7-11-2016

Location Inference of Social Media Posts at Hyper-Local Scale

Brian D. McClanahan
University of Connecticut, brian.mcclanahan@uconn.edu

Location Inference of Social Media Posts at Hyper-Local Scale

Brian McClanahan

B.S., Norfolk State University, 2013

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Connecticut

2016

APPROVAL PAGE

Master of Science Thesis

Location Inference of Social Media Posts at Hyper-Local Scale

Presented By
Brian Dillard McClanahan, B.S.

Major Advisor _____
Swapna Gokhale

Associate Advisor _____
Sanguthevar Rajasekaran

Associate Advisor _____
Karthik Konduri

Associate Advisor _____
Nicholas Lownes

Associate Advisor _____
Mohammad Maifi Hasan Khan

University of Connecticut

2016

Contents

1 Introduction and Motivation **1**

2 Background **4**
2.1 k-means 4
2.2 Logistic Regression 5

3 Geo-Location Approach **7**
3.1 Data Collection 7
3.2 Geographic Partitioning 8
3.3 Discriminative Classification 9
3.4 Feature Selection 10
3.4.1 χ^2 test 10
3.5 Information Gain Ratio 12
3.6 Geographical Density 12

4 Results and Discussion **15**
4.1 Evaluation Measures 15
4.2 Partitioning Method 17
4.3 Feature Selection Method 18
4.3.1 Number of Features 19
4.4 LR Classifier 20

5 Related Research **24**

6 Conclusions and Future Work **27**

Abstract

This paper describes an approach to infer the location of a social media post at a hyper-local scale based on its content, conditional to the knowledge that the post originates from a larger area such as a city or even a state. The approach comprises three components: (i) a discriminative classifier, namely, Logistic Regression (LR) which selects from a set of most probable sub-regions from where a post might have originated; (ii) a clustering technique, namely, k-means, that adaptively partitions the larger geographic region into sub-regions based on the density of the posts; and (iii) a range of techniques to extract a set of hyper-local words from the posts to be fed as features to the LR classifier. The approach is evaluated on a large corpus of tweets collected from Twitter over the NYC, Washington DC, and state of Connecticut regions. The results show that our approach can geo-locate tweets within 1.72 *km for NYC,* 12.5 *km for DC and* 37.00 *km for CT. These results from three geographically and socially diverse regions suggest that our approach outperforms contemporary methods that estimate locations within ranges of hundreds of kilometers. It can thus support a wide array of services such as location-based advertising, and disaster and emergency response.*

Chapter 1

Introduction and Motivation

Social media has gained a very prominent place in today's society. The wide and ubiquitous use as well as user base of social media services such as Twitter and Facebook have drawn the attention of several organizations for purposes such as event detection, public health monitoring, political sentiment analysis, targeted advertising, transportation planning, disaster management, and emergency response [20, 16, 18, 9]. The value and reach of such applications could improve significantly if they are supported with the ability to identify the location from where a post is shared. For example, location inference of social media posts could support applications visualizing and summarizing real-time events occurring in metropolitan areas [20]. The association of location to posts could also aid in the monitoring of illness outbreaks [16]. During elections post locations can be used to infer political sentiment on a regional bases [18]. Advertising organizations can tailor advertisements to the locations of users with geographical information derived from post, providing the users with advertisements which may be highly relevant to them. Applications for disaster response and emergency management can use location information to determine where resources should be allocated and more strategically coordinate efforts [9].

It would be ideal if the location of a post could be inferred through simple means such as the IP address of the device through which it is shared or from geo-tagging capabilities offered by most social media platforms that allow posts to be associated with their locations. However, device IP addresses

are often not made available and users prefer not to geo-tag their posts in order to protect their privacy; for example, only less than 1% of the tweets contain coordinates from the geo-tagging feature of Twitter [17]. Inferring the location of a post based on the home location field in a user's profile is also infeasible because of two issues. A user's home location may not coincide with the location of a post. Moreover, even when users (rarely) populate this location field it is usually too broad to be of any use or even fictitious [3]. Finally, the finest granularity at which several contemporary approaches infer location is that of a city [11, 8, 9, 10, 3], which cannot bring much value for most applications. An interesting question that then arises is whether precise geo-locations of these posts can be inferred through other means, such as the analyses of their content and/or metadata.

In this paper, we propose an approach to infer the location of a social media post at a hyper-local scale, when the larger region (such as a city) from which the post originates is known. We expect that many organizations may be able to predict a larger region of a post through their list of subscribers. Alternatively, contemporary approaches can also be used to predict such a broader location. In a suite comprising a hierarchy of location predictors, where the predictor at each level estimates the location at a particular granularity [11], our approach could be used as the last and finest predictor in the hierarchy. The methodology poses the geo-location problem as one of classification and uses a discriminative classifier, namely, Logistic Regression (LR) to select from a set of most probable sub-regions from where a social media post might have been shared. The larger region is geographically divided into sub-regions or classes for LR using k-means clustering based on how these posts are distributed within the region. Three techniques extract hyper-local words from the social media posts for use as features in the LR classifier. Our approach is evaluated on a large corpus of tweets collected from Twitter over NYC, Washington DC, and the state of CT. The primary evaluation measure, namely, mean error in the distance between predicted and actual locations suggests that the approach can correctly place tweets within 1.72 km for NYC, 12.5 km for DC and 37.00 km for CT regions. In other words, the approach can geo-locate tweets in an area that is on an average within 20% of the original, broader region's size regardless of the diverse geographical and social characteristics of the regions.

The rest of the paper is organized as follows: Section 3.1 describes the data used in the study. Section 3

details our geo-location approach. Section 4.1 defines the evaluation metrics. Section 4 discusses the results. Section 5 compares and contrasts related work. Section 6 concludes the paper with directions for future work.

Chapter 2

Background

Here we will describe the main techniques used in this work to achieve geolocation, which include the k-means and Logistic Regression algorithms.

2.1 k-means

The k-means algorithm is used to geographically partition larger regions into sub-regions. k-means is an algorithm commonly used to identify clusters of data points in a multidimensional space [2]. A cluster can be thought of a set of points which have low inter-point distances compared to points outside of the cluster. The user of the algorithm must first define the number of clusters to find in a data set, k. In the context of geographical k-means minimizes the within-cluster variance through implicit minimization of the following objective function, known as the distortion measure [2].

$$distortion = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{2.1}$$

In Equation (2.1), x_n and μ_k are both two-dimensional vectors that each contain latitude and longitude coordinates. μ_k is the center point or centroid for cluster k, x_n represents the coordinates for tweet n, and $r_{n,k}$ is an indicator variable which takes the value 1 if tweet n belongs to cluster k. N and K are the total number of tweets and clusters respectively. In general, the vectors x_n and μ_k can have arbitrary dimensionality.

The k-means algorithm functions as follows: In the first step, centroids are initialized. In the second step, each tweet is assigned to the closest centroid, where "closest" is defined in terms of Euclidean distance. In the third step, each centroid is updated to be the mean of all the tweets assigned to that cluster. The second and the third steps are repeated until the algorithm converges.

One common criticism of the k-means algorithm is that it can only find globular shaped clusters. For the purposes of this work however, this may not be a problem, in fact we conjecture that globular shaped clusters may lead to higher precision in location inference than elliptical ones. Another common criticism is that k-means is highly sensitive to centroid initialization. That is, the clusters retrieved from k-means are highly dependent on the choice of the initial centroids. In the Geographic Partitioning section we describe a technique used to overcome this difficulty.

2.2 Logistic Regression

In a data set where the data points are divided into two classes, the Logistic Regression (LR) algorithm can be used to model the posterior probability of a class given a data instance. This posterior probability can in turn be used to assign a data instance to a particular class by defining a threshold on the posterior probability. For example, a data instance can be assigned to class one if the posterior probability for class one is greater than 0.5 otherwise the data instance will be assigned to class two. Let t be a word count vector of a tweet that we wish to assign to some class (or sub-region), where entry i in the vector contains the number of times that word i occurs in the tweet. Then LR is a linear model which uses the following sigmoid function to directly model the posterior probability of a class given a tweet [2]

$$\sigma(\theta^t t) = \frac{1}{1 + exp(-(\theta^t t))} \tag{2.2}$$

In Equation (2.2), coefficients in vector θ weigh the features in t and $\theta^t t$ is the inner product of θ and t. When the classification is binary, $\sigma(\theta^t t)$ can be interpreted as $p(s_j|t)$ and $1 - \sigma(\theta^t t)$ as $p(\overline{s_j}|t)$, where s_j and $\overline{s_j}$ are sub-region j and all sub-regions other than j respectively. We achieve multi-class classification simply by training the binary classifier for each sub-region s_j and using a one vs. all

approach. Then s_j with the highest $p(s_j|t)$ is the predicted sub-region of tweet t. LR model is trained by iteratively adjusting θ to maximize the likelihood of the data by minimizing the following objective function with respect to θ [7]:

$$J(\theta) = \frac{1}{2}\theta^t\theta + C\sum_{i=1}^{N}\log(1+exp(-y_i\theta^t t_i)) \tag{2.3}$$

In Equation (2.3), N is the total number of training instances, $y_i \in \{-1,1\}$ is the class label of t_i, $C > 0$ is a penalty parameter which controls the importance of minimizing the second term, and $\frac{1}{2}\theta^t\theta$ is a regularization term which keeps θ from growing too large, preventing the classifier from overfitting.

Equation 2.3 can be minimized through the method of gradient descent. One issue with using gradient descent to minimize 2.3 is that this approach involves computing the gradient of a function which contains a summation over the entire data set. If the dimensionality of θ is large and N is also large (as is the case in this study) this process can be very expensive. In the Discriminative Classification section we discuss the solution employed to overcome this problem.

Chapter 3

Geo-Location Approach

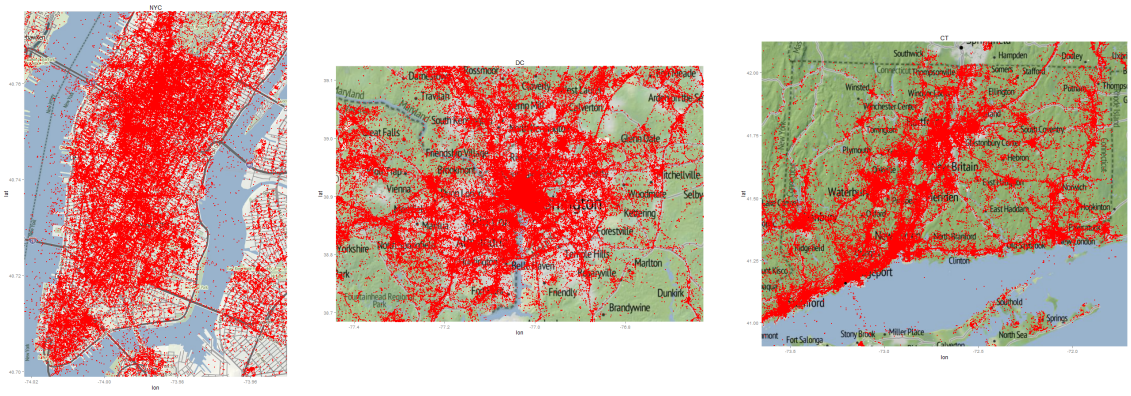
We approach the problem of geo-locating tweets by first *geographically partitioning* the larger region from which they are known to originate into sub-regions. Subsequently, we use Logistic Regression (LR) as a *discriminative classifier* to determine the sub-region from which a tweet is most likely to have been shared. Finally, we identify hyper-local words with strong ties to specific sub-regions as a set of features for the LR classifier using three *feature selection* techniques. In this section, we define these three key elements of our approach and describe our data collection process.

3.1 Data Collection

For the development and subsequent evaluation of our methodology, we collected large corpuses of tweets from three geographically and socially diverse regions, namely, NYC Manhattan (NYC), Washington DC (DC), and the State of Connecticut (CT). Table 3.1 summarizes the details of the tweet collection in the year 2013 for all the three regions.

Table 3.1. Data Collection: Regional Summary

Region	Days	Size (km^2)	Tweets	
			Volume	Density
NYC	69	51.29	574948	11209.74
DC	131	3452.57	1000000	289.63
CT	69	22101.02	950615	43.01



(a) NYC (b) DC (c) CT

Figure 3.1. Distribution of Tweets within Regions

Figures 3.1a, 3.1b, and 3.1c respectively depict that the distribution of tweets within the NYC, DC, and CT regions is non-uniform. CT region shows the highest skew because it is the largest and the most rural among the three. Most tweets in CT appear along major highways and in and around cities such as Hartford and New Haven. The tweet density is really high within the boundaries of Washington DC but becomes sparse on the outskirts. In NYC, the island of Manhattan has a fairly uniform and rich tweet density, but naturally the density off the coast of the island is low.

3.2 Geographic Partitioning

A simplistic way to define sub-regions is to uniformly partition the larger area into equally sized cells. Figures 3.1a and 3.2a illustrate how such uniform partitioning, which ignores the geographic spread of the tweets over NYC, easily leads to many sub-regions with sparse tweet densities. Therefore, we use the k-means algorithm [2] to define sub-regions based on the distribution of tweets. This process leads to sub-regions which are centered on clusters of tweet locations. Because we predict that the post location of a tweet is the center of the sub-region it is predicted to have come from, identifying these clusters and using them as subregions has the potential advantage of making location prediction more precise. In fact, the empirical results presented in the Results and Discussion chapter suggest that the use k-means leads to higher precision.

Different initializations of the centroids can cause the algorithm to converge to different local optima.

We thus use the k-means++ algorithm which randomly chooses centroids that are generally distant from each other [1]. Such clever initialization leads to clusterings that are fairly spread out rather than being cluttered, which occurs when centroid initialization is completely random. The resulting centroids are center points of sub-regions, and the class of a tweet is the center of the sub-region which is the closest to the geo-coordinates of the tweet. Figures 3.2a, 3.2b, 3.2c, 3.2d, 3.2e, and 3.2f clearly show how the centroids chosen by k-means are located in areas of fairly high tweet density compared to those identified by the uniform method. Thus, with the k-means method no sub-region is likely to have a sparse tweet density.

3.3 Discriminative Classification

We use Logistic Regression (LR) to identify the sub-region from where a tweet is most likely to have originated. As mentioned in the Background chapter, Logistic Regression models are trained by minimizing equation (2.3), which can be achieved through the method of Gradient Descent (GD) [14]. Gradient Descent iteratively updates θ by taking small steps in the direction of the negative gradient of $J(\theta)$ using the following rule:

$$\theta^{\tau+1} = \theta^{\tau} - \eta \nabla J(\theta^{\tau}) \tag{3.1}$$

In Equation (3.1), $J(\theta)$ sums over the entire data set. This can make GD iterations computationally expensive if N is large. An alternative approach is to use Stochastic Gradient Descent (SGD), where θ is updated after observing each tweet, which results in the following update rule [2]. SGD thus improves the speed while training on large data sets. This is advantageous in geo-locating tweets, which may call for frequent or real-time re-training of the classifier to adjust to temporal changes in the social media usage and the community.

$$J_i(\theta) = \frac{1}{2}\theta^t\theta + Clog(1 + exp(-y_i\theta^t t_i)) \tag{3.2}$$

$$\theta^{\tau+1} = \theta^{\tau} - \eta \nabla J_i(\theta^{\tau}) \tag{3.3}$$

3.4 Feature Selection

We pre-process the tweets by converting all the words to lower case and stripping punctuation. Additionally, all the words that appear in a stop word list are removed. Stop words are those that appear frequently but are not really associated with any theme such as “the”, “a”, or “that” [5].

To select features for the LR classifier, our naive approach uses all the relatively frequent words (excluding stop words) that occur more than some preset threshold in the entire corpus. For the sake of illustration, we set this threshold to 10, that is, a word must occur more than 10 times in the whole corpus for it to be considered as a feature. Thus, in this bag-of-words model, features are simply words that occur more than 10 times along with their frequencies. More formally, let the vocabulary V be the set of all words to be used as features. In the naive approach, V for NYC, DC, and CT regions is 26124, 32119, and 29115 words respectively. Each tweet is represented by a vector t of length $m = |V|$, where the l^{th} element corresponds to the word l and contains the number times it appears in t.

Not all frequent words, however, would be relevant to a tweet’s location. Therefore, using a large set of words as features that have no relation to a tweet’s location can introduce noise and degrade the performance of the classifier. Furthermore, not all tweets will contain geographic clues in their content and when they do not contain such clues it does not make much sense to try and determine the location of those tweets based on their words alone. Therefore, we extract words with geographical significance or “hyper-local” words and use these as features. The sections below describe the three feature selection techniques used to extract “hyper-local” words.

3.4.1 χ^2 test

The χ^2 is commonly used to determine if random events are independent [9]. The test works by quantifying the difference between what is observed and what would be expected if events were completely independent of one another. First a contingency table is constructed, like the one illustrated in table 3.2.

The first and second columns indicate whether a tweet was or was not posted from a subregion respectively and the first and second rows indicate whether a tweet did or did not contain word w respectively.

Table 3.2. Contingency Table

	subregion	not subregion
word	$O_{w,s}$	$O_{w,\overline{s}}$
not word	$O_{\overline{w},s}$	$O_{\overline{w},\overline{s}}$

O indicates the number of tweets that satisfy both criteria for its respective row and column. From the observations, the counts which would be expected if the events were independent can be computed [9].

$$E_{w,s} = P(w) * P(s) * N \tag{3.4}$$

$$P(w) = \frac{O_{w,s} + O_{w,\overline{s}}}{N} \tag{3.5}$$

$$P(s) = \frac{O_{w,s} + O_{\overline{w},s}}{N} \tag{3.6}$$

Here $E_{w,s}$ is the expected count of tweets which are from subregion s and contain word w, $P(w)$ is the probability that a tweet contains words w, $P(s)$ is the probability that a tweet was posted from subregion s, and N is the total number of observations. Expected counts for the other three events can be computed similarly. Given the observed and expected frequencies a quantity called the chi square value can be computed as follows.

$$\chi^2 = \sum_{m,n \in \{w,\overline{w}\} \times \{s,\overline{s}\}} \frac{(O_{m,n} - E_{m,n})^2}{E_{m,n}} \tag{3.7}$$

Note that this is a test evaluated per word and subregion. To select the words which will be used as features, words are ranked by there χ^2 values for each subregion. Words are then added to the feature set by iteratively choosing the highest ranked word of each region to be in the set, until the desired feature set size is reached. χ^2 is normally used as a means to get the p-value, which is the final value used to determine whether or not events are independent. However, χ^2, itself can be seen as quantifying the degree of independence between a word and a subregion, so it is directly used to determine if a word is hyper-local.

3.5 Information Gain Ratio

The Information Gain Ratio (*IGR*), comprises Information Gain normalized by intrinsic entropy [9].

$$IGR(w) = \frac{IG(w)}{IE(w)} \tag{3.8}$$

In the equation above $IGR(w)$ is the information gain ratio of word w, $IG(w)$ is the information gain, and $IE(w)$ is the intrinsic entropy. Information gain quantifies the decrease in the entropy of the sub-region probability distribution which results from the sub-region probability being conditioned on the presence or absence of a word. It is computed as follows.

$$IG(w) = H(s) - H(s|w) \tag{3.9}$$

Here $H(s)$ is the entropy of the subregion distribution and $H(s|w)$ is the entropy of the subregion distribution conditioned on word w.

Intrinsic entropy is a quantity which is usually higher for words which occur in many places and lower for words which occur in a few.

$$IE(w) = -P(w)\log w - P(\overline{w})\log \overline{w} \tag{3.10}$$

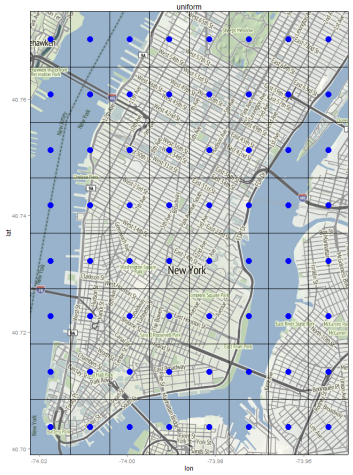
The ratio of $IG(w)$ and $IE(w)$ yields a measure which favors words that decrease subregion entropy and occur in few places.

3.6 Geographical Density

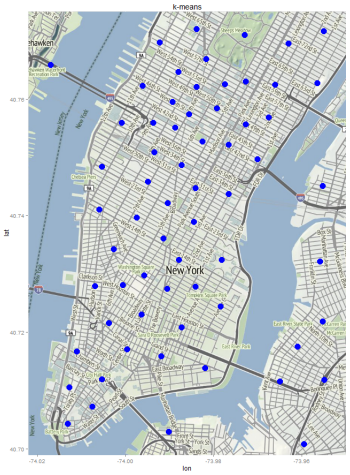
Geographical Density (*GeoDen*) identifies words with peaky location distributions, where the peaks tend to correspond to locations that are close together [9]. Like *IGR* the measure also has an affinity for words which occur in few locations. It is defined as follows.

$$GeoDen(w) = \frac{\sum_{s \in s'} P(s|w)}{|s'| \frac{\sum_{s_j, s_k \in s' s_j \neq s_k} haversine(s_j, s_k)}{|s'|-1}} \tag{3.11}$$

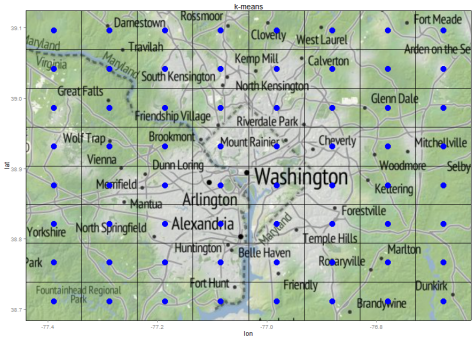
s' is the set of all subregions which contain word w and $haversine(s_j, s_k)$ is the great circle distance between the centers of subregions s_j and s_k. The numerator of $GeoDen(w)$ is the summation of all subregion probabilities in s' given word w. The denominator is the product of the average distance between all subregions in s' and the cardinality of s'. One should not include subregions in s' where the occurences of the word w are small, as doing so introduces noise. In this work, the approach used by [9] is used to define the set s'. All subregions subregions are ranked by $P(s|w)$ and then subregions are included into s' by order of rank, until the sum of $P(s|w)$ for all subregions included in s' exceeds some threshold r. We set r to be 0.2.



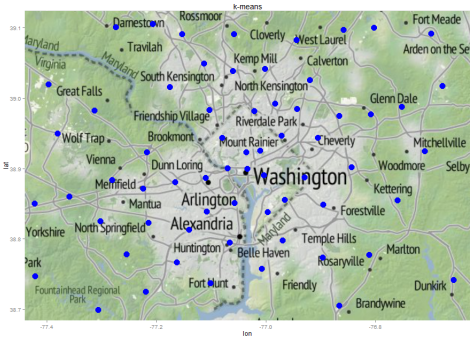
(a) NYC Uniform



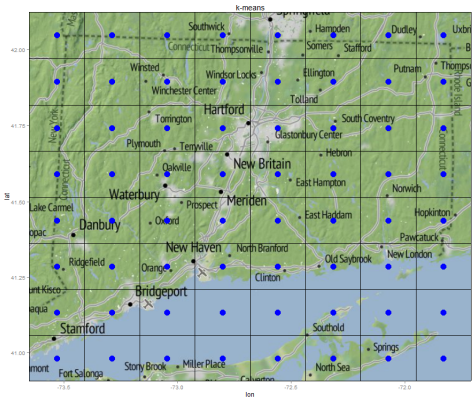
(b) NYC k-means



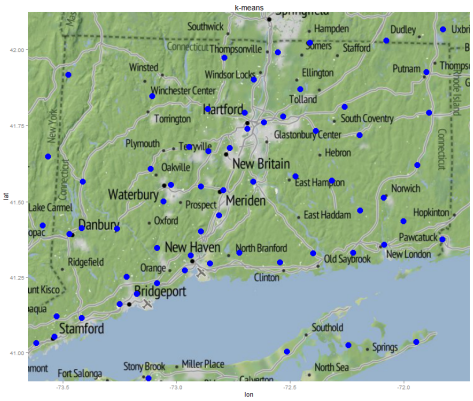
(c) DC Uniform



(d) DC k-means



(e) CT Uniform



(f) CT k-means

Figure 3.2. Uniform vs. k-means Partitioning into Sub-regions

Chapter 4

Results and Discussion

We used Python's Scikit Learn library to implement the *k*-means algorithm and the $SGDClassifier$ class to train the LR model using SGD [14]. We divided the entire corpus of tweets into training, validation, and evaluation subsets of sizes shown in Table 4.1. The training subset is used to train the LR classifier for each combination of geographic partitioning and feature selection method. The validation subset is used to compare the partitioning and feature selection techniques to determine those that give the best performance. Finally, these settings are used to asses the classifier on the evaluation subset. In four-fold cross validation, we repeat the process of dividing the corpus into three subsets, followed by training, validation, and evaluation four times. We then compute the average performance measures over the four runs.

Table 4.1. Sizes of Subsets

Region	# Training	# Validation	#Test
NYC	408095	83990	82863
DC	700000	150000	150000
CT	665431	142592	142592

4.1 Evaluation Measures

In this section, we describe the measures used to evaluate our approach.

- **Prediction accuracy (*PA*):** This measures the percentage of tweets for which the predicted sub-region is the same as the one from which the tweet originated.

- **Mean Distance Error (*MDE*):** This measures on an average the distance between predicted and actual locations, where the former is the centroid of the predicted sub-region. Because our goal is to infer precise locations of the tweets, *MDE* is our main measure of interest. It can be further decomposed into two components as in Equation (4.1), where MDE_c and MDE_i represent the mean distance errors for all correctly and incorrectly predicted tweets and are defined by Equations (4.2) and (4.3) respectively.

$$MDE = PA * MDE_c + (1 - PA) * MDE_i \tag{4.1}$$

$$MDE_c = \frac{\sum_{t_c \in T_c} haversine(pred(t_c), s_t)}{|T_c|} \tag{4.2}$$

$$MDE_i = \frac{\sum_{t_i \in T_i} haversine(pred(t_i), s_t)}{|T_i|} \tag{4.3}$$

In Equations (4.2) and (4.3), T_c and T_i are the sets of tweets for which the location is predicted correctly and incorrectly respectively and $pred(t)$ is the predicted sub-region for tweet t. MDE_c and MDE_i are useful because they measure how both correct and incorrect location predictions affect *MDE*. A classifier may often predict a sub-region that is albeit incorrect but nevertheless close to the actual one. MDE_i thus would be fairly low suggesting that even incorrect predictions may be potentially useful in inferring a tweet's location.

- **Coverage:** We use the three feature selection techniques to identify tweets which likely do not contain adequate geographical information. Thus, this metric measures the percentage of tweets that contain at least one hyper-local word and it varies depending on the feature selection method. While we use only covered tweets to train the classifier in order to eliminate noise, we handle the non-covered tweets in the test set using two approaches. In the first approach, non-covered tweets are predicted to arise from the most probable sub-region, which is the one with the most tweets. In the second approach, we filter the non-covered tweets and do not attempt to predict

their sub-regions. This leads to filtered versions of the performance measures which consider only covered tweets. We distinguish between the filtered and unfiltered versions of the measures with the superscript f. Thus, PA^f, MDE^f, ${MDE_c}^f$, and ${MDE_i}^f$ respectively denote the filtered versions of the measures PA, MDE, MDE_c and MDE_i.

4.2 Partitioning Method

To evaluate the influence of k-means and uniform partitioning on model performance in isolation, we did not employ any feature selection method in these experiments. Thus, all the words not in the stop words list, but which occur more than 10 times were used as features. Table 4.2 shows the metrics for both the partitioning strategies for a varying number of sub-regions. For all the three regions, k-means partitioning improves the mean distance error by about half a kilometer. For DC and CT regions, k-means partitioning yields the best mean distance error using a much smaller number of sub-regions than uniform partitioning. Using fewer classes could lead to better efficiency in training the LR model.

Table 4.2. k-means vs. uniform partitioning

Measure	16 sub-regions		25 sub-regions		36 sub-regions		49 sub-regions		64 sub-regions	
	Unif.	k-means	Unif.	k-means	Unif.	k-means	Unif.	k-means	Unif.	k-means
NYC Region										
PA	0.69	0.42	0.51	0.4	0.42	0.38	0.39	0.37	0.38	0.37
MDE	2.21	1.79	1.89	1.77	1.77	1.73	1.82	1.73	1.75	1.73
MDE_i	2.06	2.78	2.31	2.72	2.52	2.63	2.72	2.62	2.61	2.6
MDE_c	2.27	0.45	1.48	0.35	0.75	0.29	0.42	0.24	.34	0.2
DC Region										
PA	0.65	0.37	0.41	0.31	0.29	0.28	0.27	0.26	0.26	0.25
MDE	14.93	13.07	14.14	13.11	15.09	13.13	13.73	113.23	13.69	13.28
MDE_i	17.73	18.8	17.89	17.85	19.19	17.49	17.53	17.27	17.48	17.2
MDE_c	13.43	3.25	8.74	2.53	4.87	2.04	3.33	1.68	3.12	1.45
CT Region										
PA	0.56	0.3	0.37	0.28	0.28	0.25	0.26	0.23	0.25	0.22
MDE	39.61	39.02	40.16	39.19	40.81	38.81	29.58	39.41	39.69	40.33
MDE_i	39.87	53.14	48.58	52.32	52.07	50.57	50.54	50.35	50.58	50.75
MDE_c	39.42	6.64	25.61	5.13	11.44	4.12	7.7	3.53	7.4	2.94

We also note that for all the three regions uniform partitioning offers better accuracy than the k-means method. However, this advantage in accuracy is likely just an artifact of the skewed tweet density among sub-regions. When some sub-regions in the uniform partitions have many more tweets than others, the classifier can achieve higher accuracy just by predicting that the tweets come from those sub-regions

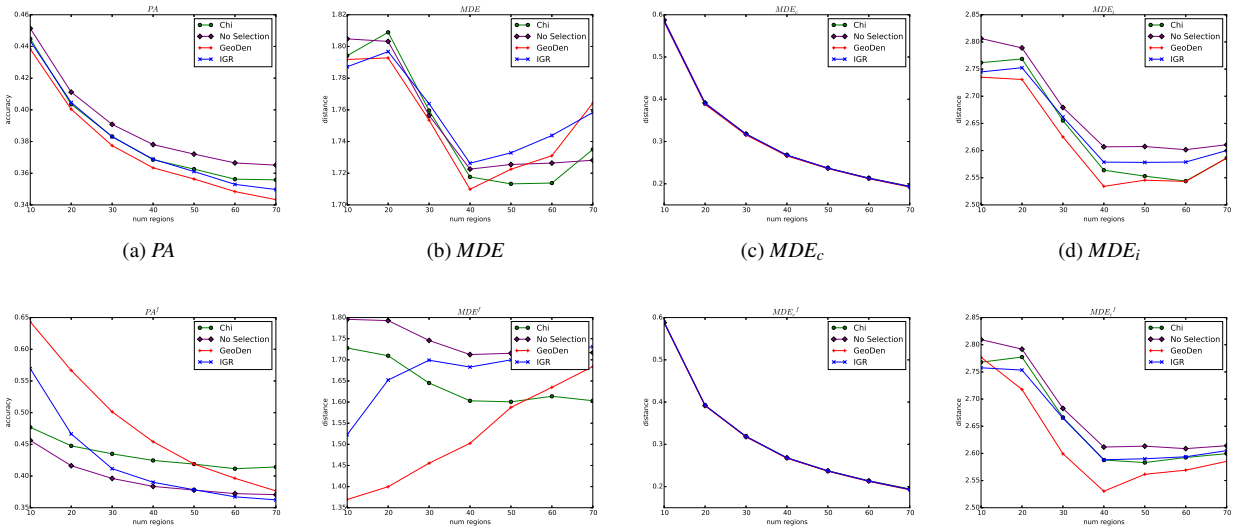
with higher tweet counts more often. It is also interesting that the k-means method has a superior MDE_c for all three regions. The difference in MDE_c for k-means and uniform partitioning is most dramatic when the number of sub-regions is low.

In summary, it appears that partitioning using k-means is advantageous over the uniform method, since the former results in an overall modest improvement in the mean distance error, and a substantial improvement in the mean distance error of tweets which are predicted correctly. Another potential advantage of the k-means over uniform partitioning is that the clusters identified by k-means could correspond to geographically significant social communities, which may be of interest to various applications. For example, the centroids in Figure 3.2 appear in cohesive neighborhoods such as Little Italy, Chinatown, Hudson Square, and Central Park in NYC.

4.3 Feature Selection Method

We compared the three feature selection methods using different numbers of sub-regions to analyze the joint effect of these two dimensions on model performance. Figures 4.1, 4.2, and 4.3 respectively show the measures for NYC, DC, and CT with filtered and unfiltered tweets. For each of these figures the size of the vocabulary V is set to be 10000. That is, the highest 10000 ranked words of the feature selection method are used as features. In each graph, the line labeled "No Selection" represents the naive approach with no feature selection. Even in the absence of feature selection, the coverage is not 100%, because elimination of stop words and words with low frequency can still lead to tweets with no features. *GeoDen* appears to be superior to the other two methods with respect to MDE and MDE^f, the unfiltered and filtered versions of the mean distance error. *GeoDen* has a lower MDE_i and ${MDE_i}^f$ than χ^2 and IGR, which could be attributed to *GeoDen*'s capability of highly ranking words with strong use in multiple sub-regions that are close together. Such words that are ranked high in multiple sub-regions can confuse the classifier as there can be multiple sub-regions which strongly correlate to a word. However, since these sub-regions are close to each other, mis-prediction is not as harmful because incorrectly predicted sub-regions are more likely to be close to the actual sub-region. The downfall of *GeoDen* is that it has lower coverage than the other two methods as shown in Figure 4.4. However, despite *GeoDen*'s low

coverage, it still achieves the lowest *MDE* when the number of sub-regions is 40 by just predicting that all non-covered tweets come from the most probable sub-region. With respect to accuracy it appears that no feature selection method offers good performance without filtering but *GeoDen* is the clear choice in the filtered case. For MDE_c and ${MDE_c}^f$ measures, all the methods offer very similar performance. In terms of coverage it appears that *IGR* is superior to χ^2 and *GeoDen* when the number of sub-regions is greater than 30. *GeoDen*, χ^2, and *IGR* all outperform no feature selection on MDE^f, ${MDE_i}^f$, and PA^f compared to when feature selection is employed. Given these results, *GeoDen* method with 40 sub-regions seems to offer the best performance.



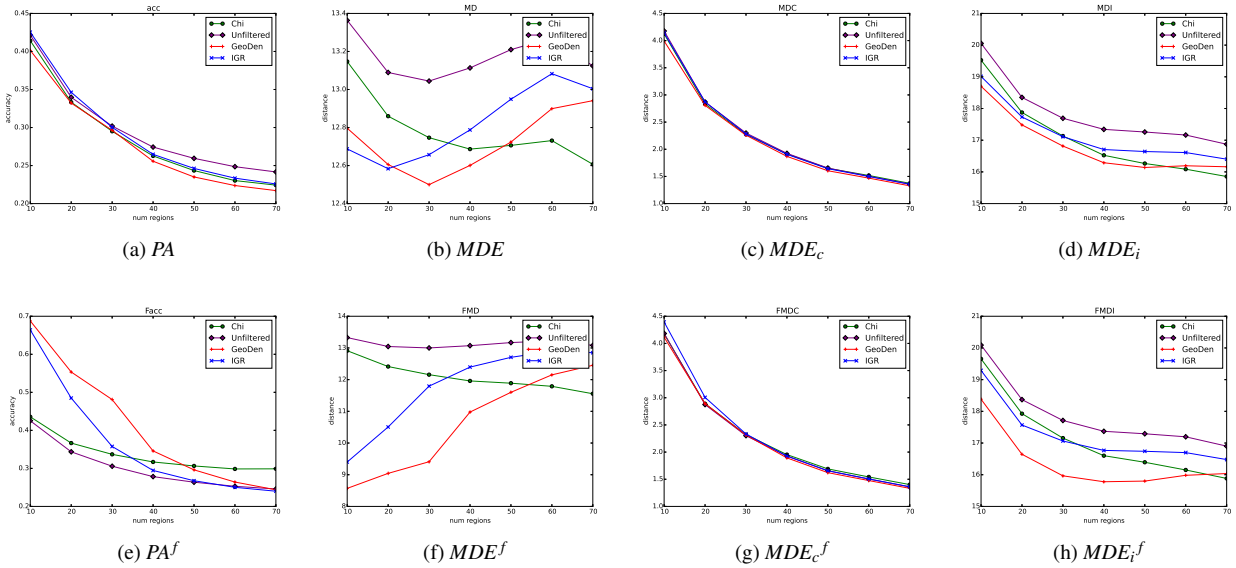
(a) *PA* (b) *MDE* (c) MDE_c (d) MDE_i

(e) PA^f (f) MDE^f (g) ${MDE_c}^f$ (h) ${MDE_i}^f$

Figure 4.1. Performance Measures for Feature Selection: NYC Region

4.3.1 Number of Features

Chi, *GeoDen*, and *IGR* were also compared using a varying number of features, these results are shown in figures 4.5 through 4.8. The plots show *GeoDen* outperforms *Chi* and *IGR* in terms of *MDE*, MDE^f, and PA^f for most feature set sizes. For all three selection methods the *MDE* decreases as more features are used for training, while MDE^f increases with the addition of features. This trend implies that adding features introduces noise which degrades the performance of the classifier. However, using a very small number of features leads to low coverage, which in turn can lead to high *MDE*. Thus there



(a) *PA* (b) *MDE* (c) MDE_c (d) MDE_i

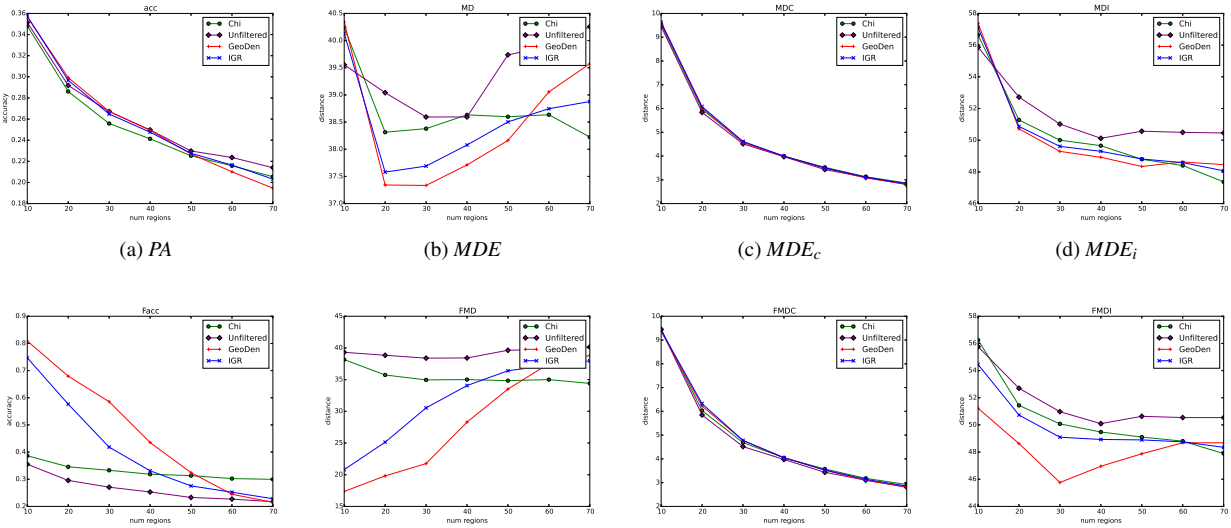
(e) PA^f (f) MDE^f (g) ${MDE_c}^f$ (h) ${MDE_i}^f$

Figure 4.2. Performance Measures for Feature Selection: DC Region

is a trade off between MDE^f and *MDE* which needs to be considered when selecting the number of features.

4.4 LR Classifier

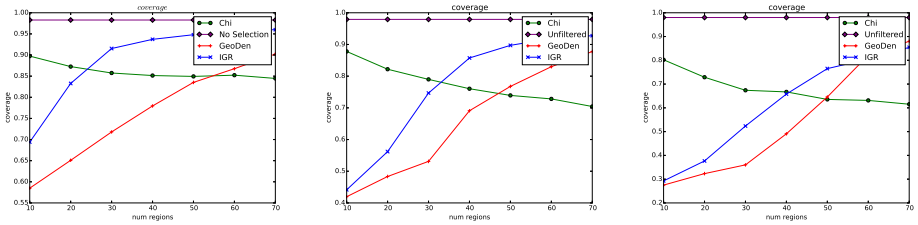
We evaluated the performance of the LR classifier for each region using 40 sub-regions and *GeoDen* method determined to give best performance as discussed in Sections 4.2 and 4.3. Table 4.3 shows the results of this evaluation averaged over four experimental runs. We noticed that *PA*, MDE_c, MDE_i, *MDE*, ${MDE_c}^f$ and MDE^f all tend to worsen with the size of the sub-regions. PA^f, which is the accuracy, however, appears to increase with the size of the sub-regions. This can likely be attributed to the centroids being more geographically spaced. Thus, the tweets in a sub-region are less influenced by neighboring clusters, substantially distinguishing between sub-regions. According to *MDE*, the predicted locations are on an average within 18% of the original area for NYC, 14% for DC, and 20% for CT regions. Except for ${MDE_c}^f$, all filtered measures outperform their unfiltered versions. However, the difference between ${MDE_c}^f$ and MDE_c is very low for all the three regions.



(a) *PA* (b) *MDE* (c) MDE_c (d) MDE_i

(e) PA^f (f) MDE^f (g) $MDE_c{}^f$ (h) $MDE_i{}^f$

Figure 4.3. Performance Measures for Feature Selection: CT Region

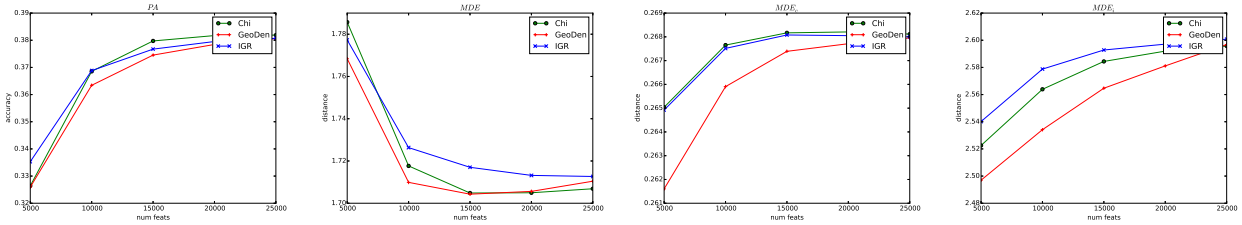


(a) NYC (b) DC (c) CT

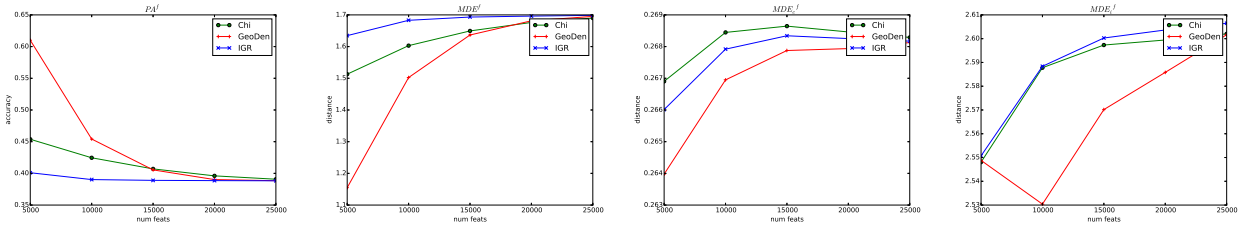
Figure 4.4. Coverage of Feature Selection Methods

Table 4.3. Performance of the LR Classifier

Measure	NYC	DC	CT
PA	0.36	0.3	0.28
MDE_c	0.26	2.26	4.92
MDE_i	2.56	16.82	49.79
MDE	1.72	12.5	37.38
PA^f	0.45	0.48	0.61
$MDE_c{}^f$	0.26	2.32	5.13
$MDE_i{}^f$	2.54	15.96	46.53
MDE^f	1.52	9.42	21.19
coverage	0.79	0.53	0.35

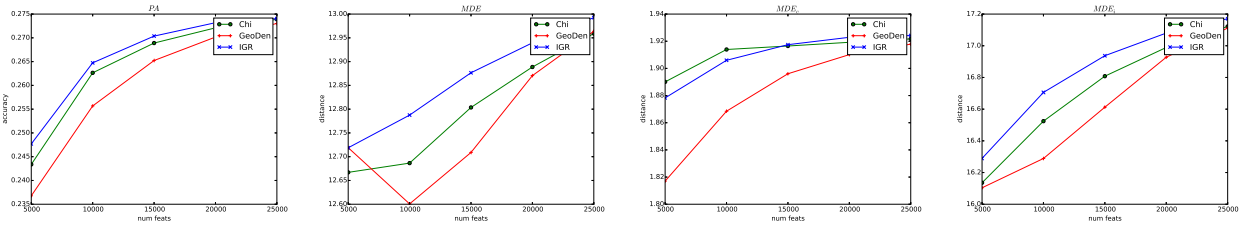


(a) PA (b) MDE (c) MDE_c (d) MDE_i

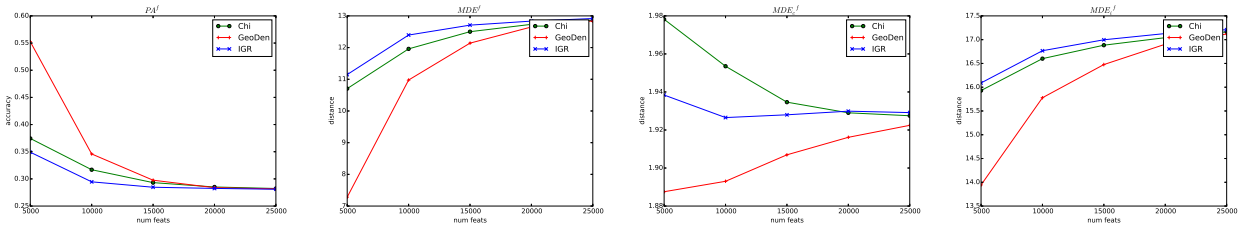


(e) PA^f (f) MDE^f (g) ${MDE_c}^f$ (h) ${MDE_i}^f$

Figure 4.5. Performance Measures for Number of Features: NYC Region

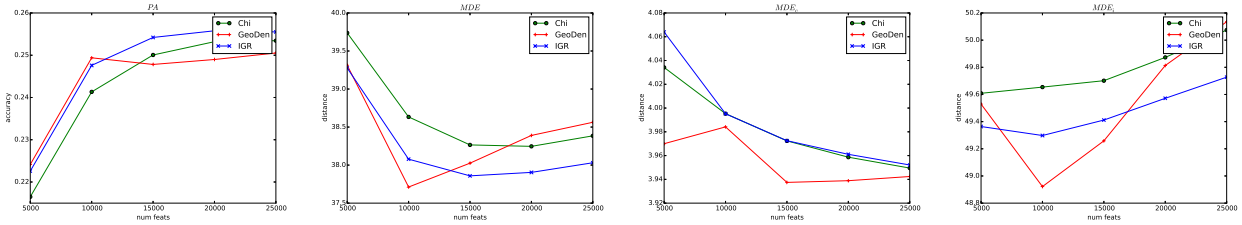


(a) PA (b) MDE (c) MDE_c (d) MDE_i

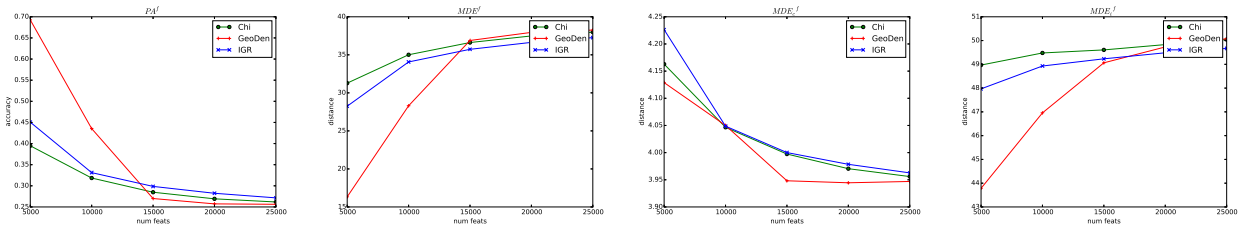


(e) PA^f (f) MDE^f (g) ${MDE_c}^f$ (h) ${MDE_i}^f$

Figure 4.6. Performance Measures for Number of Features: DC Region

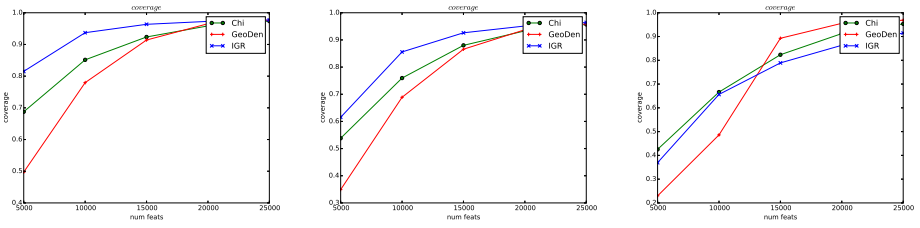


(a) PA (b) MDE (c) MDE_c (d) MDE_i



(e) PA^f (f) MDE^f (g) ${MDE_c}^f$ (h) ${MDE_i}^f$

Figure 4.7. Performance Measures for Number of Features: CT Region



(a) NYC (b) DC (c) CT

Figure 4.8. Coverage for Number of Features

Chapter 5

Related Research

In this section, we compare contemporary geo-location approaches along the following three dimensions:

Geographic Scope: Most methods locate posts within much broader regions than we consider here such as countries or the entire world even [19, 12, 6]. However, Flatow *et. al.* [8] perform hyper-local geo-location as we do, but limit their analysis to the NYC area. This approach relies on the identification of geo-specfic n-grams. An n-gram is determined to be geo-specific via an iterative process which fits a 2-dimensional geographical Gaussian to the n-gram, removes outliers outside of 2 standard deviations, and repeats. If a certain criteria is met during this iterative process then the n-gram is deemed geo-specific, otherwise if the maximum number of iterations is reached the n-gram is deemed not geo-specific. Location inference is done by identifying geo-specific n-grams in posts and associating the post with the mean of the n-gram. The results from this approach are determined using a larger region of NYC than considered in this paper. Evaluation is performed on different types of data sets such as tweets posts from Androids, I-phones, and Foursquare check ins. Overall it appears that the mean distance error is high for this approach and we achieve one much lower.

Modeling Approach: Most research, including our own [5], employs generative models to geo-locate posts. In some works, location inference is a by-product of a model designed to reveal spatiotemporal

themes in the twitter stream. The authors of [12] created a model, which aims reveal relationships between space, time, topics, and webblogs. A probabilistic model is used which defines a dependency of posted content on the time and location at which it was posted. A document is modeled as a sample of words drawn from k different topics (or distributions over words). The topics chosen are dependent upon both the time and location of the post. [10] also models the relationship between geography and posted content. Their model assumes that the topics an author chooses to write about are drawn from an additive model in which the components are a global distribution over topics, a user based distribution over topics, and a regional distribution. Location inference could be achieved by this group with an accuracy of about 120km. [6] developed a generative model which considers the effect of topic and location on posted content as well. In this work, pure base topics are thought to be corrupted by geography, thereby producing region specific versions of each base topic. This phenomenon is modeled by cascading topic models. A mean distance error of about 500km is reported.

Feature Categories: Some methods utilize more than just post content to infer location. A few approaches incorporate social network information [15] [4]. [15] infers friendships from different features, such as vocabulary similarity and location co-occurence. Determined friendships are then used in a dynamic Bayesian Network to determine user location. The Bayesian network considers only a discrete set of possible locations for each user, which are determined by previous locations the user was known to visit. [4] study the effect of friendship on human mobility patterns. Their model is based on the idea of a "check in", which is defined to be any event where the user makes his or her location known. If a user makes a non-social check in then the probability of a user's location is given by a mixture of Gaussians in which the components are home and work locations. The prior over the components is given by the time of day. If the check in is social, then the probability of a user checking in to a certain location is dependent on proximity of that location to locations visited by friends of the user on the same day. [17] infer post location by, stacking geographical polygons of spacial indicators such as time zone, location field from user profile, website links, and places identified via toponym resolution. About 750 km was reported as the mean distance error, considering tweets from the entire U.S. Some works [8, 5] use n-grams extracted from the content.

The major distinctions between our geo-location approach and other contemporary methods include: (i) location inference at a hyper-local scale; (ii) k-means for dynamic geographic partitions; (iii) discriminative LR classifier for prediction; and (iv) feature selection to extract relevant words. We utilize the knowledge that tweets come from some broad region such as a state or a city to narrowly estimate their location. Within this broad region, we wisely define sub-regions based on tweet densities rather than forming arbitrary or uniform partitions [19]. Because our primary objective is to infer the location of posts rather than identifying any spatial and/or temporal themes [12, 6], we choose a discriminative model because they generally perform better for classification tasks over generative models. Finally, feature selection offers our model a two-fold advantage. First, we can identify and geo-locate only those tweets that contain geographically relevant content unlike simply noting that tweets that lack contextual features is a limitation of the model [8]. Second, selection eliminates noisy features and boosts model performance.

Chapter 6

Conclusions and Future Work

In this paper, we propose a methodology for adaptive, hyper-local location inference of tweets. The location is predicted by associating tweets with pre-defined sub-regions through the use of a Logistic Regression (LR) classifier. k-means partitioning is used to define these sub-regions considering the distribution of tweets. Three feature selection methods, namely, χ^2, Information Gain Ratio (*IGR*) and Geographical Density (*GeoDen*) are explored to enhance performance by extracting hyper-local words to be fed as features to the LR classifier. Evaluation using large corpuses of tweets from NYC, Washington DC, and state of CT regions suggest that k-means clustering and *GeoDen* boost the performance of the LR classifier. The classifier predicts the location of tweets on an average within 1.72 km for NYC, 12.5 km for DC, and 37.00 km for CT regions. Thus, the classifier shows promise in geo-locating tweets in three geographically and socially diverse regions.

Our future research will explore additional features such as the time of a post, user's profile and social network information such as the home locations of a user's friends or followers to infer location. We also plan to use transfer learning methods [13] to identify and weigh those tweets in a training set which are most similar to those that will be seen in the application of the model, for example, if the model is to be used in emergency response then it must be trained using tweets describing similar events.

Bibliography

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proc. of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[2] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[3] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based approach to geo-locating Twitter users. In *Proc. of Intl. Conf. on Information and Knowledge Management*, pages 759–768, 2010.

[4] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *Proc. of Intl. Conf. on Knowledge Discovery and Data Mining*, pages 1082–1090, 2011.

[5] D. Doran, S. Gokhale, and A. Dagnino. "Discovering perceptions in online social media: A probabilistic approach". *Intl. Journal of Software Engineering and Knowledge Engineering*, 24(9):1273–1299, November 2014.

[6] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *Proc. of the 2010 Conf. on Empirical Methods in Natural Language Processing*, pages 1277–1287. Association for Computational Linguistics, 2010.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[8] D. Flatow, M. Naaman, K. E. Xie, Y. Volkovich, and Y. Kanza. On the accuracy of hyper-local geotagging of social media content. *arXiv preprint arXiv:1409.1461*, 2014.

[9] B. Han, P. Cook, and T. Baldwin. Text-based twitter user geolocation prediction. *J. Artif. Intell. Res.*, 49:451–500, 2014.

[10] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the Twitter stream. In *Proc. of Intl. Conf. on World Wide Web*, pages 769–778, 2012.

[11] J. Mahmud, J. Nichols, and C. Drews. Home location identification of Twitter users. *arXiv preprint arXiv:1403.2345*, 2014.

[12] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proc. of the 15th Intl. Conference on World Wide Web*, pages 533–542. ACM, 2006.

[13] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *Proc. of the Fifth ACM Intl. Conf. on Web search and Data Mining*, pages 723–732. ACM, 2012.

[16] A. Sadilek, H. A. Kautz, and V. Silenzio. Modeling spread of disease from social interactions. In *ICWSM*, 2012.

[17] A. Schulz, A. Hadjakos, H. Paulheim, J. Nachtwey, and M. Mhlhuser. A multi-indicator approach for geolocalization of tweets. In *Proc. of Intl. Conf. on Web and Social Media*, 2013.

[18] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.

[19] B. P. Wing and J. Baldridge. Simple supervised document geolocation with geodesic grids. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 955–964. Association for Computational Linguistics, 2011.

[20] C. Xia, R. Schwartz, K. Xie, A. Krebs, A. Langdon, J. Ting, and M. Naaman. Citybeat: real-time social media visualization of hyper-local city data. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 167–170. International World Wide Web Conferences Steering Committee, 2014.