately $Cd^{-\alpha}$. In the model, a larger α identifies a more compact geo-scope of a word, while a smaller α displays a more global popular distribution.

In the context of tweets, we can similarly determine the focus (C) and dispersion (α) for each tweet word by deriving the optimal parameters that fit the observed data. These parameters C and α are strong criteria for assessing a word's focus and dispersion, and hence, determining whether a word is local or not. For a word w, given a center, the central frequency C, and the exponent α, we compute the maximum-likelihood value like so: for each city, suppose all users tweet the word w from the city a total of n times, then we multiply the overall probability by $(Cd_i^{-\alpha})^n$; if no users in the city tweet the word w, we multiply the overall probability by $1 - Cd_i^{-\alpha}$. In the formula, d_i is the distance between city i and the center of word w. We add logarithms of probabilities instead of multiplying probabilities in order to avoid underflow. For example, let S be the set of occurrences for word w (indexed by cities which issued the word w), and let d_i be the distance between a city i and the model's center. Then

$$f(C, \alpha) = \sum_{i \in S} \log Cd_i^{-\alpha} + \sum_{i \notin S} \log(1 - Cd_i^{-\alpha})$$

is the likelihood value for the given center, C and α. Backstrom et al. [2008] also prove that $f(C, \alpha)$ has exactly one local maximum over its parameter space which means that when a center is chosen, we can iterate C and α to find the largest $f(C, \alpha)$ value (and hence, the optimized C and α). Instead of using a brute-force algorithm to find the optimized set of parameters, we divide the map of the continental United States into lattices with a size of two by two square degrees. For the center in each lattice, we use golden section search [Press et al. 1986] to find the optimized central frequency and the shrinking factor α. Then we zoom into the lattice which has the largest likelihood value, and use a finer-grained mesh on the area around the best chosen center. We repeat this zoom-and-optimize procedure to identify the optimal C, and α. Note that the implementation with golden section search can generate an optimal model for a word within a minute on a single modern machine and is scalable to handle Web-scale data. To illustrate, Figure 3 shows the optimized model for the word "rockets" centered around Houston.

4.1.2. Training and Evaluating the Model. Given the model parameters C (focus) and α (dispersion) for every word, we could directly label as local words all tweet words with a sufficiently high focus and fast dispersion by considering some arbitrary thresholds. However, we find that such a direct application may lead to many errors (and ultimately poor user location estimation). For example, some models may lack sufficient supporting data resulting in a clearly incorrect geographic scope. Hence, we augment our model of local words with coordinates of the geo-center, since the geographical centers of local words should be located in the continental United States, and the count of the word occurrences, since a higher number of occurrences of a word will give us more confidence in the accuracy of the generated model of the word. In addition, we also take the word's semantic meaning into consideration.

Using these features, we train a local word classifier using the Weka toolkit [Witten and Frank 2005]—which implements several standard classification algorithms like Naive bayes, SVM, AdaBoost, etc.—over a hand-labeled set of standard English words taken from the 3esl dictionary [Atkinson 2007]. Of the 19,178 words in the core dictionary, 11,004 occur in the sampled Twitter dataset. Using 10-fold cross-validation and the SimpleCart classifier, we find that the classifier has a *Precision* of 98.8% and *Recall*

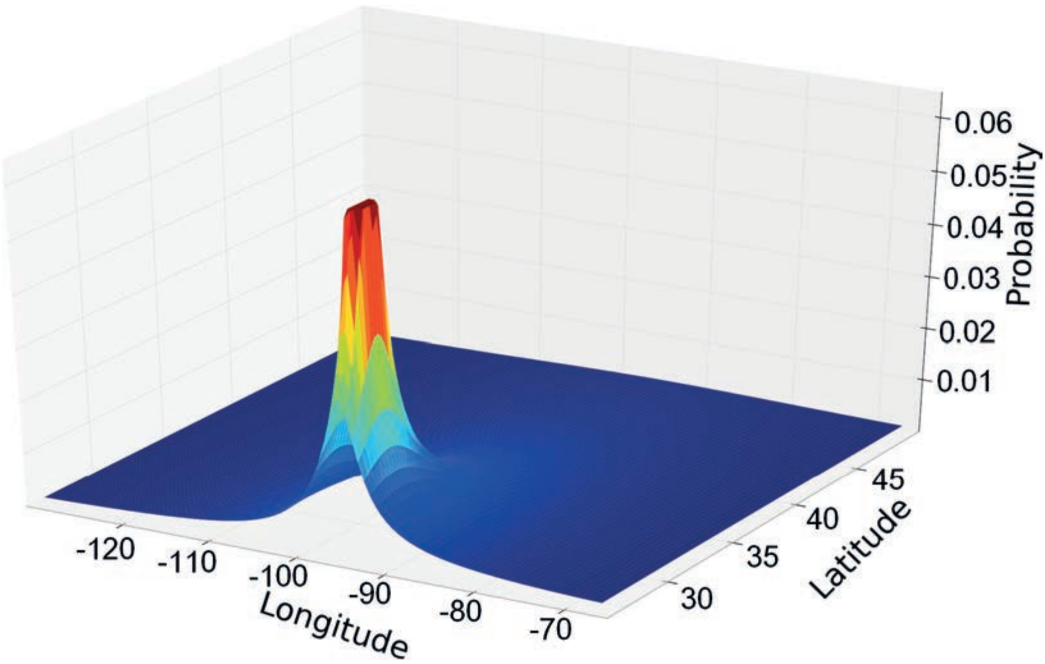


Fig. 3. Optimized model for the word “rockets”.

Table III. Example Local Words

Word	Latitude	Longitude	C_0	α
automobile	40.2	-85.4	0.5018	1.8874
casino	36.2	-115.24	0.9999	1.5603
tortilla	27.9	-102.2	0.0115	1.0350
canyon	36.52	-111.32	0.2053	1.3696
redsox	42.28	-69.72	0.1387	1.4516

and F-*Measure* both as 98.8%, indicating good quality of the classifiers trained by manually labeled data to predict a word as either locally favored or nonlocally favored. After learning the classification model over these known English words, we apply the classifier to the rest of the 14,983 tweet words (many of which are nonstandard words and not in any dictionary), resulting in 3,183 words being classified as local words. Worth mentioning, using the dictionary here is just an intuitive way that we select words for the training set. Alternatively, we can randomly select a subset of words in the tweets, and manually label them as our training set for the classifiers.

To illustrate the geographical scope of the local words discovered by the classifier, five local word models are listed in Table III. The word “automobile” is located around two hundred miles south of Detroit which is the traditional auto manufacturing center of the U.S. The word “casino” is located in the center of Las Vegas, two miles east of the North Las Vegas Airport. Also “tortilla” is centered a hundred miles south of the border between Texas and Mexico. The word “canyon” is located almost at the center of the Grand Canyon. The center for the word “redsox” is located 50 miles east of Boston, home of the baseball team.

In order to visualize the geographical centers of the local favored words, a few examples are shown on the map of the continental United States in Figure 4. Based on

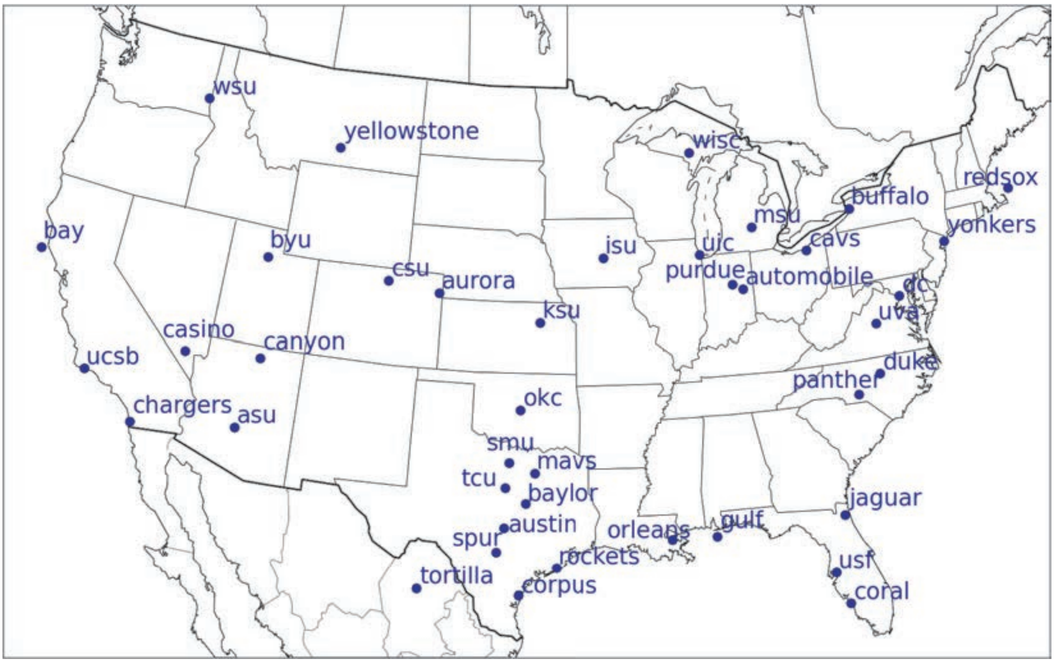


Fig. 4. Geographical centers of local words discovered in sampled Twitter dataset.

these and the other discovered local words, we will evaluate if and how user location estimation improves in the experimental study in Section 5.

4.2. Overcoming Tweet Sparsity

The second challenge for improving our content-based user location estimator is to overcome the sparsity of words across locations in our sampled Twitter dataset. Due to this sparseness, there are a large number of "tiny" word distributions (i.e., words issued from only a few cities) The problem is even more severe when considering cities with a small population. As an example, consider the distribution for the word "rockets" over the map of the continental United States displayed in Figure 2. We notice that for a specific word, the probability for the word to be issued in a city can be zero since there are no tweets including the word in our sampled dataset. In order to handle this sparsity, we consider three approaches for smoothing the probability distributions: Laplace smoothing, data-driven geographic smoothing, and model-driven smoothing.

4.2.1. Laplace Smoothing. A simple method of smoothing the per-city word distributions is Laplace smoothing (add-one smoothing) which is defined as

$$p(i|w) = \frac{1 + count(w,i)}{V + N(w)},$$

where $count(w, i)$ denotes the term count of word w in city i; V stands for the size of the vocabulary; and N(w) stands for the total count of w in all the cities. Briefly speaking, Laplace smoothing assumes every seen or unseen city issued word w once more than it did in the dataset.

Although simple to implement, Laplace smoothing does not take the geographic distribution of a word into consideration. That is, a city near Houston with zero occurrences of the word "rockets" is treated the same as a city far from Houston with zero occurrences. Intuitively, the peak for “rockets” in Houston (recall Figure 2) should impact the probability mass at nearby cities.

4.2.2. Data-Driven Geographic Smoothing. To take this geographic nearness into consideration, we consider two techniques for smoothing the per-city word distributions by considering neighbors of a city at different granularities. In the first case, we smooth the distribution by considering the overall prevalence of a word within a state; in the second, we consider a lattice-based neighborhood approach for smoothing at a more refined city-level scale.

State-level smoothing. For state-level smoothing, we aggregate the probabilities of a word w in the cities in a specific state s (e.g., Texas), and consider the average of the summation as the probability of the word w occurring in the state. Letting $S_c(s)$ denote the set of cities in the state s, the state probability can be formulated as

$$p_s(s|w) = \frac{\sum_{i \in S_c(s)} p(i|w)}{|S_c(s)|}.$$

Furthermore, the probability of the word w to be located in city i can be a combination of the city probability and the state probability

$$p'(i|w) = \lambda * p(i|w) + (1 - \lambda) * p_s(s|w),$$

where i stands for a city in the state s, and $1 - \lambda$ is the amount of smoothing. Thus, a small value of λ indicates a large amount of state-level smoothing.

Lattice-based neighborhood smoothing. Naturally, state-level smoothing is a fairly coarse technique for smoothing word probabilities. For some words, the region of a state exaggerates the real geographical scope of a word; meanwhile, the impact of a word issued from a city may have higher influence over its neighborhood in another state than the influence over a distant place in the same state. With this assumption, we apply lattice-based neighborhood smoothing.

First we divide the map of the continental United States into lattices of x by x square degrees. Letting w denote a specific word, lat a lattice, and $S_c(lat)$ be the set of cities in lat, the per-lattice probability of a word w can be formalized as

$$p(lat|w) = \sum_{i \in S_c(lat)} p(i|w).$$

In addition, we consider lattices around (the nearest lattices in all eight directions) lat as the neighbors of the lattice lat. Introducing μ as the parameter of neighborhood smoothing, the lattice probability is updated as

$$p'(lat|w) = \mu * p(lat|w) + (1.0 - \mu) * \sum_{lat_i \in S_{neighbors}(lat)} p(lat_i|w),$$

where $S_{neighbors}(lat)$ includes all the neighbors of lattice lat (the nearest lattices in all eight directions).

In order to utilize the smoothed lattice-based probability, another parameter λ is introduced to aggregate the real probability of w issued from the city i, and the probability of the smoothed lattice probability. Finally the lattice-based per-city word probability can be formalized as

$$p'(i|w) = \lambda * p(i|w) + (1.0 - \lambda) * p'(lat|w),$$

where i is a city within the lattice lat.

Worth mentioning, in practice, we arbitrarily fix the size of lattice to 1 by 1 square degrees, which is about the size of a typical county or a large city.

4.2.3. Model-Based Smoothing. The final approach to smoothing takes into account the word models developed in the previous section for identifying C and α. Applying this model directly, where each word is distributed according to $Cd^{-\alpha}$, we can estimate a per-city word distribution as

$$p'(i|w) = C(w)d_i^{-\alpha(w)},$$

where $C(w)$ and $\alpha(w)$ are taken to be the optimized parameters derived from the real data distribution of words across cities. This model-based smoothing ignores local perturbations in the observed word frequencies, in favor of a more elegant word model (recall Figure 3). Compared to the data-driven geographic-based smoothing, model-based smoothing has the advantage of "compactness", by encoding each word's distribution according to just two parameters and a center, without the need for the actual city word frequencies.

4.2.4. Wave-Like Smoothing:. The term-localizing component works well for terms which have exactly one geographical center. However, some of the words cannot be simply represented by a single peak. Let us still take the word "Rockets" as an example: Rockets is the name of the NBA team in Houston, as well as a term frequently used in NASA which is also located in Houston. Thus people tweet the word "Rockets" the most frequently in the greater Houston area, but there are also "Rockets" associated with the University of Toledo in Ohio and with particular events (like the mysterious rocket launch off the coast of California in 2010).

To handle this multipeak issue, we can extend the one-peak spatial model to a multiple-peaks version. For each word, we generate a peak at each city where the word is issued. In addition, each peak at a city becomes a radioactive source, emitting wave-like impacts towards other cities over the map. The impacts from each peak (i.e., source) decreases exponentially as the distance from the location of the peak increases. Thus, the probability distribution for a word becomes an interwoven overlapping of thousands of one-peak distributions. We visualize the wave-like distribution for the word "Rockets" in Figure 5, and at least three relative high peaks can be identified. With this wave-like smoothing, the probability of a word w issued from city i can be formalized as

$$p(i|w) = \sum_{j \in S_c} \begin{cases} p(j|w) * (d(i,j) - r_j + 1)^{-\alpha(w)} & d(i,j) \geq r_j \\ p(j|w) & d(i,j) < r_j \end{cases},$$

where $p(j|w)$ denotes the estimated probability of word w issued from city j; $d(i, j)$ is the euclidean distance between city i and city j; r_j is the radius of the city j; and $\alpha(w)$ is the shrinking parameter of word w indicating how fast the impacting probability of the word w shrinks down when distance from the center increases. With the preceding equation above, we go through all the cities in the set of large cities S_c and sum up the impacts from each city. Locations inside the area of each source city will have the same probability as the city's $p(j|w)$, and as the distance from the source increases, the probability decreases exponentially. As a consequence, with the combinations of all the local words and all the cities, a highly overlapped probabilistic distribution is generated.

4.3. Social Refinement

So far, we explored predicting an individual Twitter user's geolocation based on her tweets alone. A natural hypothesis would be: given an unlocated user's tweets and a few of her unlocated friends and their tweets, can we improve the performance of predicting the user's location by incorporating evidence from these social ties? Thus, in

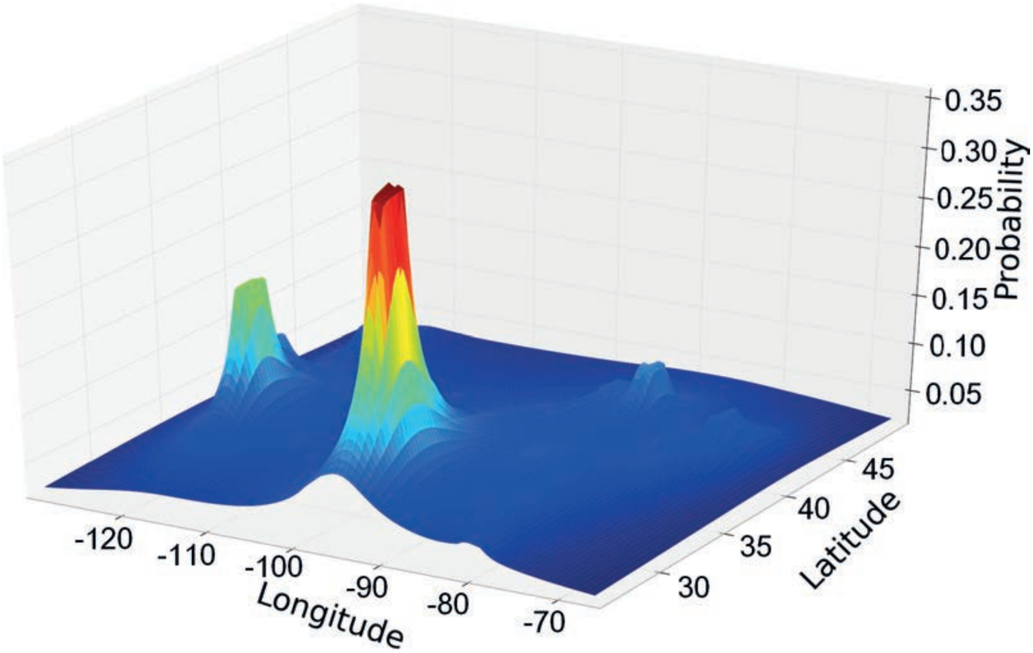


Fig. 5. Wave-like smoothing for word “Rockets”.

this section we explore the possibility of geolocating a user utilizing aggregated results from social relations. Our assumption is that users have a large portion of local friends (over 40% within 100 miles observed by McGee et al. [2011]), and strong connected (reciprocal) friends tend to live nearby to each other than friends with weak ties. The hope is that aggregates of location predictions from a user's social ties will provide additional evidence for refining the user's predicted geolocation.

For each user u, we have a collection of the user's latest tweets $S_{tweets}(u)$, a list of the user's n friends $list_{friends}(u) = \{f_j | 1 \leq j \leq n\}$, and a collection of tweets $S_{tweets}(f_j)$ from each friend f_j. Determining the appropriate choice of friends and the number of friends to consider is something we can study experimentally.

Given the setup, the *social refinement algorithm* for content-driven location estimation is as follows.

—First, we apply the baseline content-driven algorithm (the best performing content-driven algorithm so far: local filtering + neighborhood-based smoothing) to predict the location for each friend f_j of the user u's. Concretely, for each city i, we estimate a likelihood score $s_{likelihood}(i|S_{tweets}(f_j))$ for the friend f_j to be located in city i based on her tweets $S_{tweets}(f_j)$.

—Second, for each city i, we get an average likelihood score for user u to be located in the city i based the likelihood scores estimated from her friends' tweets, formalized as $s_{likelihood}(i|S_{tweets}(list_{friends}(u)) = \frac{\sum_{f_j \in list_{friends}(u)} s_{likelihood}(i|S_{tweets}(f_j))}{|list_{friends}(u)|}$.

—Third, for each city i, we predict the likelihood score for user u to be located in city i based on user u's tweets $S_{tweets}(u)$: $s_{likelihood}(i|S_{tweets}(u))$.

—Fourth, for each city i, the social inferred likelihood score for user u to be located in city i is: $s_{likelihood}(i|S_{tweets}(u), S_{tweets}(list_{friends}(u)) = \alpha * s_{likelihood}(i|S_{tweets}(list_{friends}(u)) + (1 - \alpha) * s_{likelihood}(i|S_{tweets}(u))$, where α is a predefined weight for predicted likelihood score from social relations.

Table IV. Impact of Refinements on User Location Estimation

Method	ACC	AvgErrDist (Miles)	ACC@2	ACC@3	ACC@5
Baseline	0.101	1773.146	0.375	0.425	0.476
+ Local Filtering (LF)	0.498	539.191	0.619	0.682	0.781
+ LF + Laplace	0.480	587.551	0.593	0.647	0.745
+ LF + State-Level	0.502	551.436	0.617	0.687	0.783
+ LF + Neighborhood	**0.510**	**535.564**	**0.624**	**0.694**	**0.788**
+ LF + Model-based	0.250	719.238	0.352	0.415	0.486
+ LF + Wave-Like	0.507	545.500	0.521	0.530	0.539

—Finally, according to the descending order of $s_{likelihood}(i|S_{tweets}(u), S_{tweets}(list_{friends}(u))$ we rank the cities, and consider the city with the highest likelihood score as the predicted location for user u.

In this way, the content-driven location estimation algorithm may be enhanced by incorporating the social ties of the underlying social network.

5. EXPERIMENTAL RESULTS

In this section, we detail an experimental study of location estimation with local tweet identification and smoothing. The goal of the experiments is to understand: (i) if the classification of words based on their spatial distribution significantly helps improve the performance of location estimation by filtering out nonlocal words; (ii) how the different smoothing techniques help overcome the problem of data sparseness; (iii) how the amount of information available about a particular user (via tweets) impacts the quality of estimation; and (iv) what impact social refinement has on content-driven location estimation.

5.1. Location Estimation: Impact of Refinements

Recall that in our initial application of the baseline location estimator, we found that only 10.12% of the 5,119 users in the test set could be geolocated within 100 miles of their actual locations and that the AvgErrDist across all 5,119 users was 1,773 miles. To test the impact of the two refinements—local word identification and smoothing—we update Algorithm 1 to filter out all nonlocal words and to update the per-city word probabilities with the smoothing approaches described in the previous section.

For each user u in the test set, the system estimates k (10 in the experiments) possible cities in descending order of confidence. Table IV reports the accuracy, average error distance, and accuracy@k for the original baseline user location estimation approach (*Baseline*), an approach that augments the baseline with local word filtering but no smoothing (+ *Local Filtering*), and then five approaches that augment local word filtering with smoothing – *LF+Laplace*, *LF+State-level*, *LF+Neighborhood*, *LF+Model-Based*, and *LF+Wave-Like*. Recall that accuracy measures the fraction of users whose locations have been estimated to within 100 miles of their actual location.

First, note the strong positive impact of local word filtering. With local word filtering alone, we reach an accuracy of 0.498 which is almost five times as high as the accuracy we get with the baseline approach that uses all words in the sampled Twitter dataset. The gap indicates the strength of the noise introduced by nonlocal words, which significantly affects the quality of user location estimation. Also consider that this result means that nearly 50% of the users in our test set can be placed in their actual city purely based on an analysis of the content of their tweets. Across all users in the test set, filtering local words reduces the average error distance from 1,773 miles to 539 miles. While this result is encouraging, it also shows that there are large estimation errors for many of our test users in contrast to the 50% we can place within

100 miles of their actual location. Our hypothesis is that some users are inherently difficult to locate based on their tweets. For example, some users may intentionally misrepresent their home location, say by a New Yorker listing a location in Iran as part of sympathy for the recent Green movement. Other users may tweet purely about global topics and not reveal any latent local biases in their choice of words. In our continuing work, we are examining these large error cases.

Continuing our examination of Table IV, we also observe the positive impact of smoothing. Though less strong than local word filtering, we see that Laplace, State-Level, Neighborhood, and Wave-Like smoothing result in better user location estimation than either the baseline or the baseline plus local word filtering approach. As we had surmised, the neighborhood smoothing provides the best overall results, placing 51% of users within 100 miles of their actual location, with an average error distance over all users of 535 miles.

Comparing state-level smoothing to neighborhood smoothing, we find similar results with respect to the baseline, but slightly better results for the neighborhood approach. We attribute the slightly worse performance of state-level smoothing to the regional errors introduced by smoothing toward the entire state instead of a local region. For example, state-level smoothing will favor the impact of words emitted by a city that is distant but within the same state relative to a word emitted by a city that is nearby but in a different state. We also find that wave-like smoothing performs slightly better than state-level smoothing and significantly better than the model-based smoothing due to its incorporation of multiple peaks per term, leading to more refined estimates (compared to the single-peak model).

As a negative result, we can see the poor performance of model-based smoothing, which nearly undoes the positive impact of local word filtering altogether. This indicates that the model-based approach overly smooths out local perturbations in the actual data distribution, which can be useful for leveraging small local variations to locate users.

To further examine the differences among the several tested approaches, we show in Figure 6 the error distance in miles versus the fraction of users for whom the estimator can place within a particular error distance. The figure compares the original baseline user location estimation approach (*Baseline*), the baseline approach plus local word filtering (+ *Local Filtering*), Wave-like smoothing approach (*LF+Wave-Like*), and then the best performing smoothing approach (*LF+Neighborhood*) and the worst performing smoothing approach (*LF+Model-based*). The x-axis identifies the error distance in miles in log-scale and the y-axis quantifies the fraction of users located within a specific error distance. We can clearly see the strong impact of local word filtering and the minor improvement of smoothing across all error distances. Interestingly, we see that the wave-like approach suffers from the problems of the model-based approach for small errors, but performs nearly as well as the neighborhood-based approach for larger errors. This suggests that the wave-like model has good potential to be further refined to eliminate the errors at small distance (introduced most likely due to the oversimplification of the model as compared to the more data-driven neighborhood-based approach). For the best performing approach, we can see that nearly 30% of users are placed within 10 miles of their actual location in addition to the 51% within 100 miles.

5.2. Capacity of the Estimator

To better understand the capacity of the location estimator to identify the correct user location, we next relax our requirement that the estimator make only a single location prediction. Instead, we are interested to see if the estimator can identify a good location somewhere in the top-k of its predicted cities. Such a relaxation allows us to appreciate

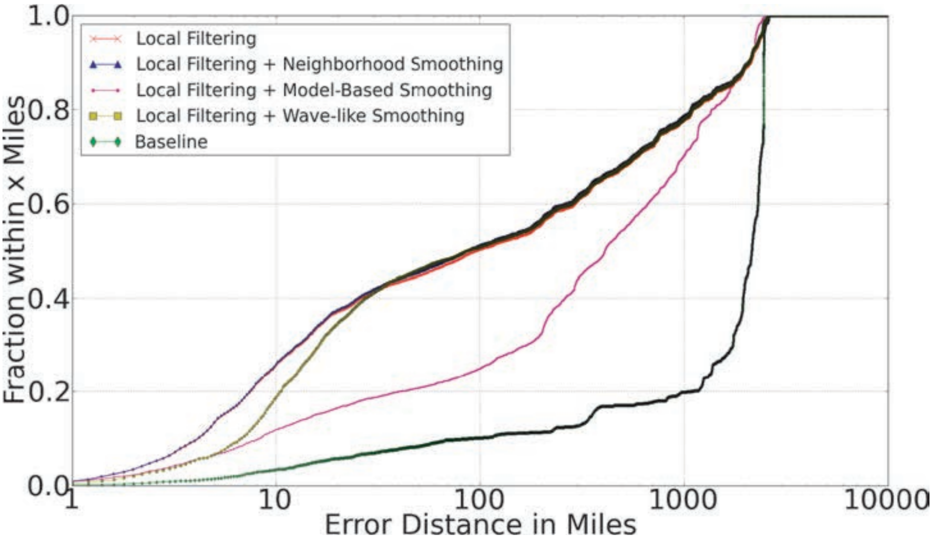


Fig. 6. Comparison across estimators.

if the estimator is identifying some local signals in many cases, even if the estimator does not place the best location in the top most probable position.

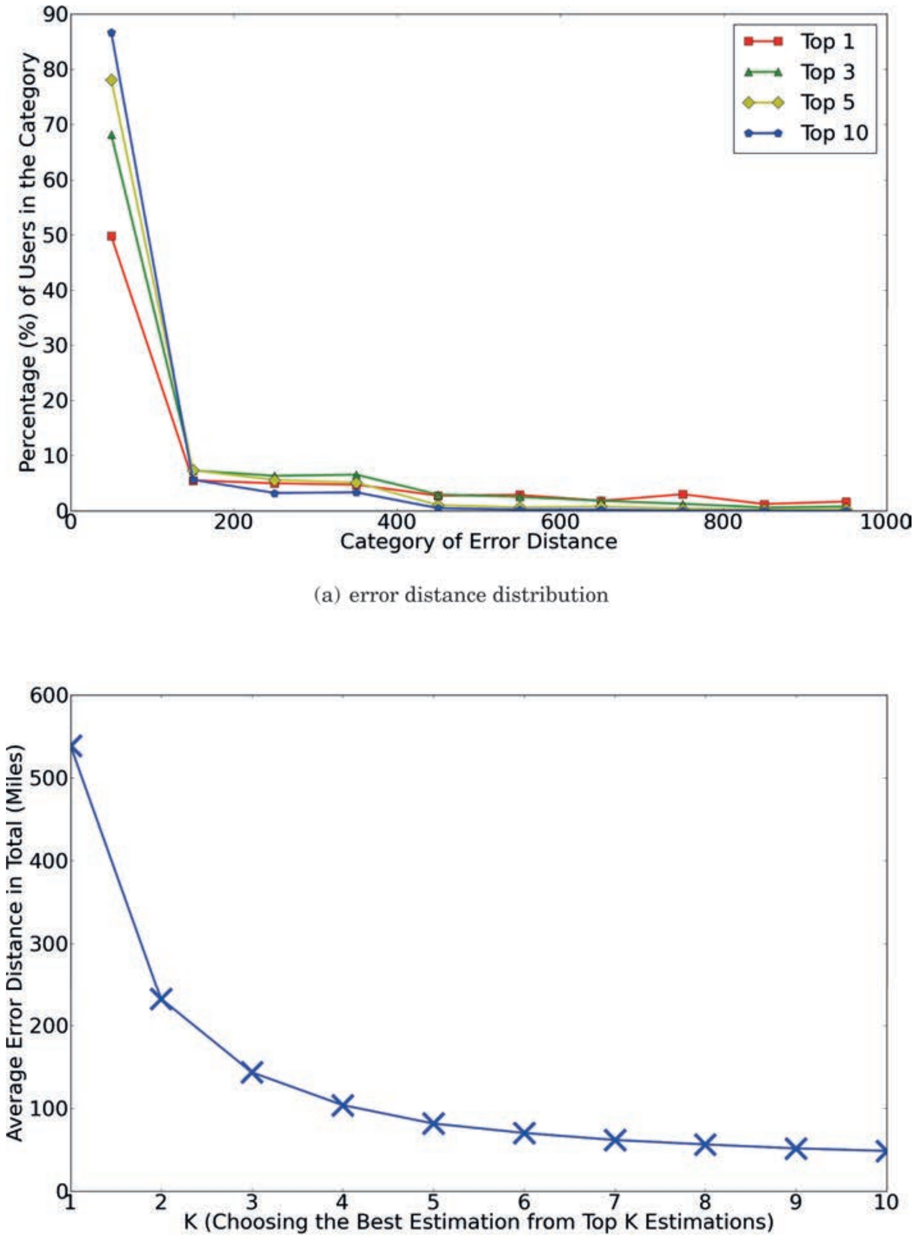
Returning to Table IV, we report the accuracy@k for each of the approaches. Recall accuracy@k measures the fraction of users located within 100 miles of their actual location, for some city in the top k predictions of the estimator. For example, for accuracy@5 for *LF+Neighborhood* we find a result of 0.788, meaning that within the first five estimated locations, we find at least one location within 100 miles of the actual location in 79% of cases. This indicates that the content-based location estimator has high capacity for accurate location estimation, considering the top-5 cities are recommended from a pool of all cities in the U.S. This is a positive sign for making further refinements and ultimately to improving the top-1 city prediction.

Similarly, Figure 7(a) shows the error distance distribution for varying choices of k, where each point represents the fraction of users with an error in that range (i.e., the first point represents errors of 0–100 miles, the second point 100–200 miles, and so on). The location estimator identifies a city in the top-10 that lies within 100 miles of a user's actual city in 90% of all cases. Considering the top-1, top-3, top-5, and top-10, we can see that the location estimator performs increasingly well. Figure 7(b) continues this analysis by reporting the average error distance as we consider increasing k. The original reported error of around 500 miles for the top-1 prediction drops as we increase k, down to just 82 miles when we consider the best possible city in the top-10.

5.3. Estimation Quality: Number of Tweets

An important question remains: how does the quality of estimation change with an increasing amount of user information? In all of our experiments so far, we have considered the test set in which each user has 1000+ tweets. But perhaps we can find equally good estimation results using only 10 or 100 tweets?

To illustrate the impact of an increasing amount of user data, we begin with a specific example of a test user with a location in Salt Lake City. Figure 8 illustrates the sequence of city estimations based on an increasing amount of user tweet data. With 10 tweets, Chicago has the dominant highest estimated likelihood score. With 100 tweets, several cities in California, Salt Lake City, and Milwaukee exceed Chicago. By 300 tweets, the



(a) error distance distribution

(b) average error distance

Fig. 7. Capacity of the location estimator: using the best estimation in the top-k.

algorithm geolocates the user in the actual city, Salt Lake City; however there is still significant noise, with several other cities ranking close behind Salt Lake City. By 500 tweets, the likelihood score of Salt Lake City increases dramatically, converging on Salt Lake City as the user data increases to 700 tweets and then 1000 tweets.

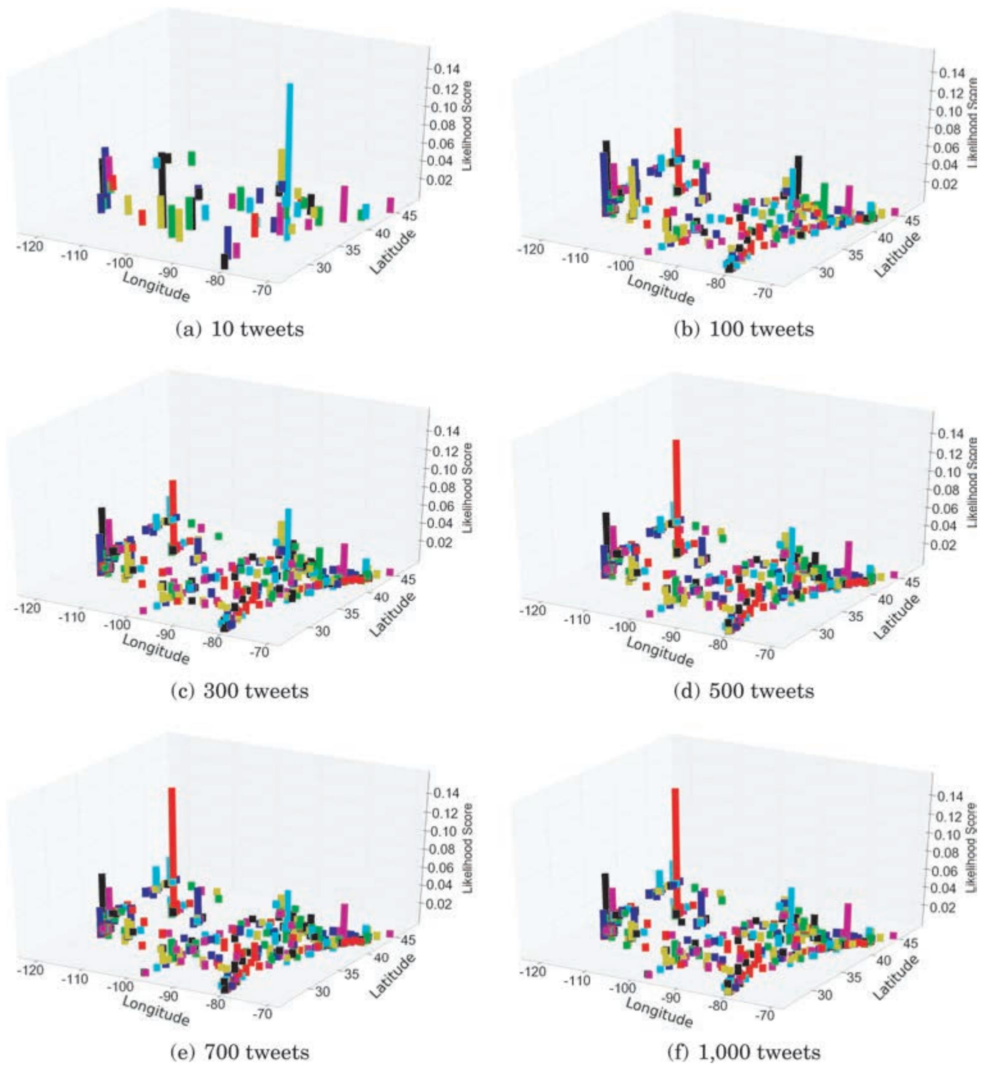
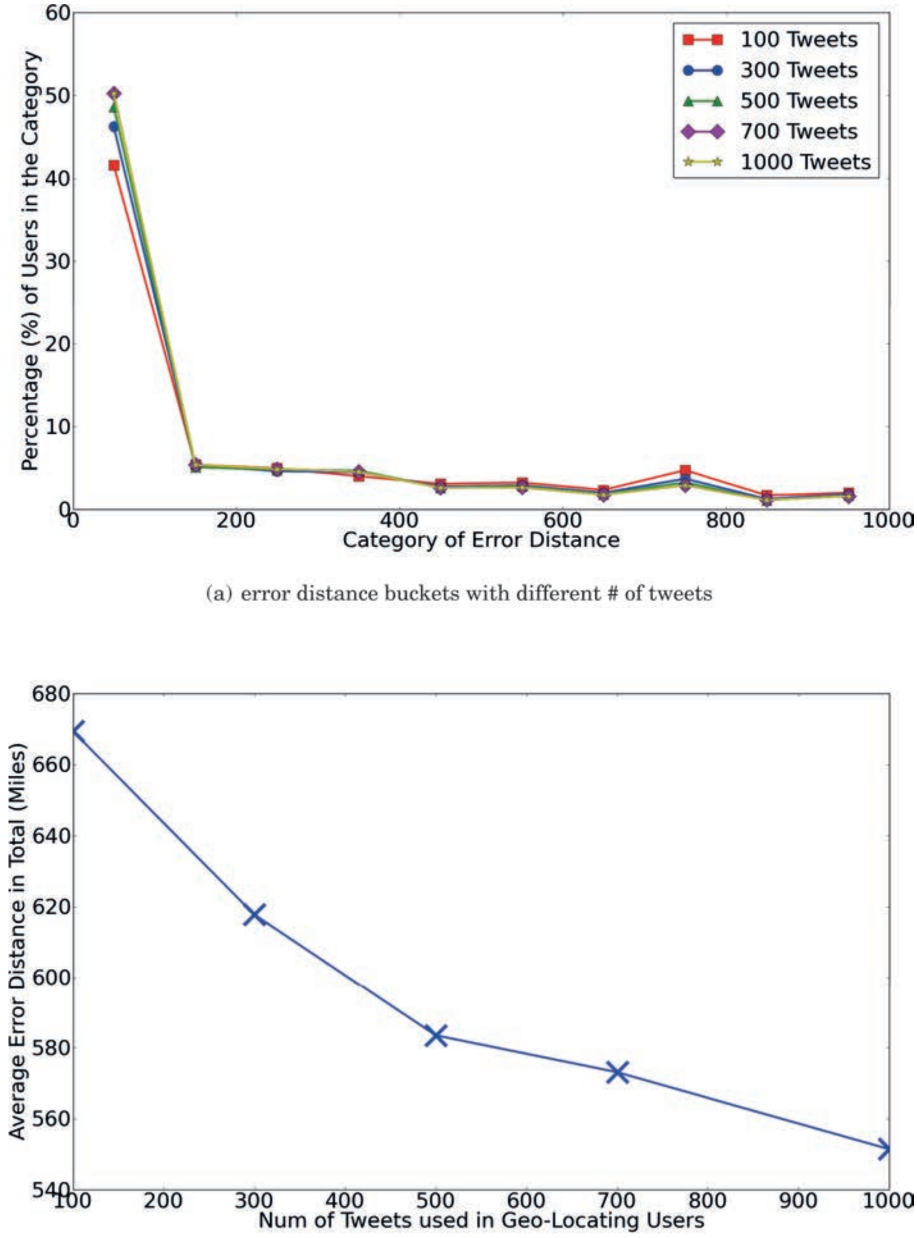


Fig. 8. Example: location estimation convergence as number of tweets increases.

To quantify the impact of an increasing amount of user information, we calculate the distribution of error distance and the average error distance across all of the test users based on the local word filtering location estimator relying on a range of tweets from 100 to 1000. Figure 9(a) shows the error distance distribution, where each point represents the fraction of users with an error in that range (i.e., the first point represents errors of 0–100 miles, the second point 100–200 miles, and so on). The errors are distributed similarly; even with only 100 tweets, more than 40% of users are located within 100 miles. In Figure 9(b), we can see that with only 100 tweets that the average error distance is 670 miles. As more tweets are used to refine the estimation, the error drops significantly. This suggests that as users continue to tweet, they “leak” more location information which can result in more refined estimation.



(a) error distance buckets with different # of tweets

(b) average error distance with different # of tweets

Fig. 9. Refinement of location estimation with increasing number of tweets.

5.4. Impact of Social Refinement

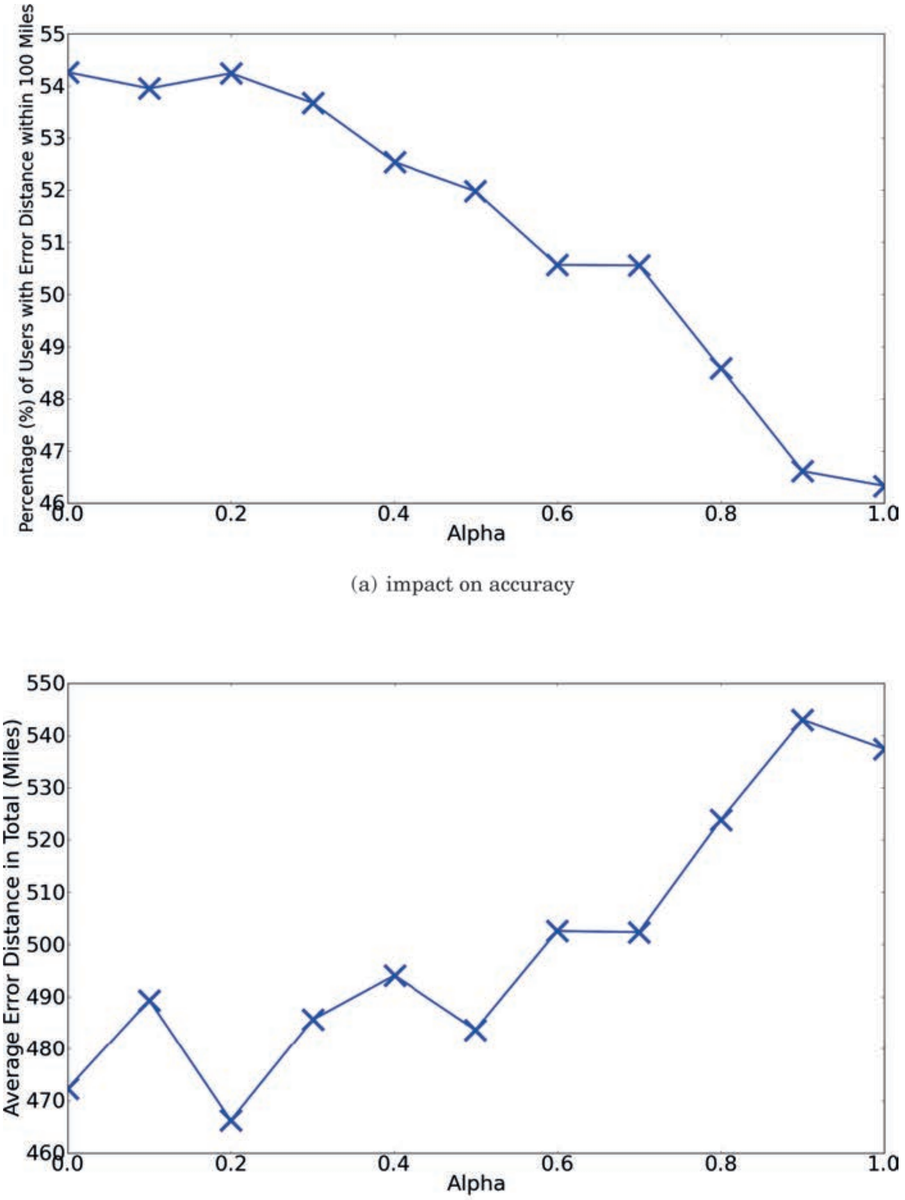
In this section, we explore the opportunity of incorporating social tie information into the content-driven location predictor (as described in Section 4.3). We are interested to understand whether social refinement can improve location estimation.

Using the test set described in Section 3, we randomly select 500 users, and crawl their social relations. 354 users out of the 500 users satisfy our precondition with at least 10 to 20 strong connected friends, where we define a strong connected friend of a user as one who is both following and followed by the user. For each of the 354 users, we crawl the user's strong connected friends, and their latest 500 tweets. In total, we have 3,137,233 tweets from 6,502 users who are strong connected friends of the 354 users. Recall that for each of the 354 users, we have the user's location in the form of latitude/longitude coordinates. Over this set of 354 users and their latest 1,000 tweets, we apply the best content-driven approach identified in the previous experiments: Local Words Filtering + Neighborhood Smoothing. We find that this content-driven approach (with no social refinement) results in an accuracy of 54.26% and an average error distance of 472.26 miles.

5.4.1. Quality of Estimator: Varying α. In the social refinement algorithm, the parameter α indicates the percentage of location estimate information for a target user based on the social ties of the target user versus the target user's own content. A value of 1.0 for α means the prediction is totally based on a user u's social relations, without any input from the user u's own tweets. On the other hand, a value of 0.0 for α means the prediction is based only on user u's tweets. We tune the parameter α from 0.0 to 1.0 with an interval of 0.1 to study to what extent more information from a user's social relations can help locate the target user. In Figure 10(a), the result shows that although we can get the highest accuracy either with an α value of 0.0 or 0.2, generally higher weights of social refinement (i.e., larger α values) produce worse results in terms of accuracy. Similarly, we show results for average error distance over different values for parameter α in Figure 10(b). The best average error distance we get is 466.20 miles with α value 0.2, which is a 1.28% increase over the nonsocial-refinement-based algorithm (472.26 miles). Interestingly, we see the same trend that incorporating some additional evidence from a target user's social ties results in better location estimation, but overreliance on social ties results in poorer location estimation. Surprisingly, even in the extreme case when none of a target user's content is used for location estimation (when $\alpha = 1.0$), the social ties alone still yield an estimate that is within 10% of the case when the target user's content is actually included in the estimator.

5.4.2. Quality of Estimator: Number of Tweets. In the second study of social refinement, we consider the impact of knowing more about the target user via additional tweets. Fixing $\alpha = 0.2$ based on the results from the previous experiment, we fix the number of tweets per social tie at 500, but vary the number of tweets for the target user from 0 to 1,000. Again, we see that even when no content is available for the target user, that the social-based estimator still achieves reasonable results (46% of users with error distance less than 100 miles; average error distance of 466 miles). As the amount of content for the target user increases, Figure 11(a) shows the improvement in accuracy, ultimately achieving around 54% accuracy. Similarly, Figure 11(b) shows how (after an initial increase from using a target user's own content) additional content from the target user results in an improved average error distance, which echoes the results described in our location estimation experiments without social refinement (recall Figure 7).

Together, these experiments on social refinement of content-driven location estimation suggest a great possibility for propagating estimated user locations along social ties to target users for whom we have no (or little) content information. We are also interested to explore more sophisticated variations of the social refinement algorithm, for example, by selectively considering only neighbors of a target user for whom we have high confidence (rather than including all neighbors) as well as PageRank-style iterative refinement approaches that aggregate not just one-hop social ties, but consider multihop social ties.



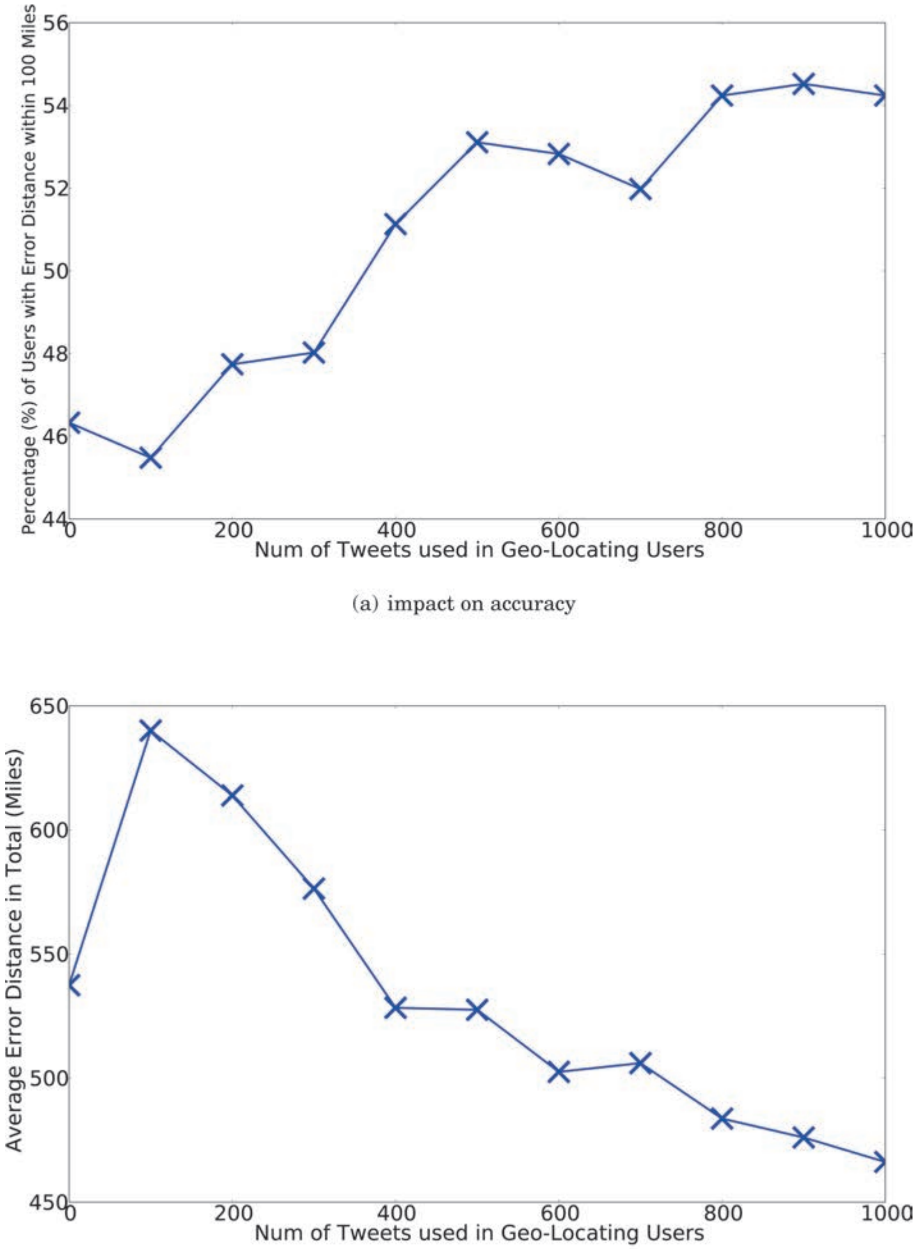
(a) impact on accuracy

(b) impact on average error distance

Fig. 10. Quality of socially refined estimator: tuning parameter α.

6. CONCLUSION

The promise of the massive human-powered sensing capabilities of Twitter and related microblogging services depends heavily on the presence of location information, which we have seen is largely absent from the majority of Twitter users. To overcome this location sparsity and to enable new location-based personalized information services, we have proposed and evaluated a probabilistic framework for estimating a microblog



(a) impact on accuracy

(b) impact on average error distance

Fig. 11. Capacity of the socially refined estimator: varying the number of tweets from the target user.

user's city-level location based purely on the publicly available content of the user's posts, even in the absence of any other geospatial cues. The content-based approach relies on two key refinements: (i) a classification component for automatically identifying words in tweets with a strong local geo-scope; and (ii) a lattice-based neighborhood smoothing model for refining a user's location estimate. We have seen how the location estimator can place 51% of Twitter users within 100 miles of their actual location.

As a purely data-driven approach, we anticipate continued refinement of this approach through the incorporation of more data (in the form of wider coverage of Twitter users and their associated tweets). Furthermore, we are interested to take the factor of population for cities into consideration when normalizing the probability for a word to occur in different cities, which is expected to further reduce the bias towards large cities in addition to local words filtering. We are also interested to further refine the combined social-tie- and content-based approaches, as well as incorporating temporal information into location estimation, to develop more robust estimators that can track a user's location over time.

REFERENCES

AMITAY, E., HAR'EL, N., SIVAN, R., AND SOFFER, A. 2004. Web-a-Where: Geotagging web content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

ATKINSON, K. 2007. Kevin's word list. http://wordlist.sourceforge.net

BACKSTROM, L., KLEINBERG, J., KUMAR, R., AND NOVAK, J. 2008. Spatial variation in search engine queries. In *Proceedings of the 17th International Conference on World Wide Web*.

BACKSTROM, L., SUN, E., AND MARLOW, C. 2010. Find me if you can: Improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*.

BERESFORD, A. R. AND STAJANO, F. 2003. Location privacy in pervasive computing. *IEEE Pervas. Comput. 2*, 1, 46–55.

CHEEMA, A. 2010. Twitter hits 20 billion tweets: Giga tweet. http://gopak.co.cc/social-media/twitter-socialmedia/twitter-hits-20-billion-tweets-giga-tweet/

CHENG, Z., CAVERLEE, J., AND LEE, K. 2010. You are where you tweet: A content-based approach to geolocating twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*.

CRANDALL, D. J., BACKSTROM, L., HUTTENLOCHER, D., AND KLEINBERG, J. 2009. Mapping the world's photos. In *Proceedings of the 18th International Conference on World Wide Web*.

CRANSHAW, J., TOCH, E., HONG, J., KITTUR, A., AND SADEH, N. 2010. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*.

FINK, C., PIATKO, C., MAYFIELD, J., FININ, T., AND MARTINEAU, J. 2009. Geolocating blogs from their textual content. In *Proceedings of the AAAI Spring Symposia on Social Semantic Web: Where Web 2.0 Meets Web 3.0*.

FRENI, D., VICENTE, C. R., MASCETTI, S., BETTINI, C., AND JENSEN, C. S. 2010. Preserving location and absence privacy in geo-social networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management.*

HEATHERLY, R., KANTARCIOGLU, M., AND THURAISINGHAM, B. 2009. Social network classification incorporating link type values. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*.

HUBERMAN, B. A., ROMERO, D. M., AND WU, F. 2008. Social networks that matter: Twitter under the microscope. Social Science Research Network Working Paper Series.

HURST, M., SIEGLER, M., AND GLANCE, N. 2007. On estimating the geographic distribution of social media. In *Proceedings of the International Conference on Weblogs and Social Media*.

JAVA, A., SONG, X., FININ, T., AND TSENG, B. 2007. Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD Workshop on Web Mining and Social Network Analysis (WebKDD/SNA-KDD '07)*.

JOHNSON, S. 2009. How twitter will change the way we live. *Time* 6/5/09.

KALNIS, P., GHINITA, G., MOURATIDIS, K., AND PAPADIAS, D. 2007. Preventing location-based identity inference in anonymous spatial queries. *IEEE Trans. Knowl. Data Engin. 19*, 12, 1719 –1733.

LEE, K., CAVERLEE, J., AND WEBB, S. 2010. Uncovering social spammers: Social honeypots + machine learning. In *Proceedings of the 33rd ACM SIGIR International Conference on Research and Development in Information Retrieval*.

LIN, J. AND HALAVAIS, A. 2004. Mapping the blogosphere in america. In *Proceedings of the Workshop on the Weblogging Ecosystem at the 13th International World Wide Web Conference*.

LINDAMOOD, J., HEATHERLY, R., KANTARCIOGLU, M., AND THURAISINGHAM, B. 2009. Inferring private information using social network data. In *Proceedings of the 18th International Conference on World Wide Web*.

MCGEE, J., CAVERLEE, J., AND CHENG, Z. 2011. A geographic study of tie strength in social media. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*.

MILLER, C. C. 2010. Sports fans break records on twitter. Blogs of *New York Times*.

PATRICK, K. AND KEVIN, B. 2009. The local business owner's guide to twitter. http://domusconsultinggroup.com/wp-content/uploads/2009/06/090624-twitter-ebook.pdf

PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., AND VETTERLING, W. T. 1986. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.

SAKAKI, T., OKAZAKI, M., AND MATSUO, Y. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the International Conference on World Wide Web*.

SERDYUKOV, P., MURDOCK, V., AND VAN ZWOL, R. 2009. Placing flickr photos on a map. In *Proceedings of the 32nd International and Development in Information Retrieval*.

TWITTER. 2007. Twitter's open api. http://apiwiki.twitter.com

USCENSUSBUREAU. 2002. Census 2000. u.s.gazetteer.http://www.census.gov/geo/www/gazetteer/places2k.html

WITTEN, I. H. AND FRANK, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann.

YAMAMOTO, Y. 2007. Twitter4j open-source library. http://yusuke.homeip.net/twitter4j/en/index.html.

YARDI, S. AND BOYD, D. 2010. Tweeting from the town square: Measuring geographic local networks. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*.

YI, X., RAGHAVAN, H., AND LEGGETTER, C. 2009. Discovering users' specific geo intention in web search. In *Proceedings of the 18th International Conference on World Wide Web*.

ZHENG, Y., ZHANG, L., MA, Z., AND MA, W. Y. 2011. Recommending friends and locations based on individual location history. http://research.microsoft.com/pubs/122435/recomfriend-zheng-published.pdf

ZONG, W., WU, D., SUN, A., LIM, E.-P., AND GOH, D. H.-L. 2005. On assigning place names to geography related web pages. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*.

Received May 2011; revised June 2012; accepted July 2012