IP Geolocation through Geographic Clicks

OVIDIU DAN, Lehigh University, USA
VAIBHAV PARIKH, Microsoft Bing, USA
BRIAN D. DAVISON, Lehigh University, USA

2

IP geolocation databases map IP addresses to their physical locations. They are used to determine the location of online users when their precise location is unavailable. These databases are vital for a number of online services, including search engine personalization, content delivery, local ads, and fraud detection. However, IP geolocation databases are often inaccurate. In this work we present two novel approaches to improving IP geolocation by mining search engine click logs. First, we show that we can derive which URLs have local affinity by clustering clicks from IPs with known locations. We demonstrate that we can further propagate these URL locations to IP addresses with unknown locations. Our approach significantly outperforms two state-of-the-art commercial IP geolocation databases by 25 and 36 percentage points at a distance error of 10 kilometers, respectively. Second, we present an alternative method of assigning locations to URLs when IP location training data is not available, by instead extracting locations from the body of web documents. This second approach also outperforms the baselines by 7 and 17 percentage points, respectively, and has higher coverage than the first method. Finally, we also demonstrate that our two approaches outperform the academic state of the art based on mining query logs.

CCS Concepts: • **Information systems → Location based services**; • **Networks → Location based services**; • **Social and professional topics → Geographic characteristics;**

Additional Key Words and Phrases: IP geolocation, geographic targeting, geotargeting, geographic personalization, click logs, query logs

ACM Reference format:
Ovidiu Dan, Vaibhav Parikh, and Brian D. Davison. 2022. IP Geolocation through Geographic Clicks. *ACM Trans. Spatial Algorithms Syst.* 8, 1, Article 2 (February 2022), 22 pages.
https://doi.org/10.1145/3476774

1 INTRODUCTION

IP geolocation databases map IP ranges to geographical locations. These databases are extensively used by online services such as search engines to determine the location of users at city-level granularity using only their IP address. This location information is then used for **geographic personalization**. For example, the generic query *weather* does not contain an *explicit* location. In order to serve an answer with the local forecast, the search engine needs to determine the *implicit* location of the user. Global positioning sensors can provide the precise location of the users if they

Ovidiu Dan is also with Microsoft Bing.
Authors' addresses: O. Dan, Lehigh University, 17024 NE 131st Pl, Redmond, WA 98052, USA; email: ovd209@cse.lehigh.edu; V. Parikh, Microsoft Bing, Redmond, 23001 36th Ave SE, Bothell, WA 98021, USA; email: vparikh@microsoft.com; Brian D. Davison, Lehigh University, 113 Research Drive (Building C), Bethlehem, PA 18015, USA; email: davison@cse.lehigh.edu.

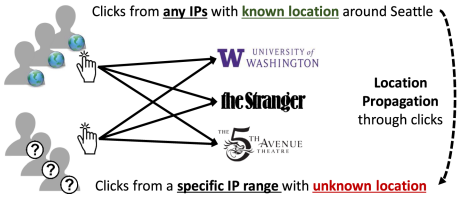
2374-0353/2022/02-ART2 $15.00
https://doi.org/10.1145/3476774

have opted in to location sharing and if their devices contain the necessary hardware. However, in the vast majority of cases, the exact location is not available if users are using a PC without GPS hardware or if they opt out of location sharing. In this case, the search engine falls back to using the IP address of the device to determine a coarse location using a commercial IP geolocation database.

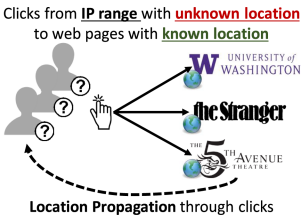
Commercial geolocation services such as MaxMind [39], Neustar IP Intelligence [41], and IP2Location [26] are considered state of the art, although the exact methods they use are proprietary. Related work has questioned their accuracy. Using a ground-truth set of 16,586 router IPs, Gharaibeh et al. found a city-level disagreement of 29% across four different vendors using pairwise distances [17]. Shavitt and Zilberman compute the distance between locations reported by commercial databases on identical IP ranges and report that some pairs of databases have disagreements in the hundreds of kilometers [48]. Poese et al. find errors exceeding 10 kilometers in 80% of the cases, across two commercial databases [45]. Laki et al. have found that MaxMind places multiple spread-out European GÉANT routers in a single location (Cambridge, UK), because that is where the network operator is headquartered and the IP WHOIS records point to that address [31]. Kester evaluates commercial databases using 3,206 IPs consisting of CAIDA Ark [6] and RIPE [51] servers. Note that a subset of Ark nodes are hosted on end-user networks. For the same threshold of 100 kilometers, he finds that MaxMind and IP2Location have an accuracy of 63% and 67%, respectively. In our previous research we evaluated three commercial databases using a ground-truth dataset of 8.4 million end-user IP addresses with known locations [12]. We show that at city level across the top 10 countries, by ground-truth IP density none of the databases achieve an accuracy above 70% at the city level, and in some cases have an accuracy below 10%. Komosnỳ et al. use a mobile application to compile a ground truth of end-user IP addresses with precise GPS locations [30]. They evaluate this dataset against eight commercial geolocation providers, including MaxMind, IP2Location, and Neustar. The dataset of 700 IP addresses covers 16 countries, 52 regions, and 270 cities across 319 ISPs. The addresses include both wireless and wired network IPs, since mobile phones connect to both cellular networks and local Wi-FI access points. Their findings show that for region-level accuracy the best database had 50% accuracy, while city-level accuracy was even worse at 30%. Xu et al. use a ground-truth set of 1.2 million IP addresses with known locations from a city in China and determine that the city-level accuracy of commercial databases varies wildly. For example, the widely used MaxMind GeoLite2 database has a city accuracy of only 14.8% [55].

Despite their shortcomings, IP geolocation databases are used in many other applications, including content personalization and online advertising to serve local content [23, 29], content delivery networks to direct users to the closest datacenter [24], law enforcement to fight cybercrime [49], geographic content licensing to restrict content streaming by region [35], and e-commerce to display variable pricing based on local taxes and shipping [52].

Given a search engine click log, our goal is to generate an IP geolocation database that maps IP ranges to city-level locations. Our work focuses on using click logs, in conjunction with a location ground-truth set and information mined from web documents, to improve IP geolocation at the city level. We propose first assigning geographic focus to URLs by mining user clicks using two alternative methods. We then propagate these locations to IP ranges with unknown locations. Figure 1 presents the intuition behind our two proposals. In the first approach, summarized in Figure 1(a), we propagate locations from IPs with known GPS locations to URLs through user clicks. In the second approach, described in Figure 1(b), instead of propagating locations from IPs with known locations, we mine the web documents themselves for location clues. Finally, the second step in both approaches further aggregates locations per IP range by clustering the coordinates of all clicks from users in each particular IP range. The example in the



(a) ***GeoClicks-GPS*** *approach*:
1) Assign locations to URLs by clustering search clicks from IPs with known GPS location to each distinct URL.
2) Assign URL locations to IP ranges by clustering all clicks to URLs coming from each distinct IP range.
Approach described in Sections 5.1, 6.1, and 7.



(b) ***GeoClicks-Index*** *approach*:
1) Instead of relying on IP locations, assign locations to URLs by mining the body and URL of the clicked documents themselves for location clues.
2) Assign URL locations to IP ranges by clustering all clicks to URLs coming from each distinct IP range.
Approach described in Sections 5.2, 6.2, and 7.

Fig. 1. Intuitive summary of our two proposed approaches. The difference between the two approaches is that in the first one we derive URL locations from IPs with known GPS coordinates, while in the second we derive URL locations from the body or URL fragments of the clicked web documents themselves. The second step of further aggregating URL locations per IP range is shared by both approaches.

figures shows that users in a particular IP range often click on URLs that have local affinity to the Seattle area. We posit that the IP range is then also likely to be in the same area.

Using click logs to improve geolocation poses several challenges. Our work addresses these challenges by clustering locations at the URL and IP range levels, which reduces noise and outliers. First, click data is noisy and sometimes contradictory. Users do not always click on URLs related to their immediate vicinity. In our preliminary investigations we found that mining the click locations coming from any particular IP address does not necessarily reveal its location. For example, a user may be researching vacation spots or may be searching for events in nearby cities. However, if we combine and cluster the clicks from **all** IPs in a particular IP range to a set of URLs with assigned locations, we can successfully weed out outliers. Continuing with our example, even if a subset of users in a particular IP range is searching for different vacation locations, clicks on pages with a local focus are still more prevalent in aggregate over the entire IP range, allowing us to ignore location outliers. Second, determining the geographical focus of URLs is difficult. Some links can have city-level affinity, while others are more dispersed geographically. Take, for instance, a regional bank that has branches in three different cities. To solve this challenge, we only select pages that have a clear single local focus. In the case of the web page that lists branches in three different cities, it is likely that the clicks to the page come from three different locations, which will cause our approach to completely drop the page and not assign it a location. Furthermore, some websites such as Yahoo Finance have no particular geographical focus or have only country-level affinity. Here our clustering algorithm similarly determines that we should skip these pages, since the radius of the top cluster would be too large.

More specifically, our contributions are:

(1) We study the geographic focus of URLs and show that depending on click location dispersion, they can have *regional*, *local*, and *hyper-local* focus.
(2) We propose two approaches to find URLs with local affinity, one based on mining search clicks from IPs with known GPS location, and the other based on mining the body of web documents.
(3) We present an approach to propagate these locations from URLs with local affinity to IP ranges with unknown locations.

(4) We evaluate the accuracy of our two approaches against two state-of-the-art commercial geolocation databases and against the academic state-of-the-art approach that uses query logs. Using a large and diverse ground-truth set of 70 million IP addresses with known locations, we show that **our approaches significantly outperform both the commercial and academic baselines on median error and cumulative error distance.**
(5) Finally, we study the agreement of the two approaches. We show that there is a high level of agreement between the two methods. We then demonstrate that they are also complementary in IP coverage and therefore can be used in conjunction.

2 RELATED WORK

We divide IP geolocation research into two broad categories, based on the methods they use: **network delay and topology** approaches use ping, traceroute, and BGP network structure information; **Internet data mining** approaches use diverse information mined from the Internet, including web page content, WHOIS databases, reverse DNS, and social graphs.

The majority of IP geolocation research relies on active network delay measurements to locate addresses. Early work on IP geolocation by Padmanabhan and Subramanian discusses *GeoPing* [43], which sends ICMP packets from geographically distributed landmark servers to the target IP. It then assigns the target IP the location of the closest landmark server in terms of latency. *CBG* [19] goes further by creating circles on the surface of the earth around each landmark server, where it calculates the radius of each circle based on its measured network delay. It then uses multilateration to infer the location of the target IP at the intersection of these circles. *GeoCluster*, also proposed by Padmanabhan and Subramanian [43], combines BGP routing information with sparse IPs of known locations to assign geographical locations to whole address prefixes. *TBG* [28] uses traceroute from landmark servers to the IP target and performs global optimization to find the location of both landmarks and targets. Youn et al. [57] develop a statistical method for IP geolocation based on applying kernel density estimation to delay measurements. More recently, Ciavarrini et al. [11] presented a framework to understand how the position of landmarks and their distribution affect localization performance. Multiple systems such as Octant [54], Alidade [8], and HLOC [47] combine delay measurement methods with other data sources such as reverse DNS and WHOIS information.

Network delay and topology methods have significant limitations. First, all such methods require access to nodes spread throughout the globe to perform measurements. Second, geolocating a large number of IP addresses using network measurements can run into scalability issues, as each target IP address or range requires separate measurements. The ZMap project from the University of Michigan can scan the entire IPv4 address space using a gigabit connection [14]. However, performing useful network delay measurements would require a significant number of such machines distributed around the world, and attempting to perform traceroutes would require running this probe step once for every hop distance. Third, not all networks allow ICMP pings or fully disclose their network topology. Fourth, routes on the Internet do not necessarily map to geographic distances. Fifth, the ground-truth data for work in this area is usually limited to a few tens of IP addresses, typically located in the United States. For example, *GeoPing* and *GeoCluster* are evaluated on only 256 target IP addresses, all located at universities in the United States; *CBG* is evaluated on only 95 IP addresses in the United States and 42 addresses in Western Europe; *TBG* only targets IP addresses located at U.S. universities; and so forth. Sixth, previously reported mean and median errors of tens to hundreds of kilometers show that these methods cannot be used for practical applications at the city granularity. For instance, *GeoPing* has an error distance of 150 kilometers at the 25th percentile, and *CBG* has a median error of 100 kilometers for some datasets. Seventh, Ciavarrini et al. have demonstrated that network delay approaches

have a best-case error of 20 kilometers and that obtaining an error below this threshold requires a number of active measurement servers so large as to be unpractical [11].

Our work addresses several of these limitations. First, our model does not require issuing active network delay measurements, and therefore does not have the same scalability problems as prior work. Since we use information mined from search engine click logs, our approach can scale to millions of IP addresses. Second, our model has higher accuracy than previous work. Third, we evaluate our method on a ground truth of 70 million IP addresses, which is the largest test set reported in the geolocation literature.

Web mining approaches use diverse information mined from the web. Structon [20] is an approach proposed by Guo et al. that mines the contents of Chinese websites for mentions of locations, using regular expressions. The authors assign these locations to the IP addresses of the web servers hosting this content. They then use IP location interpolation to increase both accuracy and coverage by estimating the location of entire IP ranges from the location of a few individual constituent IP addresses. They assume all IP addresses in the same /24 segment are in the same city and they combine multiple types of IP location interpolation. First, if a majority of IPs in a range are in the same city, they assign that city to the entire range. Second, they continue iteratively applying this heuristic on increasing IP range sizes until they reach a netmask of size /18 (16,384 IPs). If smaller IP ranges inside a larger IP range agree on location, they assign the location to the larger IP range as well. Third, they use a BGP routing table snapshot combined with Autonomous system network information to assign locations to all ranges of small ISPs, if the location of one of the ranges is known. Finally, they perform traceroutes to IPs in /24 segments that still do not have a location. They retain only traceroutes where all nodes in the path responded to ICMP packets. For a target IP, they assign its location to be that of the closest router with known location on the traceroute path. They also propagate locations backwards, starting from a range with known location, assigning it to a router preceding it on a traceroute path, then assigning the interpolated location of the router to all its neighboring ranges. All these approaches taken together achieve an accuracy of 87 percent at the city level. Instead of computing error distance as in other previous work, they map coordinates to cities, and they check if the city of their location candidate matches exactly to that of the ground-truth data point. While these results are impressive, this work has several problems. The starting assumption that the web server hosting a website is in the same location as the organization that owns the website and its users may not hold today. With the advent of cloud computing, many websites are now hosted in centralized data centers and not in decentralized local business offices. Second, the evaluation is performed on a crowdsourced ground-truth set with unknown freshness and accuracy. Third, the manually created extraction rules used to mine location information are tailored specifically for China, and the authors admit they may not work in the rest of the world. Fourth, the paper states the task is made easier by the fact that China only has a few hundred cities, compared to 35,000 cities and towns in the United States. This difference can skew the results favorably when evaluating this approach on Chinese data at city-level granularity. Nevertheless, several approaches described in this work are interesting, especially for IP location interpolation.

One of the two approaches we present here (GeoClicks-Index) is similar to Structon in that it uses locations extracted from websites. However, this is where the similarities stop. Instead of assigning the locations extracted from web pages to the IPs of the servers hosting those pages, we propagate locations through clicks from websites to the IPs of users who searched for and clicked on these websites. In contrast with Structon, we perform a comprehensive evaluation using ground-truth IP addresses covering the entire world.

Backstrom et al. [3] propose an interesting approach that relies on a user's social graph to determine their location. This work is not specifically aimed at improving IP geolocation, and in fact

in later steps they use a commercial IP geolocation as a secondary source of user location. They derive the locations of target users based on the locations of friends. Using self-reported locations as ground truth, they show an improvement over an unnamed IP geolocation database. For an error distance of less than 25 km, the amount of correctly classified IPs increases from 57.2% for the baseline to 67.5% for the proposed method. This approach yields a median error distance of 590 km on a test dataset of 2,830 IPs. The authors state that this method works so long as an individual has a sufficient number of friends whose locations are known, preferably more than 16. To successfully predict locations of users who have not provided a location, they need to be connected to a relatively large number of friends with known locations. There is a long line of other similar research that aims to determine user location, as opposed to IP geolocation, by mining the contents of their social posts [7, 9, 10, 13, 21, 25, 27, 33, 36, 59]. For example, Cheng et al. [10] show that they can geolocate 51% of Twitter users within 161 kilometers of their actual location, using only the textual contents of their posts. However, their ground-truth set contains only 5,119 users and their average error is 2,853 kilometers. In this work, however, we focus specifically on locating IP addresses.

Perhaps the closest in spirit to our *clicks* proposals is our previous work to improve IP geolocation by mining search *queries* [12]. There we extracted locations from explicit queries such as *restaurants in Easton* and showed that we can improve the accuracy of geolocation databases in 49 of 50 countries.

However, that previous work suffers from two problems. First, many queries contain ambiguous locations. There are at least 22 cities called Easton in the United States, which means the query-based approach fails for these types of locations. In contrast, our two new proposals start from precise GPS locations and locations mined from the body of web documents, respectively. In the case of GPS locations, there is no ambiguity, so this problem is completely sidestepped. In the case of locations extracted from web documents, the text body provides much more context to use for disambiguation when compared to queries that are much shorter. Take, for instance, the website of a local restaurant in Easton, PA. In addition to the city name *Easton*, the pages on this website are more likely to contain other disambiguation hints such as *PA*, *Pennsylvania*, *Northampton County*, or the zip code *18045*. These hints are often not available when the input text is a much shorter user query.

A second problem of our previous work is that the query-based method only improves *existing* geolocation databases by correcting some of their records, while our two new proposals start from scratch and are not based on an existing commercial geolocation database. Finally, we demonstrate in the Evaluation section that our new click-based approaches significantly outperform our previous query-based approach.

3 DATASETS AND PRIVACY

Online privacy is becoming increasingly important. Pew Research found in 2016 that while many Americans are willing to share personal information in exchange for accessing online services, they are often cautious about disclosing their information and are frequently unhappy about what happens to that information once companies have collected it [46]. We have designed both our approach and our evaluation with this sensitive subject in mind.

Our ground-truth set contains 70 million IP addresses with known locations, compiled during the 28-day period ending on October 26, 2018. To the best of our knowledge, it is the largest and most diverse set used in the geolocation literature. It was derived from the query logs of a major commercial search engine from devices with global positioning sensors, where users opted in to provide location information. These users agreed to share their location at query time in order

to receive personalized local results. To maintain privacy, an automated pipeline anonymized our ground-truth set by modifying raw locations in a random direction by 584 meters. These anonymized coordinates cannot be used to pinpoint individual addresses but can locate an IP at a neighborhood level. Next, the pipeline aggregated all locations reported for an IP address and reduced location accuracy to city level. IP addresses with a large variance in reported locations were removed as outliers. That is, we discarded any IP address that was present in multiple cities over the course of a month. The result of this filtering step is that our ground-truth set contains mobile IP addresses that are located within a single city, as well as fixed broadband IP addresses (Wi-Fi), since users often connect their mobile devices to their home Internet connections. The resulting dataset contains mappings of IP addresses and their corresponding cities. The location distribution of these addresses roughly follows that of worldwide Internet penetration. While throughout this article we refer to this location data as derived from *GPS* for succinctness, the dataset actually covers all global positioning systems, including *GPS*, *GLONASS*, *Galileo*, *BeiDou*, and so forth. [40].

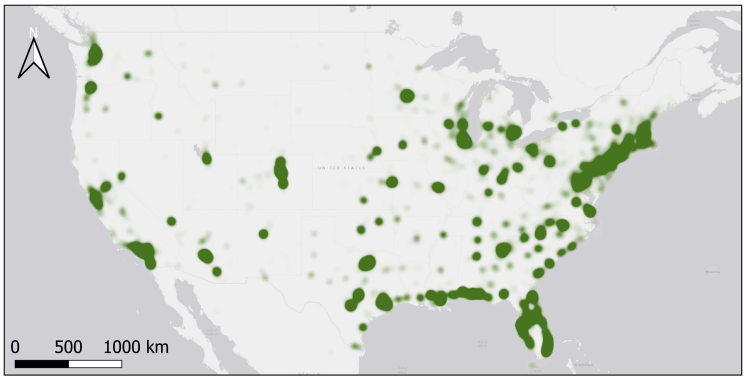
Throughout this article we used this ground-truth set for both training and testing by performing **10-fold cross-validation**. In other words, we split our dataset into 10 equally sized subsets (folds), and then we repeatedly trained on 9 folds and tested on the remaining one. We ran our approach on the data in the 9 training folds. We then evaluated the result on the testing fold by comparing the distance between the location predicted by our approach and the actual location of each IP address.

The **GPS clicks dataset** contains 1.1 billion clicks issued from IPs with known locations. To obtain it, we first extracted a sample of clicks on any search result page element on the same 28-day period ending on October 26, 2018. Then, we intersected this data with the ground-truth set and only retained the clicks that were issued from IP addresses with known locations. The search engine was also aware of the location of users at the time each query was issued initially; therefore, this subset of the data is not skewed by IP locations from commercial databases. To also reduce user click frequency bias, we only retained one click per IP per URL in the entire period. For example, the IP of a user clicking on *https://www.miamiherald.com/* 30 times on five different days would only contribute a single click in the dataset. We normalized all URLs by removing the scheme (*http://*, *https://*), the www. prefix from hostnames, and the # fragments. For instance, we would normalize the URL *https://www.company.com/About_Us#Board* to *company.com/About_Us*. Since this dataset relies on the IPs in the ground-truth set, we also segmented it by the same 10 folds. We use this dataset in Step 1 of the GPS approach Figure 1(a).

The **web index locations dataset** contains 4.1 billion distinct web pages with city-level locations extracted from the textual contents of the web pages or from their URL fragments. We obtained this dataset by randomly sampling from the web index of a large search engine on October 27, 2018. Each address in the dataset is mapped to a single primary location. Section 5.2 discusses the extraction process in more detail. Locations obtained from the text of web pages pose a low privacy concern since the web pages in the index are public. We use this dataset in Step 1 of the Index approach in Figure 1(b).

The **bulk clicks** dataset contains 14 billion clicks from IPs with unknown locations. These clicks were collected from the opt-in logs of a popular browser over a 3-month period ending on October 25, 2018. To obtain the dataset, we randomly sampled from the impressions that contained an HTTP referrer header, which means they were most likely clicks. We use this dataset in both Step 2 of the GPS approach in Figure 1(a) and Step 2 of the Index approach in Figure 1(b).

All click logs were anonymized. We did not have access to the identity of users. During our experiments we aggregated clicks at distinct URL levels, and then further at IP range levels. We never used clicks at an individual user level.



(a) Locations of users clicking on *wunderground.com*

(b) Clicks to *wunderground.com/weather/us/ny/new-york*

Fig. 2. Comparison of a URL that has clicks that are geographically dispersed with a URL that has clear local affinity.

4 GEOGRAPHIC FOCUS

At the onset of our study we set out to determine the viability of assigning locations to URLs using clicks. We also wanted to investigate if it would be enough to assign locations directly to domains as opposed to individual subpages. As a preliminary analysis we aggregated the 1.1 billion clicks from the *GPS clicks* dataset by distinct URLs. We considered the number of click coordinates for each URL as a proxy for their popularity. We then randomly sampled and visualized the coordinates of 100 URLs with varying popularity. Based on our observation, we classify the links into two main categories: URLs that are *geographically dispersed* and URLs that have *local affinity*. We further divide the links with local affinity into *regional*, *local*, and *hyper-local*.

Figure 2(a) displays a heatmap of the click coordinates on *wunderground.com*, which is a weather forecast website. We consider this URL to be *geographically dispersed*, because its click probability roughly follows the population density of the United States. There is no apparent geographical sensitivity to the coordinates. On the other hand, Figure 2(b) plots a similar coordinates heatmap for *wunderground.com/weather/us/ny/new-york*, which is a specific subpage on the same website. We can immediately recognize that the clicks are concentrated toward the New York City metro area. Based on this example, we can draw two conclusions. First, some URLs do indeed show strong local affinity. Second, aggregating clicks only by domain is insufficient. In the case of

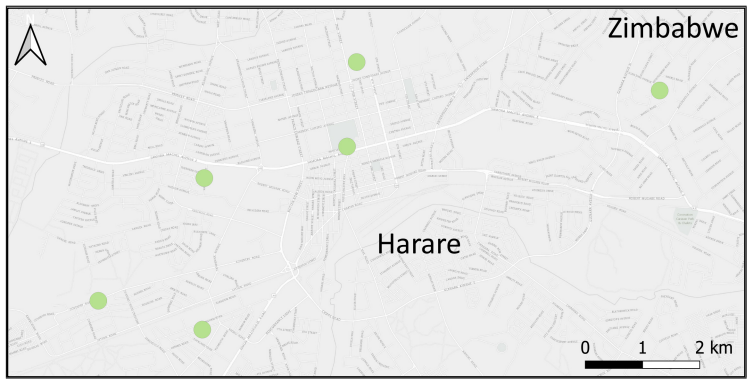


Fig. 3. Location of clicks to *arlington.co.zw*, a real estate website selling houses in Harare, Zimbabwe.

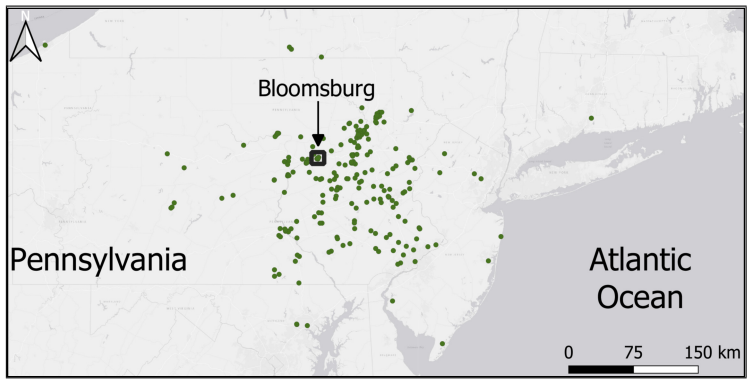


Fig. 4. Location of clicks to Bloomsburg Fair, a yearly event held in Bloomsburg, Pennsylvania.

wunderground.com, the domain is *geographically dispersed*, while city-specific subpages exhibit *local affinity*.

To determine which URLs have a geographic focus, we first implemented a naïve approach that used a reverse-geocoding service to determine the city, state, and country of all coordinates in the sampled URLs. We aggregated the coordinates in each URL by city and sorted by number of occurrences. We then manually visited all the URLs where the top city was present in at least 30% of the clicks. First, we observed that the majority of these links had *local affinity*. Examples include websites for local government and utilities, local businesses such as shopping centers, theaters and concert venues, medical practices, local newspapers and radio, and schools and universities. Some of the links are *local* to a city. Figure 3 shows that clicks on the website *arlington.co.zw* originated within the confines of Harare, the capital of Zimbabwe. This website advertises houses for sale in a local gated community. Others are more *regional*. Figure 4 displays clicks to *bloomsburgfair.com*, which is the website of a yearly fair held in Bloomsburg, Pennsylvania. Since the fair draws attention from multiple neighboring counties, it is not possible to assign it a single geographic location. Another similar example of regional focus is *bosch.in/careers*, which is the careers website for the Bosch company in India. Clicks are concentrated in multiple cities where Bosch has factories or training centers.

URLs can also have *hyper-local* focus. A common example that we observed is student login pages for internal university websites, which are centered on campus locations. But **the most**

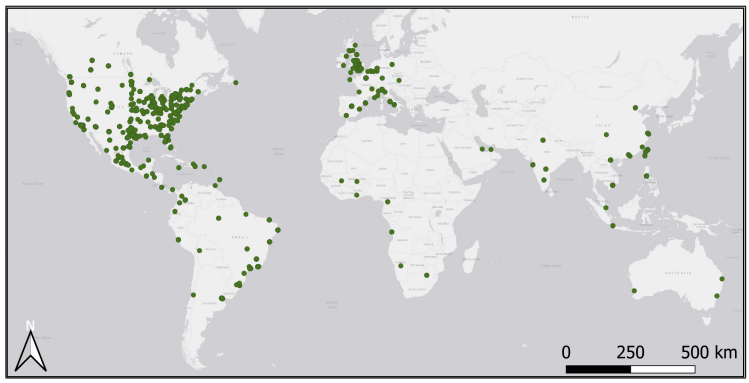


Fig. 5. Locations of clicks to *Yahoo Answers*. Each point represents the mean location of an individual URL.

unexpected finding is that there are links that do not have any obvious geographical focus, yet click information shows that they have in fact local affinity. We found more than 1,500 distinct URLs from the *answers.yahoo.com* domain where most of the clicks were within a small radius of a couple of kilometers. Many of these locations were located on campuses of English-speaking universities and schools. Upon studying the content of the pages, we determined that the questions on these pages were not related to any particular location but were specific math, physics, and literature homework problems. For instance, one popular question that asks, *"Enter the net ionic equation for the reaction of aqueous sodium chloride with aqueous silver nitrate?"* was accessed by 14 different IPs from the campus of a well-known university in Upstate New York. This finding also suggests that some URL locations might have a temporal aspect, which one might explore in future work. For instance, it is likely that the number of these clicks is reduced during the summer holiday. We note that these types of links are still just a fraction of the 426 million distinct URLs in the *GPS clicks* dataset. Figure 5 shows one mean point for each *Yahoo Answers!* URL that received clicks from at least five IP addresses.

5 ASSIGNING LOCATIONS TO URLS

We propose two methods of assigning geographic focus to URLs. The first method requires access to a seed list of IPs with known GPS locations. We aggregate and cluster clicks from these IPs per distinct URL. The advantage of this approach is that, as we will see in the Evaluation section, it is very accurate in assigning locations to URLs. The disadvantages are that it requires having access to the coordinates of a subset of IP ranges, and it has low coverage. The second method instead derives locations from the contents of the clicked documents themselves. The advantages of this approach are that it has much higher coverage, and it only requires access to the web index instead of IP location information. However, these advantages are at the expense of slightly lower accuracy.

5.1 Locations Extracted from IPs with GPS Data

The approach of reverse-geocoding coordinates and aggregating by city names in Section 4 is not sufficient to find URLs with local affinity. We have found that clicks outside the boundary of a city can also contribute directionally to finding the location of links. For example, a blood bank that serves three neighboring cities receives clicks from all three, but only the mean location correctly indicates the neighborhood where the organization is located. Furthermore, using reverse-geocoding services has its own share of problems and might incorrectly place coordinates in the wrong cities.

Instead of reverse-geocoding IP coordinates, we propose using spatial clustering. To better represent geographic focus, we use a modified version of DBSCAN [15], which is a density-based clustering algorithm. Intuitively, it groups together coordinates in high-density areas. One feature of this algorithm is that it does not require specifying the number of clusters a priori. Clusters can reach any size as long as they satisfy the density requirements. Another feature is that it can find arbitrarily shaped clusters, which is not possible with other clustering approaches such as the **expectation-maximization (EM)** algorithm for Gaussian Mixtures. The algorithm has a complexity of $O(n \log n)$ if the implementation uses an indexing structure for finding neighbors.

DBSCAN requires two parameters, ϵ (epsilon) and *minPoints*. The ϵ parameter represents the radius of the search density range around the current point. If the current point has *minPoints* neighbors that are at most ϵ distance away, then the density bar is met and a cluster is formed. The cluster can grow in any direction and to any size as long as the added points are also in a dense area with at least *minPoints* neighbors. Points in low-density areas are considered noise (outliers) and are ignored.

However, using DBSCAN directly yields poor results because of the underlying prior click probability of each geographical area. Cluster sizes are skewed by the presence of *primate cities*, which are cities that dominate the surrounding populated places economically and culturally due to their size [5]. For instance, a person living in a small town at the outskirts of Miami may often search for and click on events in the larger city. To account for this natural bias, we propose re-ranking clusters. For a URL, given a set of coordinates $G = \{ g_1, g_2, \ldots g_n \}$, DBSCAN partitions G into m clusters, $C = \{ c_1, c_2, \ldots, c_m \}$ clusters, each with one or more points. We define the adjusted confidence of a cluster as its size, divided by the prior probability of clicks on its surface:

$$Confidence(c_i) = \frac{|c_i|}{P(click \mid Surface(c_i))}, \tag{1}$$

where we define the surface of a cluster by the polygon that contains all of its points. Note that prior click probability is computed based on the clicks in the entire dataset. In Section 7 we propose a method to estimate this probability.

Figure 6 shows the click coordinates for the *rosalindfranklin.edu* domain, which is a medical school in North Chicago. DBSCAN extracts two clusters, a larger one with more clicks located in Chicago, and a smaller one with fewer clicks located in North Chicago, where the school is actually located. Using just the size of each cluster directly would incorrectly lead us to choose the larger cluster. However, re-ranking the clusters by prior probability gives a higher score to the smaller (correct) cluster.

After ranking the clusters, we pick the top cluster by score and compute its bounding radius, which is the radius of the circle that encompasses all of its points. We then assign the center (mean) location of the cluster to the URL if the bounding radius is within a certain threshold, as discussed in the Evaluation section. This ensures we retain only URLs that have local affinity.

5.2 Locations Extracted from Web Documents

Obtaining a seed list of IPs with known GPS locations can be difficult as location from global positioning sensors may only be available to medium and large online services. Since the size of the resulting dataset is directly proportional to the size of the seed IP list, a large set of IPs is needed to obtain high URL coverage, which may not always be possible. We describe an alternate approach of assigning geographic focus to URLs that only uses the content of clicked documents themselves, instead of using IPs with GPS locations. Our hypothesis is that some web documents contain physical addresses, and these addresses can be later used in aggregate for IP geolocation.

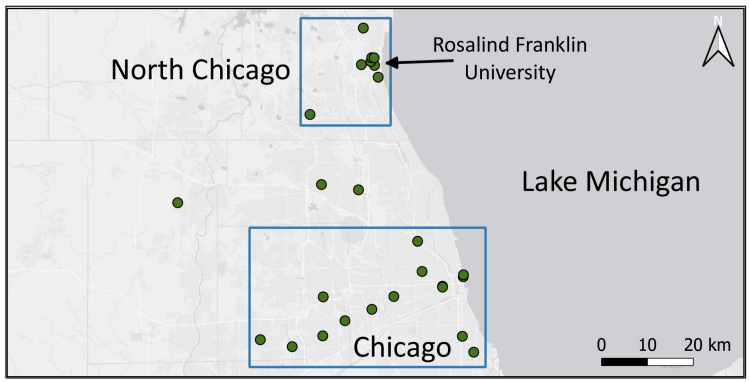


Fig. 6. Re-ranking clusters based on region click density. Coordinates shown are clicks on a specific page in the *rosalindfranklin.edu* domain, which is a medical school in North Chicago. Initially, the bottom cluster in Chicago was ranked first. After adjusting confidence based on prior click density, our approach promoted the cluster in North Chicago to be highest ranked.

There has been ample prior work on extracting addresses from the body of text documents. Amitay at al. [1] parse web documents to extract a taxonomy of locations using a gazetteer. They report an accuracy of up to 82% on multiple document collections that together covered 600 pages and 7,000 geotags. Silva et al. similarly use an ontology of geographical concepts to recognize and disambiguate location references, but they also introduce a graph-ranking algorithm similar to PageRank as a second step to further disambiguate locations. They obtain an F-score of up to 0.81 on document collections in four languages [50]. Martins et al. take a machine learning approach to this problem by using a Hidden Markov Model learner to find location references, then using an SVM classifier to disambiguate references. They outperform two state-of-the-art commercial systems [37]. Locations extracted from web documents are typically used to personalize web search [2, 4, 38]. However, these approaches assume user location is already known and correct. The ranking function then finds documents that are close geographically to the user. If user IP geolocation is incorrect, this assumption may lead to irrelevant search results.

Although the focus of this article is not on parsing locations from text but on using them indirectly through *clicks* for geolocation, we briefly describe the extraction approach used during web index generation. Locations are found either in the body of the document or from URL fragments. The parser attempts to locate full postal addresses, zip codes, or mentions of popular cities in the text of documents. For example, the page *nibbanarestaurant.com* is mapped to the coordinates of *Bellevue, WA*, since it is the web page of a local restaurant and the body of the document contains its geographical address. Sometimes URLs also contain location information. For example, weather websites often contain the forecast location in the link. Another example is *redfin.com/zipcode/30305*, a real estate search web page where the URL contains the zip code of *Atlanta, GA*.

Each document is mapped to at most a single location. If multiple locations are present in the text, only the first one is used. While this data can be noisy at an individual URL level, we posit that aggregating these locations over millions of clicks can lead to reasonable results. The evaluation results will demonstrate that this is a reasonable approach.

We sampled 4.1 billion pages with city-level locations from the index of a large commercial search engine on October 27, 2018, to obtain the web index locations dataset. This alternate approach has a much higher coverage at the URL level of 261 million distinct URLs as compared to the IP GPS location seed list approach, which only yields 3.4 million distinct URLs. However, this

second method may introduce higher noise because the locations listed in text documents may not be representative of the locations of users clicking those documents. We further explore this difference in coverage and accuracy in the Evaluation section.

6 IP RANGE GEOLOCATION

In the previous section we presented two approaches to assign locations to distinct URLs. Here we further propagate these locations to IP ranges with unknown locations using the separate bulk clicks dataset. **Our goal is to determine a single location per IP range** at the city level, which is the same granularity used by commercial geolocation services. To match the typical layout of these services, we segment the IPv4 space into contiguous ranges of 256 IP addresses (*/24* netmask). For example, the *131.107.174.0/24* range starts with address 131.107.174.0 and ends with address 131.107.174.255.

By grouping IP addresses by IP ranges and assigning locations to ranges instead of individual addresses, our assumption is that addresses that are numerically colocated are often also geographically colocated. This assumption is supported by previous research, which has used multiple terms for extrapolating the location of an entire IP range from a few individual addresses, including clustering [43, 44], geographic locality [16], block-based geolocation [18, 32], segment inference [20], IP segmenting [34], and aggregation [8]. Here we call this approach **IP interpolation**. Early work by Padmanabhan and Subramanian proposed a technique called *GeoCluster*, which consists of obtaining IP network prefixes from BGP router table dumps and then propagating IP addresses with known location throughout these prefixes [43, 44]. They also proposed breaking up larger network blocks into smaller segments if the contained ground-truth IPs did not agree on a location. Alidade makes an even stronger assumption that all of the IPs in a prefix must be located in the same location [8]. Structon, proposed by Guo et al., uses interpolation as a technique to increase the IP coverage of a web-mining-based geolocation approach [20]. They assume all IP addresses in the same /24 segment are in the same city. They iteratively apply majority voting to increase IP range sizes until they reach a netmask of size /18. They also combine interpolation with information from BGP routes and traceroutes. Finally, Liu et al. also apply IP location interpolation as part of a location-sharing social-network-based geolocation method called Checkin-Geo [34].

Although we evaluate our approach on IPv4, all methods described in this article can be equally applied to IPv6 IPs. The main difference between the two IP addressing schemes is that the IPv6 ranges are much larger in size; therefore, the IP interpolation needs to happen at a different granularity. At the time when we ran our experiments we did not have access to IPv6 data, but we are considering revisiting this subject in the future.

We begin by describing this step using URL location data derived from IP GPS data discussed in Section 5.1, then detail the same step for the alternate approach using the web index from Section 5.2.

6.1 Using URL Locations from GPS Coordinates

We first intersect the clicks from the bulk clicks dataset with the URLs with assigned locations we found in Section 5.1. The resulting subset contains only clicks to URLs that we previously determined have a certain local affinity. Similar to the previous section, to reduce bias in the data, we count clicks from an IP to a URL a single time in the 3-month period. Then, we aggregate the locations of these clicked URLs per IP range. So for each separate contiguous range of 256 IP addresses we now have a list of coordinates, where each coordinate is derived from the location of the underlying URLs that have local affinity. Finally, we run DBSCAN on the coordinates in each IP range to determine their predominant locations.

We propose a second method to improve the output from DBSCAN at the IP range level. Given an IP range and its top location cluster, the coordinates that make up the cluster are **each** derived from the location of a single URL. For each of these URLs we have previously computed a confidence score in Section 5.1. As the score increases we are more confident that the URL has affinity to that location. Using these scores, we adjust the centroid of the DBSCAN cluster using a weighted average. Since all of the clusters we extract have a small radius of a few kilometers, we can ignore the curvature of the earth. In the next section, we will demonstrate that this proposal results in a noticeable improvement in distance error.

6.2 Using URL Locations from Web Documents

We also perform the same IP range clustering step on locations extracted from the body of web documents. The implementation is very similar to the approach we just took on locations from IPs with GPS coordinates. The main difference is that for the web index data we extracted at most a single location per URL, as previously discussed in Section 5.2. Therefore, the location for each URL has a confidence of 1. In this case, it is unnecessary to use the DBSCAN weighing scheme and we can directly use the standard DBSCAN output. This alternate method has 13 times higher coverage than the IP GPS method, which leads to more IP range clusters and therefore higher IP coverage.

7 MODEL PARAMETERS

Before evaluating our two approaches, we first discuss tuning their model parameters. We begin by discussing the three parameters we use for the model based on IP GPS locations: ϵ (epsilon), *minPoints*, and the maximum cluster bounding radius. We show that by filtering on the bounding radius of the output clusters in the first step, we can obtain a desired balance of accuracy and IP coverage in the second step.

Our geolocation approach consists of two DBSCAN clustering steps. In the first step we cluster locations at the URL level, and in the second step we further cluster URL locations at the IP range level. We run the clustering algorithm separately for each URL and then separately for each IP range. DBSCAN requires two parameters, ϵ and *minPoints*. To find the optimal values for our task, we experimented using a separate validation set of 3 million IP addresses. We set ϵ to 16 kilometers (10 miles) for both clustering steps. This parameter does **not** represent our desired cluster radius, but it represents the neighboring density threshold. DBSCAN can find clusters of any size as long as the density requirements are met. Figure 7 helps demonstrate this property of DBSCAN. The long tail of the figure shows that the output clusters can sometimes cover a large surface as long as the points are dense. We initially set the second parameter *minPoints* to a fixed size, but we soon discovered that we obtained better results if we assigned it dynamically to be 5% of the number of input points. So, for instance, if a URL contained 100 click coordinates, we set *minPoints* to 5.

To compute the confidence score for each URL (Equation (1)), we approximate the prior click density in an area by using Geohash [42, 58], which is a well-known geocoding system for latitude and longitude. We aggregate all coordinates in the GPS clicks dataset by Geohash ID. We set the Geohash precision to five characters, which divides the entire world into 4.9-km-by-4.9-km tiles. In each tile we count how often we observe location clicks across the entire dataset. This allows us to determine a rough prior click probability for any location in the world by consulting the density in its equivalent geohash tile. After re-ranking the clusters by confidence, we pick the cluster with the highest score.

In addition to the ϵ and *minPoints* parameters, we also set a maximum bounding radius for the clusters generated in the first step. The bounding radius of the cluster is determined by the circle that encompasses all points in the cluster. **By filtering on the bounding radius at the end of**

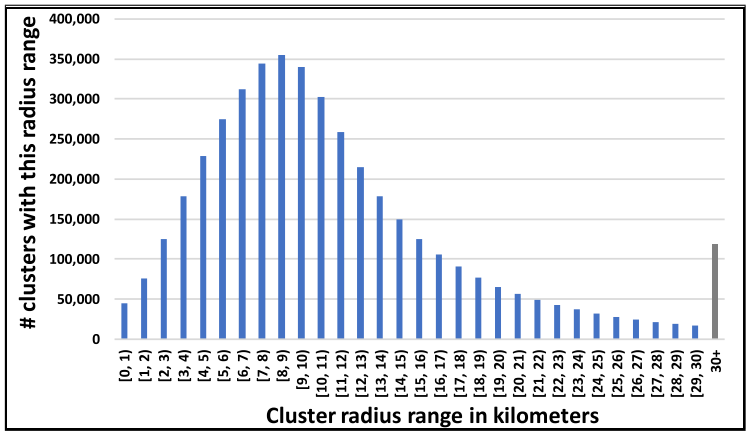


Fig. 7. Distribution of radius for clusters extracted from URL click locations (first clustering step) for $\epsilon = 16$, *minPts* = 5%. The figure shows a normal distribution centered around the [8, 9) data point, which shows that there were about 355,405 clusters with radius between 8 and 9 kilometers. The long tail demonstrates that DBSCAN can generate clusters of dramatically different sizes, as long as the underlying coordinates abide by the density criteria.

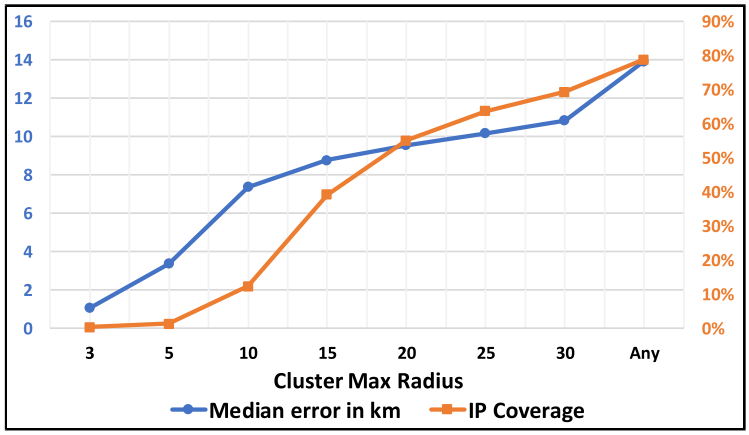


Fig. 8. Effect of varying cluster max radius on median error and on IP coverage. As we increase the maximum radius parameter, both the median error and the IP coverage increase. This setting allows selecting a balance of accuracy and coverage.

the first step, we can tune the amount of accuracy and IP coverage we eventually achieve in the second step. Figure 8 demonstrates the effect that varying this parameter has on both median distance error and IP coverage. We define distance error as the distance between where a model places the coordinates of an IP and the actual location of the IP as given by our ground truth. We define IP coverage as the percentage of IPs from the ground-truth set for which a model makes a decision.

To further show the effect of tuning parameters for accuracy or coverage, we will evaluate two instances of our model based on GPS data: *GPS-HigherAcc*, which is tuned for higher accuracy by setting the maximum bounding radius to 6, and *GPS-HigherCov*, which is tuned for higher coverage by setting the radius to 20. The higher-accuracy variant has a ground-truth IP coverage of 2.2%, while the higher-coverage one has a coverage of 52.2%.

Table 1. Improvement in Accuracy When Using Weighted Centroids for IP Range Locations in GeoClicks-GPS-HigherCov

Cumulative Error in km	Unweighted Centroids	Weighted Centroids	Improvement for Weighted
<10 km	49.9%	52.1%	4.2%
<20 km	74.7%	75.4%	0.9%
<30 km	81.1%	81.3%	0.2%

Table 1 demonstrates the effect of our proposal from Section 6.1 to modify DBSCAN by weighing the centroid locations in the second step by the URL confidence scores computed in the first step. We obtain an improvement of 4.2% in the *error < 10 km* band.

The alternate method to assign geographic focus to URLs makes use of locations extracted from the text of web documents. Since in this approach we do not have to cluster IP coordinates, in the first step we directly assign at most one location to each URL. In the second step we aggregate and cluster URL locations based on clicks issued by users in each IP range. This step allows us to tune the DBSCAN clustering parameters for higher accuracy or coverage. For the index-based approach we also create and evaluate two instances of our model: *Index-HigherAcc* and *Index-HigherCov*. They have higher ground-truth IP coverage than the GPS variants, at 10.9% and 75.4%, respectively.

8 EVALUATION

We compare our approach against three baselines: two state-of-the-art commercial geolocation databases a state-of-the-art academic baseline. We then determine the overall and agreement between the two approaches, and finally we evaluate a combined variant that uses both GPS and index data.

8.1 Commercial Baselines

We compare our approaches against two state-of-the-art commercial databases, *ProviderA* and *ProviderB*. We cannot reveal the names of the proprietary databases since their terms of use forbid comparative benchmarking. They are among the most popular and accurate databases and they are both available to the public.

Figure 9 compares error distance between four *GeoClicks* instances and the two commercial providers. The x-axis represents the cumulative error distance, while the y-axis shows how many points fall within that particular error distance band. For instance, the second column shows that *GeoClicks-GPS-HigherAcc* places 80.5% of the predicted locations within 20 km of their actual location in the ground-truth set. **The figure shows that our approaches significantly outperform the commercial location services** in cumulative error distance.

Table 2 also compares the methods across several metrics. Our four variants achieve better results in median error and percentage of ground-truth IPs with error smaller than 10 kilometers. The last column in the table shows Root-Mean-Squared-Error [22] in kilometers. One difference between RMSE and median is that RMSE easily gets swayed by large outliers, whereas median does not. Our four base variants come close but do not surpass the commercial providers in RMSE. While our models generally yield more accurate locations than the commercial baselines, when they do make a mistake the distance error is sometimes larger than that of the commercial providers, since click data can be noisy. Nevertheless, overall our proposals yield much better results than the commercial services. Furthermore, in Section 8.3 we evaluate a variant that combines the GPS and index methods and achieves a better RMSE result than all of the baselines.

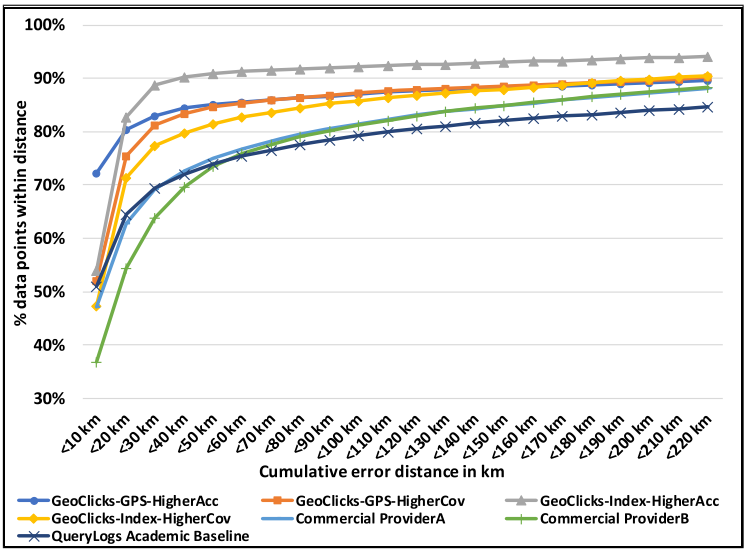


Fig. 9. Error distance in kilometers with 10-fold cross-validation between our four variants, two commercial geolocation services, and an academic baseline that mines query logs [12].

Table 2. Performance of Four Instances of Our Approaches, Two State-of-the-Art Commercial Geolocation Services, a Strong Academic Baseline, and a Combination of Our Methods, on Several Metrics: Median Error (Lower Is Better), Percent of Matching Ground-Truth Points Where Error Is Smaller Than 10 km (Higher Is Better), and RMSE (Lower Is Better)

	Median Error	% Distance <10 km	RMSE in km
GeoClicks-GPS-HigherAcc	**4.5**	**72.2%**	893.4
GeoClicks-GPS-HigherCov	9.5	52.1%	711.1
GeoClicks-Index-HigherAcc	9.2	54.0%	1,327.4
GeoClicks-Index-HigherCov	10.7	47.3%	1,498.6
Commercial Provider A	11.1	47.2%	545.9
Commercial Provider B	16.7	36.7%	545.3
Query Logs Geolocation [12]	9.6	51.0%	2,126.4
GeoClicks-Intersect-HigherCov	8.7	57.2%	**375.5**

The disadvantage of our two approaches is that they have lower ground-truth coverage than commercial databases. In our previous work we have evaluated three commercial databases, which had IP coverage between 94.1% and 97.3% [12]. In comparison, here our instance with highest coverage only achieves a coverage of 75.4%. However, our coverage still far surpasses prior academic work. We discuss one approach to further improve coverage in Section 8.3.

8.2 Academic Baseline

We compare our two approaches against the aforementioned academic state-of-the-art IP geolocation approach based on mining query logs [12]. We re-implemented this academic baseline by mining query logs over a period of 28 days, ending on October 26, 2018. We reduced bias

caused by single addresses by selecting one query instance per IP per day. Using the same methodology as in the original paper [12], we then retained queries with local intent such as business searches, directions, local cinema showtimes, and local weather. Finally, we filtered the remaining impressions to keep only the ones that contained explicit locations. This resulted in 374 million queries that were issued from 3.4 million distinct /24 (256 IPs) buckets. After grouping and filtering locations by IP range, we evaluated the baseline on our ground-truth set.

Figure 9 shows that the query logs approach generally has lower accuracy than the click-based approaches, with the exception of accuracy at <10 km, where the baseline surpasses our web index-based variant that is tuned for higher coverage, but still comes up short when compared to our three other instances.

Table 2 contains a comparison across several metrics. Our four click-based instances significantly outperform the query logs approach in RMSE, and three of four variants outperform the baseline in median error and accuracy at the 10-kilometers threshold.

In conclusion, our click-based approaches outperform a baseline based on mining query logs, but the choice of using one click-based variant over another depends on the application. For applications in need of higher coverage, the index-based approach is the best option. However, if instead the goal is to achieve the highest accuracy, then a GPS-based approach is the best choice.

8.3 Agreement and Overlap

Our two approaches are based on propagating location information extracted from GPS sensors and the body of web pages, respectively. In this section we aim to quantify the degree to which there is overlap and agreement between these techniques. We compare the higher-coverage variants. The variant based on GPS data has a ground-truth IP coverage of 52.2%, while the ones using data from the body of web pages has a higher coverage at 75.4%.

The method based on GPS locations has a total coverage of 1.32 million IP ranges, while the one based on mining web content has a coverage of 1.69 million ranges. Their intersection results in 821,571 ranges. **This result shows that while the techniques output many IP ranges in common, they are also quite complementary**, with 504,109 IP ranges only covered by the GPS method, and 870,971 IP ranges only covered by the web index approach.

We now analyze the IP ranges that the methods share in common. We examined IP ranges shared by both approaches and found out that in 74.5% of cases the locations emitted by the two approaches are within 10 kilometers of each other. If we expand this range to 20 kilometers, then the agreement increases to 80.5%. The results show that even though the approaches derive locations from very different data sources, they have excellent agreement.

Finally, an obvious question one may ask is if the intersection of the two approaches yields better evaluation results than the individual methods. First, we retain the common IP ranges where the two locations are within 20 km of each other. Second, for each IP range we take the mean point between the two locations. Third, we evaluate the resulting dataset against the ground truth. Figure 10 shows that taking the intersection of the approaches results in better accuracy across the entire distance spectrum. Table 2 further shows that the median error is 8.7 km, which makes it second in accuracy only to the higher-accuracy variant of the GPS approach. RMSE is only 375.5, which is the lowest (best) across all variants and baselines. However, the combined approach has a coverage of only 40.5%, which as expected is lower than either approach.

9 CONCLUSIONS

We studied propagating locations from IPs with known locations to IPs with unknown locations using user clicks. **To the best of our knowledge, search using clicks to improve IP geolocation has never been attempted before in the literature.** Using click logs to improve geolocation poses several challenges. First, click data is noisy and sometimes contradictory. Users

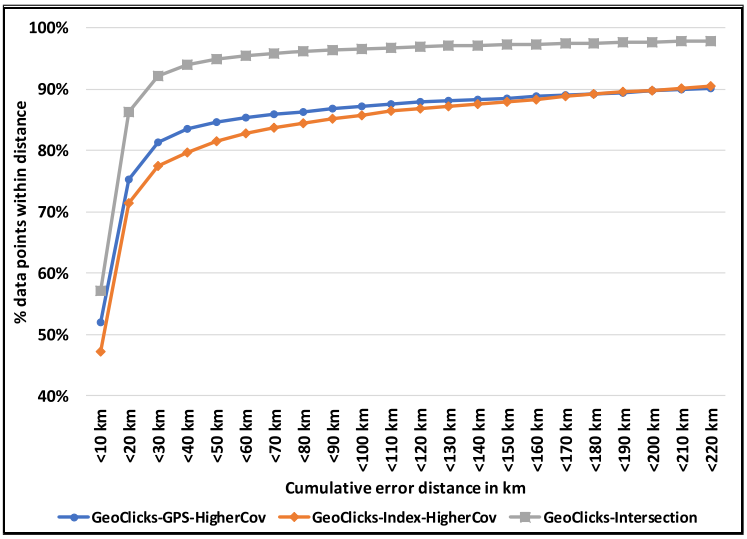


Fig. 10. Evaluation of blended results against ground truth.

do not always click on URLs related to their immediate vicinity. For example, they may be researching vacation spots, or they may be searching for events in nearby cities. Second, determining the geographical focus of URLs is difficult. Some links can have city-level affinity, while others are more dispersed geographically. Take, for instance, a regional bank that has branches in three different cities. Furthermore, some websites such as Yahoo Finance have no particular geographical focus or have only country-level affinity.

Our research has practical applications in improving search engine personalization, as well as other online services. It can also augment academic research in geographic user cohort modeling [53, 56]. Results show that our two proposals significantly outperform two widely used commercial geolocation databases and a strong academic baseline. The results also show that our two approaches are complementary, with roughly half of the IP ranges overlapping, and that their intersection is highly accurate, with a median error of only 8.7 kilometers.

Both the GPS and index-based approaches propagate locations through user clicks. We do not distinguish between the type of page or page element that was clicked. One could further develop these approaches by further breaking down the types of clicks. For example, if a click is issued inside a search result page, did the user click on a simple algorithmic result or inside an answer module such as business listings, weather, or movie showtimes? Also, does the intent of the user query before the click matter? Furthermore, is there a difference between users who click on news articles versus people who click on Wikipedia pages? In summary, it might be worthwhile to investigate if specific categories of clicks better reveal the locations of users. Finally, it would also be interesting to further study the temporal aspect of certain URLs with hyper-local focus, such as the *answers.yahoo.com* example from Section 4.

Lastly, in this article we have exclusively used IPv4 data. We believe that the same approach can be used on IPv6 by modifying the IP range size used in Section 6. Nevertheless, it would be good to perform experiments to validate our assumption.

REFERENCES

[1] Einat Amitay, Nadav Har'El, Ron Sivan, and Aya Soffer. 2004. Web-a-where: Geotagging web content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 273–280.

[2] Leonardo Andrade and Mário J. Silva. 2006. Relevance ranking for geographic IR. In *Workshop on Geographic Information Retrieval (GIR'06), colocated with SIGIR*.
[3] Lars Backstrom, Eric Sun, and Cameron Marlow. 2010. Find me if you can: Improving geographical prediction with social and spatial proximity. In *WWW 2010*. ACM, Raleigh, North Carolina, USA, 61–70. https://doi.org/10.1145/1772690.1772698
[4] Paul N. Bennett, Filip Radlinski, Ryen W. White, and Emine Yilmaz. 2011. Inferring and using location metadata to personalize web search. In *SIGIR 2011*. ACM, Beijing, China, 135–144. https://doi.org/10.1145/2009916.2009938
[5] Brian J. L. Berry. 1961. City size distributions and economic development. *Economic Development and Cultural Change* 9, 4 (1961), 573–588. http://www.jstor.org/stable/1151867.
[6] Ark CAIDA. 2015. Archipelago Measurement Infrastructure. Retrieved 10 August, 2021 from https://www.caida.org/projects/ark/.
[7] Swarup Chandra, Latifur Khan, and Fahad Bin Muhaya. 2011. Estimating Twitter user location using social interactions–A content based approach. In *2011 IEEE 3rd International Conference on Privacy, Security, Risk and Trust (PASSAT'11) and 2011 IEEE 3rd International Conference on Social Computing (SocialCom'11)*. IEEE, 838–843.
[8] Balakrishnan Chandrasekaran, Mingru Bai, Michael Schoenfield, Arthur Berger, Nicole Caruso, George Economou, Stephen Gilliss, Bruce Maggs, Kyle Moses, David Duff, et al. 2015. *Alidade: IP Geolocation without Active Probing*. Technical Report. Department of Computer Science, Duke University.
[9] Hau-wen Chang, Dongwon Lee, Mohammed Eltaher, and Jeongkyu Lee. 2012. @ Phillies tweeting from Philly? Predicting Twitter user locations with spatial word usage. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM'12)*. IEEE Computer Society, 111–118.
[10] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: A content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 759–768.
[11] Gloria Ciavarrini, Maria S. Greco, and Alessio Vecchio. 2018. Geolocation of Internet hosts: Accuracy limits through Cramér–Rao lower bound. *Computer Networks* 135 (2018), 70–80.
[12] Ovidiu Dan, Vaibhav Parikh, and Brian D. Davison. 2016. Improving IP geolocation using query logs. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. ACM, 347–356.
[13] Clodoveu A. Davis Jr, Gisele L. Pappa, Diogo Rennó Rocha de Oliveira, and Filipe de L. Arcanjo. 2011. Inferring the location of twitter messages based on user relationships. *Transactions in GIS* 15, 6 (2011), 735–751.
[14] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide scanning and its security applications. In *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security'13)*. 605–620.
[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. 226–231.
[16] Michael J. Freedman, Mythili Vutukuru, Nick Feamster, and Hari Balakrishnan. 2005. Geographic locality of IP prefixes. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*. USENIX Association, 13–13.
[17] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A look at router geolocation in public and commercial databases. In *Proceedings of the 2017 Internet Measurement Conference*. ACM, 463–469.
[18] Bamba Gueye, Steve Uhlig, and Serge Fdida. 2007. Investigating the imprecision of IP block-based geolocation. In *International Conference on Passive and Active Network Measurement*. Springer, 237–240.
[19] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. 2006. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Transactions on Networking* 14, 6 (Dec. 2006), 1219–1232. https://doi.org/10.1109/TNET.2006.886332
[20] Chuanxiong Guo, Yunxin Liu, Wenchao Shen, H. J. Wang, Qing Yu, and Yongguang Zhang. 2009. Mining the Web and the Internet for Accurate IP Address Geolocations. In *INFOCOM 2009*. 2841–2845. https://doi.org/10.1109/INFCOM.2009.5062243
[21] Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research* 49 (2014), 451–500.
[22] Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data Mining: Concepts and Techniques*. Elsevier.
[23] Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. 2013. Measuring personalization of web search. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 527–538.
[24] Cheng Huang, D. A. Maltz, Jin Li, and Albert Greenberg. 2011. Public DNS system and Global Traffic Management. In *INFOCOM 2011*. 2615–2623. https://doi.org/10.1109/INFCOM.2011.5935088
[25] Mans Hulden, Miikka Silfverberg, and Jerid Francom. 2015. Kernel density estimation for text-based geolocation. In *AAAI*. 145–150.

[26] IP2Location.com. 2018. Geolocate IP Address Location using IP2Location. Retrieved August 13, 2018, from https://www.ip2location.com/.
[27] David Jurgens. 2013. That's what friends are for: Inferring location in online social media platforms based on social relationships. *ICWSM* 13, 13 (2013), 273–282.
[28] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. 2006. Towards IP geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. ACM, 71–84.
[29] Bernhard Kölmel and Spiros Alexakis. 2002. Location based advertising. In *1st International Conference on Mobile Business*. Athens, Greece.
[30] Dan Komosnỳ, Miroslav Vozňák, and Saeed Ur Rehman. 2017. Location accuracy of commercial IP address geolocation databases. *Information Technology and Control* 3, 46 (2017), 334.
[31] Sándor Laki, Péter Mátray, Péter Hága, Tamás Sebők, István Csabai, and Gábor Vattay. 2011. Spotter: A model based active geolocation service. In *2011 Proceedings IEEE INFOCOM*. IEEE, 3173–3181.
[32] Yeonhee Lee, Heasook Park, and Youngseok Lee. 2016. IP Geolocation with a crowd-sourcing broadband performance tool. *ACM SIGCOMM Computer Communication Review* 46, 1 (2016), 12–20.
[33] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, and Kevin Chen-Chuan Chang. 2012. Towards social user profiling: Unified and discriminative influence model for inferring home locations. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1023–1031.
[34] Hao Liu, Yaoxue Zhang, Yuezhi Zhou, Di Zhang, Xiaoming Fu, and K. K. Ramakrishnan. 2014. Mining checkins from location-sharing services for client-independent ip geolocation. In *2014 Proceedings IEEE INFOCOM*. IEEE, 619–627.
[35] Lori MacVittie. 2012. Geolocation and Application Delivery. Retrieved August 2, 2018, from https://www.f5.com/pdf/white-papers/geolocation-wp.pdf.
[36] Jalal Mahmud, Jeffrey Nichols, and Clemens Drews. 2014. Home location identification of Twitter users. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 47.
[37] Bruno Martins, Ivo Anastácio, and Pável Calado. 2010. A machine learning approach for resolving place references in text. In *Geospatial Thinking*. Springer, 221–236.
[38] Bruno Martins and Pável Calado. 2010. Learning to rank for geographic information retrieval. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*. ACM, 21.
[39] MaxMind, Inc. 2018. Detect Online Fraud and Locate Online Visitors. Retrieved August 13, 2018, from https://www.maxmind.com/en/home.
[40] Pratap Misra and Per Enge. 2006. *Global Positioning System: Signals, Measurements and Performance*. 2nd ed. Ganga-Jamuna Press.
[41] Neustar, Inc. 2018. IP Intelligence. Retrieved August 13, 2018, from https://www.security.neustar/digital-performance/ip-intelligence.
[42] Gustavo Niemeyer. 2008. Geohash. Retrieved on 10 August, 2021 from https://forums.geocaching.com/GC/index.php?/topic/186412-geohashorg/.
[43] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. 2001. An investigation of geographic mapping techniques for internet hosts. In *SIGCOMM 2001*. ACM, San Diego, California, USA, 173–185. https://doi.org/10.1145/383059.383073
[44] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. 2001. Determining the geographic location of Internet hosts. In *SIGMETRICS/Performance*. 324–325.
[45] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. 2011. IP geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review* 41, 2 (2011), 53–56.
[46] Lee Rainie and Maeve Duggan. 2016. Privacy and information sharing. *Pew Research Center*. Pew Research Center. Retrievd on 8 August, 2021 from https://www.pewresearch.org/internet/2016/01/14/privacy-and-information-sharing/.
[47] Quirin Scheitle, Oliver Gasser, Patrick Sattler, and Georg Carle. 2017. HLOC: Hints-based geolocation leveraging multiple measurement frameworks. *arXiv preprint arXiv:1706.09331* (2017).
[48] Yuval Shavitt and Noa Zilberman. 2011. A geolocation databases study. *IEEE Journal on Selected Areas in Communications* 29, 10 (2011), 2044–2056.
[49] Craig A. Shue, Nathanael Paul, and Curtis R. Taylor. 2013. From an IP address to a street address: Using wireless signals to locate a target. In *WOOT 2013*. USENIX, Washington, D.C. https://www.usenix.org/conference/woot13/workshop-program/presentation/Shue.
[50] Mário J. Silva, Bruno Martins, Marcirio Chaves, Ana Paula Afonso, and Nuno Cardoso. 2006. Adding geographic scopes to web resources. *Computers, Environment and Urban Systems* 30, 4 (2006), 378–399.
[51] RN Staff. 2015. Ripe atlas: A global internet measurement network. *Internet Protocol Journal* 18, 3 (2015), 2–26.
[52] Dan Jerker B. Svantesson. 2007. E-commerce tax: How the taxman brought geography to the "Borderless" Internet. *Revenue Law Journal* 17, 1 (2007), 11.

[53] Ryen W. White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. 2013. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 1411–1420.

[54] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. 2007. Octant: A comprehensive framework for the geolocalization of Internet Hosts. In *NSDI 2007*. USENIX Association, Berkeley, CA, 23–23. http://dl.acm.org/citation.cfm?id=1973430.1973453.

[55] Wei Xu, Yaodong Tao, and Xin Guan. 2018. Experimental comparison of free IP Geolocation services. In *International Conference on Security with Intelligent Computing and Big-data Services*. Springer, 198–208.

[56] Jinyun Yan, Wei Chu, and Ryen W. White. 2014. Cohort modeling for enhanced personalized search. In *Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 505–514.

[57] Inja Youn, Brian L. Mark, and Dana Richards. 2009. Statistical Geolocation of Internet Hosts. In *ICCCN*. 1–6. https://doi.org/10.1109/ICCCN.2009.5235373

[58] Shubin Zhang, Jizhong Han, Zhiyong Liu, Kai Wang, and Shengzhong Feng. 2009. Spatial queries evaluation with mapreduce. In *8th International Conference on Grid and Cooperative Computing, 2009 (GCC'09)*. IEEE, 287–292.

[59] Xin Zheng, Jialong Han, and Aixin Sun. 2018. A survey of location prediction on Twitter. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1652–1671. https://doi.org/10.1109/TKDE.2018.2807840

Received December 2020; revised May 2021; accepted July 2021