Learning Boundaries of Vague Places from Noisy Annotations *

Suradej Intagorn
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292 USA
intagorn@usc.edu

Kristina Lerman
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292 USA
lerman@isi.edu

ABSTRACT

What ordinary people mean by places may differ dramatically from what experts consider them to be. This is especially evident in how people talk about places in social media, where 'Los Angeles', for instance, could include areas well outside of the city or even in another county. In order to make best use of the information in social media, we need to understand what people mean when they refer to a place. Social annotations provide valuable evidence for harvesting knowledge about places, e.g., learning their boundaries and relations to other places. However, social annotations are noisy, and this can dramatically distort the learned boundaries. In this paper we propose a method that exploits the distinctive property of social annotations — that it is created by many people — to filter out noise. Using a large data set extracted from Flickr we show that our crowd-based noise filtering method can learn accurate boundaries of places, including vague places.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Geospatial

Keywords

vague places, social tagging, social annotation, social media

1. INTRODUCTION

What do people mean when they talk about a place? This question has become more salient as people have turned en masse to social media to share information with others. The content they create on Twitter, Flickr, and YouTube, often contains place references. However, these place names are often *vague*, with no formal boundaries [6, 5, 7], or boundaries that differ from the experts' definition. When friends post their vacation photos from 'Los Angeles', what they consider to be 'Los Angeles' is very different from what the experts define as the City of Los Angeles, the County of Los Angeles, or even the Los Angeles–Long Beach–Santa Ana Metropolitan Statistical Area. However, when someone posts a message about a "wildfire in Los Angeles," it is vital to understand what "Los Angeles" means.

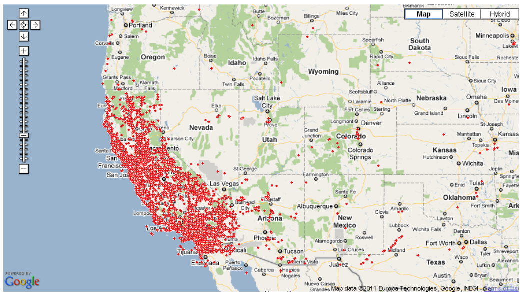
In order to enrich existing gazetteers and improve geospatial information systems, researchers have attempted learn about vague places from the data people create. Most of the work in this area relies on user-generated data in both paper and electronic formats. Early work asked human subjects to draw a boundary of a vague place [1]; then, judgements of many people were aggregated to estimate its boundary [6]. In recent years, new methods were developed to extract geospatial concepts from Web pages [5] and user-generated data [8, 2, 10]. The latter methods exploit users' ability to annotate the content they create on social media sites by attaching descriptive labels, known as *tags*, and geographic coordinates, known as *geo-tags*, to it.

Mining geospatial knowledge from user-generated data and annotations, however, presents many challenges. User-generated data on the Social Web is sparse, ambiguous, and very noisy. The standard solution to these challenges is to aggregate data from many people and use statistical methods, such as clustering, to reduce uncertainty. Such methods were used to identify spatially significant terms from their spatial distribution [8, 2, 9]. However, they are not appropriate for learning boundaries of places, because they do not properly eliminate noise, which can arise because of geocoding and data annotation errors. Even a small amount of noise can significantly distort the shape of a place. Though computational methods exist for handling geospatial noise, they are not appropriate for user-generated data, which is both sparse and very inhomogeneous. In this paper we propose a noise-filtering method that leverages evidence from many people to learn accurate place boundaries.

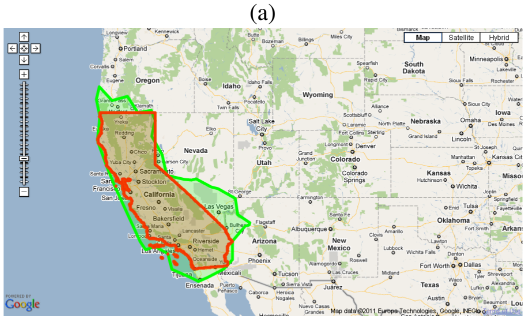
2. FILTERING NOISE IN GEOSPATIAL ANNOTATIONS

Acquiring accurate geospatial knowledge from evidence contributed by many different people presents a number of challenges. Individuals vary in their level of expertise, experience, expressiveness, and their enthusiasm for creating and annotating content. As a re-

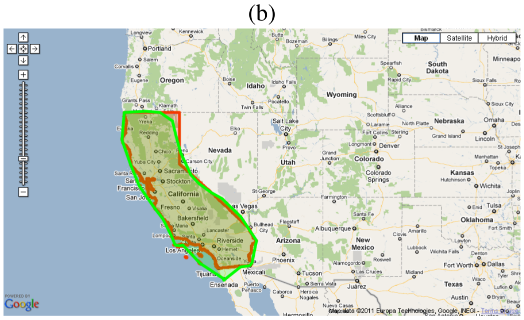
ACM SIGSPATIAL GIS '11, November 1-4, 2011. Chicago, IL, USA.
Copyright 2011 ACM ISBN 978-1-4503-1031-4/11/11...$10.00



(a)



(b)



(c)

Figure 1: Learning the boundary of 'california' on Flickr. (a) Data points associated with the tag 'california'. Results of clustering the points with (b) Dbscan (green polygon) and (c) applying crowd-based noise filtering (green polygon). The ground truth is shown in red.

sult, social annotations are sparse, ambiguous and very noisy. Figure 1 shows spatial distribution of geo-referenced photos on Flickr tagged with the term 'california'. The boundary is quite fuzzy. Some of the noise is the result of geo-coding errors, or mistakes people make when tagging images. Other sources of noise are more subtle, resulting from convention or convenience. For example, users may tag a batch of vacation photos 'California', forgetting about a side trip to Las Vegas, Nevada.

Clustering can be used to disambiguate a place name. For example, in our 'california' example, points in Las Vegas may be assigned to a different cluster than the points in California because they are a bit farther from other 'california' points. However, as we show in this paper, clustering is not a robust method to eliminate noise in user-generated data. Another difficulty is lack of well-defined methodology to find optimal clustering parameters. Instead, we propose a new algorithm that only associates points with a place if enough people support this assignment. Below we explain details of an existing algorithm for clustering geospatial data with noise and then describe our approach to filter noise in social annotations.

Clustering with Dbscan. Dbscan (Density Based Spatial Clustering of Applications with Noise) is a clustering algorithm for noisy spatial data [4]. It has two parameters: radius (Eps) and the minimum number of points ($MinPts$), which is called *support*. Dbscan distinguishes between three types of data points. Core points are those that have at least $MinPts$ neighbors within radius Eps. Border points are data points that are density reachable from the core points but have fewer than $MinPts$ neighbors within a radius Eps. A point p is density reachable from q, when p is a neighbor of q with respect to parameter Eps, and the number of neighbors of q is at least $MinPts$. Noise points are not density reachable from any core points and have fewer than $MinPts$ neighbors within a radius Eps.

Dbscan can be used to cluster user-generated data. However, since this data is sparse, ambiguous and highly inhomogeneous, Dbscan with a fixed parameter may not differentiate between noisy and valid data points. For example, while the tag 'california' has a high concentration in California, there are also many points in Las Vegas, Nevada and Phoenix, Arizona, since many people travel from California to the neighboring states, and may tag photos taken in those places as 'california'. As a result, points associated with 'california' may have enough support in Nevada and Arizona to be included in the 'california' cluster by Dbscan. However, points in the rarely visited regions of Northern California may not have enough support to be included in 'california'. Figure 1(b) shows the boundary of 'california' found by Dbscan, with the ground truth in red. Dbscan creates a boundary far larger than the official boundary for this place and all other examples we studied.

Crowd-based Noise Filtering. The basic idea of our noise filtering approach is to assume that the tag generation process has some error rate. We measure support not by the number of points (photos), but by the number of *distinct people* who have tagged photos with a place name. This reduces a potential source of bias in user-generated data where a single dedicated person annotates a large number of photos. Also, rather than look for a fixed level of support, we classify points in a region as belonging to a place if they have support from a significant number of distinct people who have tagged photos in that region. If there is no support, then at some confidence level we decide that these points do not belong to the place. This modification allows our method to identify points belonging to a place even in the presence of stochastic noise.

Our computational procedure for noise filtering first separates the world into different regions $\mathbf{R}$ specified by the grid size parameter α. In experiments, we set $\alpha = 10$ km. We perform a partial computation for each region $r \in \mathbf{R}$ to capture a statistic which counts the number of distinct people $U_{r,x}$ who used each of distinct tags $x \in X_r$ in region r. In addition, we find the mean and the standard deviation of the statistic. This can be done offline, as a preprocessing step for all tags in the data set, with time complexity $O(n)$. We then check whether significantly many distinct people used a specific tag in the region. We simply check that the number of distinct users is greater than the mean plus some multiple k of the standard deviation, which is the only parameter of our method. In other words, if $U_{r,x} >= (\mu + k\sigma)$, we classify the points in region r as valid points associated with place x; otherwise, they are noise.

Noise filtering may result in clumps of well-separated points, because the place name is ambiguous, e.g., 'victoria' or 'turkey', or because the place itself could be composed of non-contiguous regions, e.g., 'usa'. Therefore, we identify disjoint clusters associated with a place name and find the boundary of each cluster. To identify disjoint clusters we use the connected component method. We create a graph of data points, with an edge between two points if the Euclidean distance between them is less than some radius specified by parameter Eps. After creating the graph, we find its connected components, each of which represents a cluster.

Finally, we estimate the boundary of the shape that subsumes all points in a cluster using Chi-Shape algorithm [3]. The algorithm has only one parameter, chi-parameter, which controls detail of the shape. The input of Chi-Shape is a set of points and chi-parameter.

3. EVALUATION

We used Flickr API to retrieve information about more than 14 million geotagged photos created by 260K distinct users. These photos were tagged over 89 million times, with 3 million unique tags. To evaluate our method, we chose place names for which ground truth is available. We picked 165 tags that correspond to the names of 50 US states and 115 countries for which geotagged photos were available. As the ground truth we use the shape files of US states[1] and countries[2] extracted from online sources. A shape file contains a set of ordered points that define a polygon that represents the boundary of the place.

For each place name in the data set, we extracted all photos tagged with that name. These became the data points. We used the proposed method to eliminate noise, clustered the remaining points and found the boundary of each cluster using chi-shape. We fixed the radius of the connected component in the clustering procedure to be $Eps = 500$ km for the countries data set and $Eps = 100$ km for the states data set. As baseline, we used Dbscan to cluster all points associated with the place name (with radius of support $Eps = 500$ km) and found the boundary of each cluster using chi-shape.

We measured the quality of results both at the level of points and polygons using Precision and Recall. To evaluate points, we measure how many points lie within the ground truth polygon. To evaluate the learned boundary, we measure how much of the area it defines overlaps with the ground truth polygon.

Table 1 compares the performance of the noise filtering algorithm to the baseline at different parameter values: noise threshold (k) for the crowd-based algorithm and support level ($MinPts$) for Dbscan. The first line of the crowd-based algorithm reports results without any noise filtering. As expected, precision increases for higher values of these parameters, while recall drops. This is because at higher parameter values fewer points end up in a cluster, leading to smaller shapes that overlap almost completely with the ground truth, but do not cover all of it.

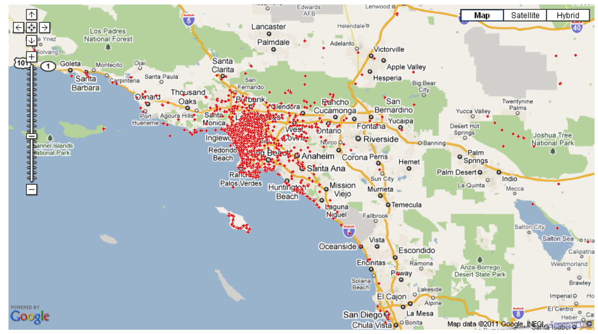
In the US states data set, the crowd-based algorithm results in much higher precision, leading to higher F-scores than baseline. Baseline has higher recall, because it learns large boundaries. The countries data set is much sparser than the states data set, with fewer points contributing evidence to each place concept. As a result, both precision and recall values are lower than for the states data set. Still, the crowd-based noise filtering algorithm learns boundaries that are much more precise than those learned by the baseline algorithm.

[1] http://www.diva-gis.org/gdata

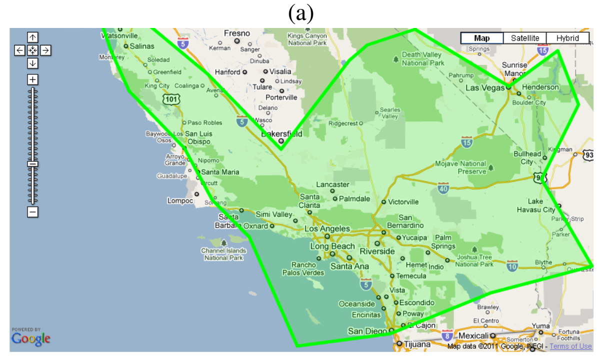
[2] http://thematicmapping.org/downloads/world_borders.php

4. VAGUE PLACES

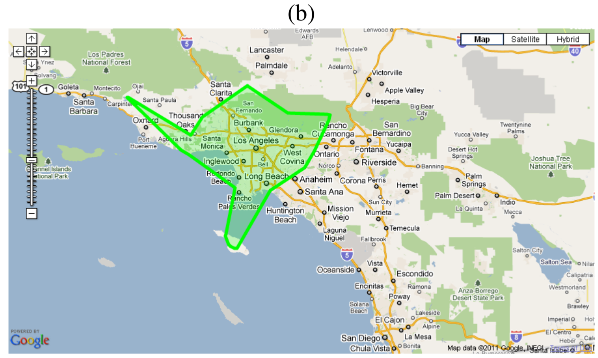
The noise filtering method is general and can be used to find boundaries of vague places for which ground truth does not exist, or which may be different from how people conceptualize the place. For example, during the 2009 Station wildfire in a small city near Los Angeles, many people uploaded photos of the fire that were tagged 'losangeles'. These photos offered information about the fire and the destruction is caused, making it vital to understand where they were taken.



(a)



(b)



(c)

Figure 2: Representation of a vague place associated with the tag 'losangeles'. (a) Points associated with the tag. (b) Boundary learned by (b) Dbscan and (c) after noise filtering. Parameters used in noise filtering were $k = 2$, and clustering $Eps = 500\ km$.

Figure 2(a) maps locations of photos Flickr users tagged with 'losangeles'. These points are noisy and this significantly distorts the boundary learned by Dbscan, shown in Fig. 2(b). The boundary was constructed using connected component clustering with radius $Eps = 100$ km. This boundary is much larger than what most people would consider to be Los Angeles: it extends all the way south to San Diego and all the way north to San Francisco. Fig-

US states						
Crowd-based noise filtering						
$Param$	*points*			*polygon*		
k	Pr	Re	F	Pr	Re	F
no filt	0.933	1.000	0.965	0.516	0.955	0.670
0	0.951	0.933	0.942	0.854	0.867	0.861
1	0.957	0.933	0.945	0.892	0.856	0.874
2	0.960	0.928	0.943	0.910	0.847	0.877
Dbscan						
$Param$	*points*			*polygon*		
$MinPts$	Pr	Re	F	Pr	Re	F
0	0.933	1.000	0.965	0.516	0.955	0.670
40	0.940	1.000	0.969	0.592	0.951	0.729
80	0.944	0.999	0.971	0.619	0.949	0.749
120	0.946	0.999	0.972	0.637	0.947	0.761
160	0.948	0.999	0.973	0.645	0.945	0.767

Countries						
Crowd-based noise filtering						
$Param$	*points*			*polygon*		
k	Pr	Re	F	Pr	Re	F
no filt	0.689	1.000	0.816	0.295	0.684	0.412
0	0.737	0.816	0.774	0.644	0.428	0.514
1	0.743	0.812	0.776	0.699	0.402	0.510
2	0.740	0.788	0.764	0.707	0.374	0.489
Dbscan						
$Param$	*points*			*polygon*		
$MinPts$	Pr	Re	F	Pr	Re	F
0	0.689	1.000	0.816	0.295	0.684	0.412
40	0.709	0.979	0.823	0.507	0.631	0.562
80	0.717	0.927	0.809	0.522	0.608	0.562
120	0.731	0.894	0.804	0.526	0.592	0.557
160	0.723	0.852	0.782	0.519	0.578	0.547

Table 1: Performance results for the crowd-based noise filtering method with $Eps = 100km$, $\alpha = 10km$, and different noise threshold levels and baseline (Dbscan with $Eps = 500km$ with different support levels) on the two data sets.

ure 2(c) shows the boundary learned after noise filtering. We used low threshold $k = 1$ but radius $Eps = 500$ km in the clustering procedure. This is because data was quite sparse and unless we increased the radius, it would have led to disjoint clusters. Despite larger radius, the learned boundary is smaller than before noise filtering, and better agrees with the authors' conceptualization of "Los Angeles." We obtained similar results for other vague places, including 'socal', 'midwest', 'newengland' and 'southwest'. We note that the shape may not be complete due to lack of data. However, as people continue to tag photos and other content, the amount of data from which we could learn about vague places will only grow with time, enabling us to learn more complete representations of places in the future.

5. CONCLUSION

User-generated content and metadata offers unique source of evidence for mining geospatial knowledge, which can then enrich and supplement existing geospatial directories and gazeteers and lead to novel geo-aware applications. This data source is well-suited for learning about vague places, which rarely have official definitions, yet are often invoked in everyday speech and on social media. Social annotations, however, are very noisy, and this noise will substantially distort the boundaries learned for these places. We presented a method that robustly identifies noise in social annotations. We evaluated the method quantitatively on places, including countries and US states, for which ground truth data exists, and showed that our method learns substantially better boundaries than alternative method. We also demonstrated that we can learn reasonable boundaries of vague places, for which no ground truth exists. We plan to integrate this boundary finding method within a larger effort that learns geospatial knowledge.

Acknowledgement

This work is supported, in part, by National Science Foundation under awards CMMI-0753124 and IIS-0812677. Suradej Intagorn gratefully acknowledges the support of the Thai Government.

6. REFERENCES

[1] S. Aitken and R. Prosser. Residents' spatial knowledge of neighborhood continuity and form. *Geographical Analysis*, 22(4):301–325, 1990.

[2] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *WWW '09: Proc. of the 18th Int. Conf. World wide web*, pages 761–770, New York, NY, USA, 2009. ACM.

[3] M. Duckham, L. Kulik, M. Worboys, and A. Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224 – 3236, 2008.

[4] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Int. Conf. Knowledge Discovery and Data mining*, volume 1996, pages 226–231. Portland: AAAI Press, 1996.

[5] C. B. Jones, R. S. Purves, P. D. Clough, and H. Joho. Modelling vague places with knowledge from the web. *International Journal of Geographical Information Science*, 22(10):1045–1065, 2008.

[6] D. R. Montello, M. F. Goodchild, J. Gottsegen, and P. Fohl. Where's downtown?: Behavioral methods for determining referents of vague spatial queries. *Spatial Cognition & Computation*, 3(2):185–204, 2003.

[7] R. Pasley, P. Clough, and M. Sanderson. Geo-tagging for imprecise regions of different sizes. In *Proc. 4th ACM workshop on Geographical information retrieval*, pages 77–82. ACM, 2007.

[8] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *Proc. 30th Int. ACM SIGIR Conf. Research and development in information retrieval*, LNAI, pages 103–110, New York, NY, USA, 2007. ACM.

[9] T. Rattenbury and M. Naaman. Methods for extracting place semantics from flickr tags. *ACM Trans. Web*, 3(1):1–30, 2009.

[10] F. Wilske. Approximation of neighborhood boundaries using collaborative tagging systems. *GI-Days*, 32:179–187, 2008.