Locality-adapted Kernel Densities for Tweet Localization

Ozer Ozdikis
NTNU
Trondheim, Norway
ozer.ozdikis@ntnu.no

Heri Ramampiaro
NTNU
Trondheim, Norway
heri@ntnu.no

Kjetil Nørvåg
NTNU
Trondheim, Norway
noervaag@ntnu.no

ABSTRACT

We propose a location prediction method for tweets based on the geographical probability distribution of their terms over a region. In our method, the probabilities are calculated using Kernel Density Estimation (KDE), where the bandwidth of the kernel function for each term is determined separately according to the location indicativeness of the term. Prediction for a new tweet is performed by combining the probability distributions of its terms weighted by their information gain ratio. The method we propose relies on statistical approaches without requiring any parameter tuning. Experiments conducted on three tweet sets from different regions of the world indicate significant improvement in prediction accuracy compared to the state-of-the-art methods.

KEYWORDS

Location prediction, Kernel density estimation, tweet localization

1 INTRODUCTION

Geographical information in terms of latitude-longitude associated with tweets represents the geographical origin where a tweet is posted from. However, such explicitly geotagged tweets constitute only a small portion of all tweets (around 1-3%). Therefore, predicting tweet locations using other information in tweets, primarily the tweet text itself, has been the objective of numerous recent studies, e.g., [3, 5, 8–10, 15].

Text-based approaches usually model the region of interest as a grid and apply a document classification method to estimate the most probable grid cell for a tweet [3–5]. In this work, considering terms in tweets as distinct sources of geographical evidence, we present a location estimation method that builds term probability distributions over the region according to a training set. Probability distribution for each term is calculated by using a term-specific setting of Kernel Density Estimation (KDE) [12]. We hypothesize that each term should have a different density estimation setting in consistence with its location indicativeness. In other words, probability distributions of highly local terms (e.g., city/town names) should be concentrated on specific areas, whereas more common words (e.g., stop-words) should have a more dispersed probability distribution over the entire region. After these term-level probabilities are modeled in a training stage, the location prediction for a new tweet is performed according to a weighted combination of probability distributions of its terms.

The primary contributions of our work can be summarized as follows: 1) we investigate the use of kernel density estimators to analyze geographical distributions of terms in tweets, and propose a fine-grained location prediction method based on integrated densities of terms, 2) our method relies on statistical techniques to obtain term-specific KDE settings based on location indicativeness of the terms without requiring parameter tuning, 3) we present a weighing method for the combination of probability distributions to obtain higher prediction accuracies.

The remainder of this paper is organized as follows: We present a summary of related work in Section 2, and describe our proposed location estimation method in Section 3. Section 4 is devoted to our evaluation results and discussions. Finally, we conclude the paper in Section 5.

2 RELATED WORK

Recent efforts for text-based geolocation of tweets apply various information retrieval and machine learning techniques [2, 8, 9, 15]. Multinomial Naive Bayes (MNB) and Kullback-Leibler (KL) divergence are among the most widely used methods [3–5], and further enhancements by selecting local terms are also proposed to improve their accuracy [3, 10, 13]. For example, clustering and dispersion tendency in term co-occurrences are investigated in [10]. In other recent studies, [1] estimates users' home locations by finding spatial word usage probabilities with Gaussian Mixture Models (GMM). Another method using weighted sum of GMMs for tweet localization is presented in [11].

Applying KDE for spatial analysis in social networks has recently received considerable attention. Its advantages over GMMs are discussed in [7, 14]. An adaptation of KDE to MNB classifier and KL-divergence measure for text-based geolocation was introduced in [5]. In [13], the authors proposed a term selection technique using KDE to geotag Flickr photos and Wikipedia articles. In [7], a mixture-KDE approach was applied for modeling and predicting individuals' locations according to their activity history. Similarly, in [14], user check-ins were analyzed using KDE to make location recommendations.

Our approach differs from previous methods by modeling the geographical distribution of each term according to their locality-adapted kernel density estimators. Probability distributions, which are calculated as integrated densities, are also weighted and combined based on term localities measured by information gain ratio. We would like to note that, although we use only the terms in tweet contents, our model can be extended to include the distributions of additional tweet features as well.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210109

3 LOCATION PREDICTION USING KDE

In a grid-based model where the region of interest is discretized into smaller grid cells, our location prediction method finds the most probable grid cell for a tweet according to a combination of probability distributions of its terms. We first describe how we determine the probability distributions of terms over the grid in a training stage. Then, we explain how we combine these probabilities to perform location prediction for a new tweet. In the remainder of this paper, we use $x{=}\langle t_x, l_x\rangle$ to represent a geotagged tweet, where t_x denotes the list of terms in x, and l_x is its coordinates in terms of latitude-longitude. X_t represents the set of tweets in our training data that include a term t in their texts.

3.1 Probability Distributions of Terms

KDE is a widely-adopted non-parametric statistical tool to estimate the probability density function (pdf) of a random variable based on previously observed data [7, 12]. For a location l for which we wish to compute the density of a term t, the density function $\hat{f}$ using KDE is defined as:

$$\hat{f}_t(l) = \frac{1}{|X_t|h} \sum_{x \in X_t} K\left(\frac{l - l_x}{h}\right) \quad (1)$$

where $K(\cdot)$ is the kernel function and h denotes the bandwidth controlling the smoothness of the density distribution. In this work, we adopt the Gaussian kernel, which is one of the most widely used kernel functions, and use the `gaussian_kde` class provided by the SciPy[1] library as our implementation for the $\hat{f}$ function. For each distinct term $t{\in}T$ in our training tweet set, we initialize `gaussian_kde` using X_t and a bandwidth value h. Selection of an optimal bandwidth plays a critical role in KDE, since it directly affects the sharpness/smoothness of the peaks in density distribution. The default method provided by the SciPy library for bandwidth selection is Scott's rule, which assigns $h{=}|X_t|^{-1/(d+4)}$, where d is the number of dimensions (in our case, $d{=}2$). In this work, we first evaluate our location prediction method using Scott's rule in KDE, and we propose the following enhancement for term-specific bandwidth selection to improve the prediction accuracy.

Locality-adapted bandwidth: A common approach for bandwidth selection in KDE is to tune a scalar value on separate validation data and apply this fixed optimized value at density estimation [5, 13]. However, each term can exhibit different spatial characteristics [1, 3, 13]. Therefore, we claim that the accuracy of density estimators would be improved if the bandwidth parameter h is adapted for each term in accordance with their spatial strength. Specifically, the kernel function of a term with strong locality should be given a lower bandwidth so that its density distribution concentrates on the local neighborhood of observation points, while weakly local terms should have a higher bandwidth to have less peaky density distribution over the entire region.

The method we propose to obtain locality-adapted bandwidths uses an information theoretic metric, namely the information gain ratio (IGR), which is an effective feature selection metric to obtain location indicative terms [3, 8, 10]. IGR measures the ratio of information gain (IG) of a term t to its intrinsic entropy. Calculation of IG is given in Eq. 2, where C represents the set of grid cells, $P(t)$ and $P(\bar{t})$ denote the probabilities of presence and absence of the term t, respectively.

$$IG(t) = P(t) \sum_{c \in C} P(c|t) \log P(c|t) + P(\bar{t}) \sum_{c \in C} P(c|\bar{t}) \log P(c|\bar{t}) \quad (2)$$

Calculation of IGR for a term t is given in Eq. 3. The denominator in the equation represents the intrinsic entropy of t over the region.

$$IGR(t) = \frac{IG(t)}{-P(t) \log P(t) - P(\bar{t}) \log P(\bar{t})} \quad (3)$$

We calculate IGR for each distinct term in the training set in order to evaluate their location indicativeness. The reason for selecting IGR is twofold. First, IGR is shown to yield the most accurate results among other feature selection techniques (e.g., IG, χ^2, geospreading [3, 10]), and second, IGR values are in the range of [0,1], where a more location indicative term is expected to have a higher IGR value. In practice, we observe that the most local term in a training set is assigned the IGR value of 1, whereas the least local term has an IGR value around 0.05. We use IGR values to select the kernel bandwidth for each term, such that the bandwidth is determined as being inversely proportional to the locality represented by IGR. Therefore, we introduce the setting in Eq. 4 to adapt the value assigned by Scott's rule (h_{Sc}) specifically for t. In this equation, λ represents a small enough value to avoid zero bandwidth for terms having IGR=1. In our implementation, we use $\lambda{=}min_{t \in T} IGR(t)$, which is around 0.05 as mentioned above. As a result, $h_{IGR}(t)$ is calculated as $\lambda \times h_{Sc}(t)$ for the most local term, and it becomes $h_{Sc}(t)$ for the least local term. We discuss the improvement in accuracy obtained by this setting in Section 4.

$$h_{IGR}(t) = (1 - IGR(t) + \lambda) \times h_{Sc}(t) \quad (4)$$

Integration of densities: After the probability density functions based on Gaussian KDE are initialized for each term, the next step in our training is to assign probability masses to grid cells for each term according to the density functions. Given a pdf $\hat{f}_t$ for a term t, the probability of observing t in a grid cell c is calculated by integrating density values over the area of c, such that $p_t(c){=}\iint_c \hat{f}_t(lat, lon)\, d_{lat}\, d_{lon}$ [7, 12]. In our implementation, we apply the `integrate_box` function provided by the SciPy library for `gaussian_kde`, and calculate probability values for grid cells using their boundary coordinates in terms of latitude-longitude.

We note two specific advantages of using integrated densities as probability masses rather than the density values at selected points. Firstly, every point inside a grid cell can have a different density for a term, and thus, integrating densities over the cell area provides an aggregated value. Secondly, unlike density values, calculated probabilities of a token on each cell range from [0,1], which yields more interpretable results.

3.2 Combination of Probability Distributions

Location prediction for a new tweet is performed by combining the probability distributions of its terms and selecting the grid cell maximizing the cumulative probability. We adopt a weighted sum approach for combination [7, 11, 14]. Specifically, for a tweet x with terms t_x, we apply Eq. 5 to obtain cumulative probabilities for grid cells according to the term probability distributions calculated in

[1] https://pypi.python.org/pypi/scipy/0.19.1

the training stage. In this equation, w_t represents the weight that we assign for term t. Finally, $argmax_{c \in C} p(c|x)$ is selected as the estimated grid cell and its midpoint is assigned as the estimated coordinates for tweet x.

$$p(c|x) = \sum_{t \in t_x} w_t \times p_t(c) \quad (5)$$

One option in the selection of w_t's is to use uniform weighing (e.g., w_t=1 for every $t \in T$). On the other hand, because of different geographical characteristics of terms, higher prediction accuracy can be achieved if weights are determined for each term separately. Applying an optimization algorithm such as Expectation Maximization using a validation dataset could be an alternative for their tuning [7, 14]. However, in our case where we have thousands of distinct terms and thus thousands of weights to tune, this approach would require a considerable amount of tweets for validation. The method that we propose in this work for the selection of weights is to use IGR values that we have already calculated for training. Since local terms are expected to have higher IGR, their effect on the combined results would be directly proportional to these values. In our evaluations, we demonstrate the improvement obtained by using IGR-based weighing over the uniform weights.

4 EVALUATION

We evaluated our proposed method on three different datasets composed of geotagged tweets from London, Paris, and Berlin that are collected using Twitter Streaming API between October-December 2015. We modeled each region as a 100x100 grid, where each grid cell covers an area of approximately 0.5km^2. Following the common practices for data cleaning and spam removal, we excluded exact duplicate tweets, Foursquare check-ins, and tweets from users with more than 1000 friends or followers or who posted more than two tweets per day [2, 3, 6]. This process resulted in 306,731 tweets for London, 153,789 tweets for Paris, and 38,334 tweets for Berlin. Tweets are assigned to the grid cells according to their associated GPS coordinates, and their texts are tokenized using the Twokenize[2] library. Tokens that appear in less than five tweets, hyperlinks, and single characters are excluded in training to reduce data sparsity. This has yielded 39,160 tokens for London dataset. We do not apply any restriction on the language of a tweet. In our experiments, we used randomly selected 95% of tweets in each dataset for training, and the remaining 5% for test.

In the remainder of this section, we use *LocKDE* to refer to our KDE-based location estimation method. Considering the choices for bandwidth selection and probability weight assignments explained in Section 3, we evaluate the performance of *LocKDE* under four different settings. Regarding the bandwidth selection, we use *h=Sc* to denote the setting that uses Scott's rule, and *h=IGR* to refer to our enhancement using IGR (i.e., locality-adapted bandwidth). Uniform weighing of probabilities in Eq. 5 is represented by *w=1*, and our enhanced weighing based on IGR is denoted by *w=IGR*.

We implemented several baseline methods from the literature for comparison. The first one is Class Prior (*CP*), which we use to show that assigning all test tweets to the most populous cell does not yield accurate results [3, 10]. Other two baselines are *MNB* with additive smoothing and *KL*-divergence [3, 5, 13]. Finally, we implemented improvements over *MNB* and *KL* using feature selection according to information gain ratio (MNB_{IGR} and KL_{IGR}), following the descriptions in [3, 13]. Our experiments with other feature selection methods, namely IG, χ^2, and geospreading, did not perform better than IGR for the baselines, which is also consistent with the findings in previous studies. Therefore, we only present the results of baselines with IGR due to limited space. We selected these baselines since they are among the most widely-used techniques and are shown to yield high accuracy. Moreover, similar to *LocKDE*, they also do not require a separate validation dataset for parameter tuning. For a test tweet without any term in the training set, we select the grid cell with the highest class prior.

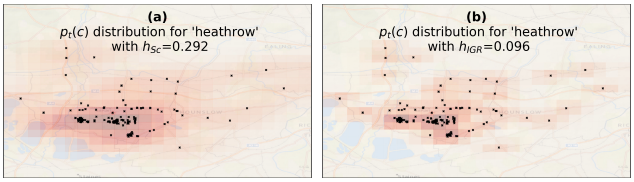

Figure 1: Probability distributions of *heathrow* using different KDE bandwidths. (a) uses h_{Sc} and (b) uses h_{IGR}

The results of our evaluation are given in Table 1. As for our evaluation metrics, *Median* represents the median of the distances between the predicted location and the true location for test tweets. *ExactAcc* is the proportion of correctly estimated grid cells, and *Acc@n* is the proportion of tweets for which the estimated location is at most *n* kilometers away from the true tweet location. We experimented MNB_{IGR} and KL_{IGR} by selecting different numbers of top *n* terms ranked by their IGR, and we demonstrate their best results that yield lowest median error distance in Table 1.

These results show that the most accurate estimations in terms of median error distance are obtained by *LocKDE* with our fourth setting (the rightmost column with *h=IGR*, *w=IGR*). Other three settings of *LocKDE* also perform better than *CP*, *MNB* and *KL* in terms of median error distance. We observe that our enhancements for locality-adapted bandwidth (*h=IGR*) and IGR weights (*w=IGR*) result in an improvement even when they are applied separately. Moreover, when applied together in the fourth setting, the lowest error distances are obtained for every dataset.

The effect of our locality-adapted bandwidth is exemplified in Fig. 1. In this figure, black dots represent locations of tweets mentioning *heathrow*, and the shadings in red represent their probability[3] values $p_t(c)$ for grid cells around the Heathrow airport. Since *heathrow* has a relatively higher locality (IGR=0.711), its pdf uses a smaller bandwidth for *h=IGR*, and thus, its probability distribution in Fig. 1(b) becomes more peaky than the distribution in (a).

The results in Table 1 show that IGR based feature selection improves the accuracy of *MNB* and *KL*. Since the most accurate baseline is MNB_{IGR}, we discuss its comparison with *LocKDE* in more detail. It is notable that *LocKDE* is not as accurate at predicting the exact grid cell as MNB_{IGR}. This is shown by the higher *ExactAcc* values for MNB_{IGR} in Table 1. However, as we increase the error tolerance, we observe that predictions of *LocKDE* are in

[2] https://github.com/brendano/ark-tweet-nlp/

[3] Power-law scaling is applied to probability values for better illustration

Table 1: Comparison of LocKDE and baselines on three datasets. Best results are written in bold.

Dataset	Evaluation Metric	CP	MNB	KL	MNB_{IGR}	KL_{IGR}	LocKDE under different settings			
							h=Sc, w=1	h=Sc, w=IGR	h=IGR, w=1	h=IGR, w=IGR
London	Median (km)	4.048	2.054	3.540	1.155	2.340	1.356	1.091	1.068	**0.957**
	ExactAcc	0.071	0.353	0.321	**0.386**	0.284	0.300	0.312	0.339	0.346
	Acc@0.5km	0.084	0.371	0.336	**0.409**	0.305	0.356	0.372	0.394	0.403
	Acc@1.0km	0.179	0.438	0.391	0.485	0.405	0.467	0.490	0.493	**0.505**
	Acc@5.0km	0.548	0.630	0.547	0.670	0.612	0.691	0.713	0.709	**0.718**
Paris	Median (km)	3.576	1.475	3.153	0.768	2.567	1.075	0.818	0.818	**0.742**
	ExactAcc	0.169	0.420	0.352	**0.466**	0.393	0.360	0.378	0.410	0.420
	Acc@0.5km	0.175	0.435	0.364	**0.481**	0.407	0.407	0.428	0.455	0.465
	Acc@1.0km	0.205	0.467	0.394	0.512	0.431	0.494	0.525	0.522	**0.536**
	Acc@5.0km	0.668	0.694	0.582	0.735	0.589	0.764	0.783	0.779	**0.786**
Berlin	Median (km)	2.811	1.954	3.245	1.595	2.520	1.282	1.101	1.141	**0.975**
	ExactAcc	0.133	0.372	0.305	**0.416**	0.156	0.357	0.364	0.377	0.372
	Acc@0.5km	0.135	0.388	0.320	0.428	0.165	0.406	0.419	0.431	**0.447**
	Acc@1.0km	0.163	0.418	0.349	0.462	0.213	0.469	0.485	0.484	**0.503**
	Acc@5.0km	0.761	0.734	0.599	0.783	0.792	0.820	0.834	0.828	**0.836**

fact not very distant from the true tweet locations. In other words, starting from 0.5-1km, accuracy values in terms of *Acc@n* become higher for *LocKDE*. This is probably due to the smoothing of probabilities provided by KDE, since a term in a grid cell also affects the distribution in the neighboring cells. However, for *MNB*-based methods, since term priors are calculated without any effect on the neighborhood, an inaccurate prediction by *MNB* methods becomes more likely to be in a distant cell than for *LocKDE*. Therefore, our KDE-based method performs better than the baselines and yields lower median error distance. We also analyze the difference in error rates between *LocKDE* (with h=*IGR*, w=*IGR*) and MNB_{IGR} for *Acc*@1.0*km* by employing McNemar's test on their predictions. The test results indicate statistically significant improvement for every dataset in our experiments ($p \ll 0.001$).

Despite its higher accuracy, one drawback of *LocKDE* is its longer training time compared to the baselines, which is mainly due to the computation of integrated densities over the grid. One solution we applied was to parallelize training. Since probability calculations for terms are independent from each other, pdf functions and integrated densities are calculated in multiple parallel processes. Our second solution to speed-up the training was to apply pruning at a higher level of the grid structure. That is, we built a discretization of the grid at a resolution of 10x10, calculated probabilities for terms at this higher level first, and discarded those cells having zero probability without any drill down. This pruning provided nearly 25% decrease in probability computations at 100x100 level. As a result, the training for London (our largest dataset) took approximately 1 hour on a 16-core server. We also note that once the training is complete, location prediction for test tweets takes much shorter time that would not hinder online processing (30-40ms for a tweet).

5 CONCLUSION

In this paper, we proposed a location prediction method for tweets using locality-adapted kernel density estimators. KDE bandwidths and term weights are determined according to the locality of terms represented by their information gain ratios, without requiring any parameter tuning. Our experiments conducted on three datasets from different regions of the world indicate significant improvement in accuracy in comparison to the widely-used tweet localization methods. Using locality-adapted KDE on tweet localization problem has yielded promising results. In our future work, we plan to enhance the model to include the distributions of other tweet features, such as user profile, language, and timezone.

REFERENCES

[1] Hau-wen Chang, Dongwon Lee, Mohammed Eltaher, and Jeongkyu Lee. 2012. @Phillies Tweeting from Philly? Predicting Twitter User Locations with Spatial Word Usage. In *Proc. of ASONAM '12*. 111–118.

[2] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. 2010. A Latent Variable Model for Geographic Lexical Variation. In *Proc. of EMNLP '10*. 1277–1287.

[3] Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based Twitter User Geolocation Prediction. *J. Artif. Int. Res.* 49, 1 (2014), 451–500.

[4] Claudia Hauff and Geert-Jan Houben. 2012. Placing Images on the World Map: A Microblog-based Enrichment Approach. In *Proc. of ACM SIGIR '12*. 691–700.

[5] Mans Hulden, Miikka Silfverberg, and Jerid Francom. 2015. Kernel Density Estimation for Text-based Geolocation. In *Proc. of AAAI'15*. 145–150.

[6] Kyumin Lee, James Caverlee, and Steve Webb. 2010. Uncovering Social Spammers: Social Honeypots + Machine Learning. In *Proc. of ACM SIGIR '10*. 435–442.

[7] Moshe Lichman and Padhraic Smyth. 2014. Modeling Human Location Data with Mixtures of Kernel Densities. In *Proc. KDD '14*. 35–44.

[8] Fernando Melo and Bruno Martins. 2017. Automated Geocoding of Textual Documents: A Survey of Current Approaches. *Transactions in GIS* 21, 1 (2017), 3–38.

[9] Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. 2016. A Simple Scalable Neural Networks based Model for Geolocation Prediction in Twitter. In *Proc. of W-NUT*.

[10] Ozer Ozdikis, Heri Ramampiaro, and Kjetil Nørvåg. 2018. Spatial Statistics of Term Co-occurrences for Location Prediction of Tweets. In *Proc. of ECIR'18*. 494–506.

[11] Reid Priedhorsky, Aron Culotta, and Sara Y. Del Valle. 2014. Inferring the Origin Locations of Tweets with Quantitative Confidence. In *Proc. of CSCW '14*.

[12] B. W. Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

[13] Olivier Van Laere, Jonathan Quinn, Steven Schockaert, and Bart Dhoedt. 2014. Spatially Aware Term Selection for Geotagging. *IEEE Trans. on Knowl. and Data Eng.* 26, 1 (2014), 221–234.

[14] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: Personalized Geo-social Location Recommendation: A Kernel Density Estimation Approach. In *Proc. of SIGSPATIAL'13*. 334–343.

[15] Xin Zheng, Jialong Han, and Aixin Sun. 2017. A Survey of Location Prediction on Twitter. *CoRR* abs/1705.03172 (2017).