IP Geolocation Estimation using Neural Networks with Stable Landmarks

Hao Jiang
The University of Chicago

Yaoqing Liu
Clarkson University

Jeanna N. Matthews
Clarkson University

***Abstract*—The ability to accurately determine the geographic location of an arbitrary IP address has potential in many applications. Previous methods based on observing the relationship between network delay and physical distance are inaccurate. Methods based on delay similarity are more accurate, but inefficient because they need information on a large number of landmark nodes near the destination to be collected and maintained. We propose a method that can overcome both problems. Our method maintains a stable collection of network observers and landmark nodes that covers the target area. Observers are network nodes from which we can issue measurement command such as ping and traceroute. Landmarks are IP addresses which are reachable from observers and for which the physical locations are well-known. With measurement results collected from these landmarks, we trained two-tier neural networks that estimated the geolocation of arbitrary IP addresses. Our experiments demonstrate that a high accuracy similar to earlier methods retained with a limited number of landmarks. More specifically, the median error of our estimation is 4.1 km on the datasets with 1547 landmarks across US territory. The median error decreased to 3.7 km on half of the test regions which contain more than 100 landmarks.**

I. Introduction

The purpose of IP geolocation is to estimate the geographic location of a given IP address. Accurate IP geolocation techniques have high potentials in many applications. For example, knowing the geographic location of a new client, a distributed system like Netflix and Amazon AWS can dispatch the requests to a nearest datacenter node and thus ensure a better service quality. Online advertisement systems like Google AdWords can offer their viewers more accurate information about local discount activities. Online credit card transactions originating in a region far away from the owner's usual region of activity can be investigated for potential fraud.

We gain two important insights from previous work. First, for IP addresses that are geographically adjacent, their measurement results from the same observer should be similar ([1][2]). Hence we can collect measurement data from landmarks together with their locations, and use the data to train a model that describes the distribution of locations conditioned on measurement results. We then use this model to estimate the location of an IP address from the measurement results. Second, we notice that in a small region (of radius 100-300 km), this estimation problem will be much simpler than it is in a wide area ([3][1]). For example, in a small area the earth surface can be seen as an Euclidean space. The IPs that are physically adjacent will more likely share the same ISP and use similar type of cables. In addition, based on our observation, it takes only a single hop for most subnets to reach another in such a small region. Thus the latency between them can be linear to their distance. Therefore, it may lead to more accurate estimating results to use local data than the entire dataset.

With these insights, we designed and implemented a new IP geolocation approach. Our work consists of two parts: data collection and model training on two-tiered neural networks. We first collect stable landmarks from different well-known data sources, then measure these landmarks from different reliable observers. These measurement results, together with their known locations in the format of latitude and longitude, form our training data. We then built two-tier neural networks as the model. Tier 1 was trained with the entire dataset and responsible for estimating a region the target IP resides in. Once we determined the region, Tier 2 was trained with only the local data to identify the final geolocation.

To estimate the geographic location of arbitrary IP address, we first collect the network latency data from the observers to the target IP, and use tier 1 to locate a region this IP resides in, then use the Tier 2 network belonging to that region to give an accurate estimate of the location of this IP address. This model will allow us to calculate the geographical location of arbitrary measurable IP address.

We collected datasets from three different sources - Ripe Atlas probes, universities websites and city government websites. Ripe Atlas probes reside primarily in residential networks and academic networks. University websites are in academic networks and city government websites use primarily commercial networks. These data sources offer substantial diversity to our datasets, which are far larger than most of the datasets that were used in previous research ([3][4][5]). Therefore, it gives more accurate and reliable results, e.g., we have a median error of 4.1 km with 1547 landmarks in US-based experiments.

The rest of the paper is organized as follows. Section II presents our data collection process and results; Section III introduces our two-tiered networks structure; Section IV describes our experimental results and analysis; Section V discusses related work; and Section VII concludes the paper.

II. Data Collection

We used Ripe Atlas network [6] to collect measurement data. Ripe Atlas network is composed of thousands of small hardware devices ("probes") that spread across the world and connect via the Internet. Through Atlas API, we can issue *ping* and *traceroute* commands from these probes and collect measurement results. To make sure that measurement results capture the characteristics of landmarks distribution from various geographic locations, we need observers to be distributed as evenly among the target areas as possible. We chose 14 probes from Ripe Atlas network as the observers as shown in Figure 1.

We collected our landmarks from three datasets: Ripe Atlas probes, university websites and city government websites. Figure 2 shows the landmark distribution. Table I lists the statistics

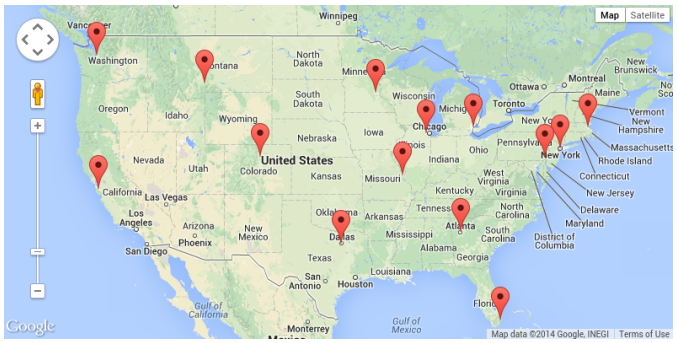

Fig. 1: Observer Locations

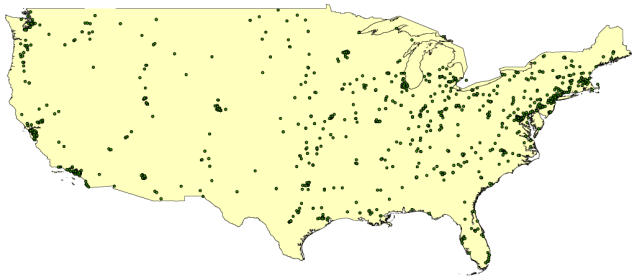

Fig. 2: Landmark Distribution

of these landmarks. We collected 3235 potential landmarks and 1547 of them returned valid ping results (successfully ping five times) that we used for our experiments.

Ripe Atlas Probes: Probe owners are required to provide physical locations of their probes when they register them. This information and their IP addresses are available to us. These probes are designed to respond to *ping* and *traceroute*, as long as their networks don't filter out these requests, which makes them a perfect choice for landmarks. At the time we wrote this paper, Ripe Atlas manages 1021 public and active probes in the United States. Comparing to PlanetLab platform that was widely used in earlier research ([3][4][7]), Ripe Atlas provides a lightweight hardware solution and thus lowers the threshold for participants. It also has a wider coverage than PlanetLab, which has fewer than 300 sites in US.

University Websites: Academic institutions are believed to have a higher chance to host their web servers locally. Thus University datasets were widely used in earlier research as either landmarks or test sets ([8][3][1]). However, the size of those datasets are limited ($\sim$150) and is too small for our purpose. Our goal is to maintain a complete university dataset covering all universities in the US. With information from [9], we retrieved a list containing 2170 distinct entries of university names and their states. For each entry, we automatically queried Google Websearch API [10] for the corresponding website, then used *host* and *whois* service to

Category	Raw	Valid	Reachable
Ripe Atlas Probes	637	637	429
University Websites	2170	1858	826
City Government Websites	2880	740	292
Total	5687	3235	1547

TABLE I: Landmark Detail (Raw: All landmark candidates. Valid: Landmarks after filtering and cross-validation. Reachable: Landmarks that respond to *ping*)

Hosting Service	Count
Rackspace Hosting	134
Amazon.com	102
Media Temple	34
Unified Layer	33
GoDaddy.com, LLC	29
Linode	27

TABLE II: Top Cloud Providers for University Websites

retrieve corresponding IP address, alias name and the owner organization name. We adopted various methods to filter out invalid data. For example, we exclude entries that use cloud services such as Amazon AWS, Rackspace Hosting and other top cloud providers to hold their web servers based on the well-known IP address ranges of these services. We also use Whois information to filter out organizations that don't contain keywords such as "university", "college" or "institute" in their names, but own more than one IP address from different universities. For example, if the IP addresses from University X and Y are held by company A named "Network Solution Corp.", we will notice University X and Y delegate their website to company A instead of running a web server locally. This kind of data will be filtered out.

Table II lists top cloud providers providing web hosting service for universities. 1858 IP addresses were left after the cleansing, and the corresponding geographic locations are again obtained through Google Websearch API. This constructed our University Website dataset.

City Dataset: Our University dataset contains landmarks spreading among most part of the United States. However, their coverage is more dense in the Eastern US and on the western coast, leaving gaps in Mid-west areas. We hope that the introduction of city dataset can mitigate this problem. Most US local governments have their own websites and we want to extract those who are hosting their websites locally. We retrieved US atlas data from [11], which gives us 38186 cities and towns together with their geographic locations and population. We assumed that bigger cities with more population had higher chances to host their web servers locally. We chose top 60 cities from each state, again made use of Google's Websearch API[10] and applied similar methods that we used to process University Dataset. We also cross-validated the IP addresses using MaxMind GeoLite City Database [12]. After preprocessing, we obtained 740 entries as our city dataset. Unlike the case of university dataset, where more than 85% of the universities host their website locally, only 25% of the cities are found hosting their website locally.

III. System Design

Our system consists of two tiers, each of which contains a neural network trained with data collected from the landmarks. Both tiers take measurement results from a target IP as an input and output a latitude-longitude coordinate. The first tier is trained using all of our training data including Ripe Atlas probes, university websites and city government websites. It is expected to locate a large region that contains the target IP. The second tier is then trained with only data from that region, to obtain the final estimate of target IP's geolocation.

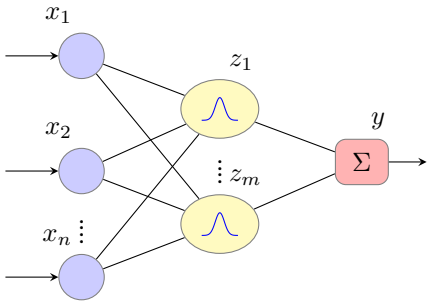

Fig. 3: Radial-basis function network

A. Training Data Generation

Assume that we have N landmarks, and k observers in the network. We send *ping* requests to each landmark from the observers, collect RTT as a vector $\mathbf{x}_n$ of size k. The landmark's physical location in the form of latitude and longitude can be recorded as vector t_n of size 2. Thus with the entire landmark set, we have an input matrix $\mathbf{X}$ of size $k \times N$, and a target matrix $\mathbf{t}$ of size $2 \times N$.

We adopt multiple methods to mitigate the impact of variance in our input data. First, by adding a regularization term to the error function, we make the network invariant to the variance in input. This is detailed in the next section. Second, our input data is designed to be multi-dimensional, hence variance in one dimension will be diminished in the final output. In practice, we did not observe a big impact from the variance in *ping* results.

B. Tier 1 - Radial-basis Function Network

In tier one, our task is to find the region in which the target IP resides. This is equivalent to finding the region that has the closest distance to the target IP in the Euclidean space based on the measured results. We achieved this by using a radial-basis function network (RBF) [13]. A Radial-basis function is a function in the form $h(x) = h(||x - x_c||)$. Each radial basis function has a center x_c, and the output only relies on distance $||x-x_c||$. A Radial-basis function network is a weighted linear combination of radial-basis functions. Figure 3 demonstrate a radial-basis function network. Let $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ be the input vector, $\mathbf{x}_j$ be the center point for the j-th radial-basis function, z_j be the outputs from radial-basis functions, and y be the network output. We have

$$z_j = h(||\mathbf{x} - \mathbf{x}_j||)$$
$$y = \sum_j^m w_j z_j$$

where w_i are the weight parameters to be learned from the dataset.

We first applied K-mean clustering algorithm to the entire set, determined an appropriate number of clusters, and obtained the center point of these clusters. Subsequently, we created radial-basis functions centered at them and trained the network.

We chose all landmarks from the lower 48 US states. With Wikipedia data of US extreme points [14], we calculated a minimal rectangle R = $[x_{min}, y_{min}, x_{max}, y_{max}]$ enclosing the lower 48 states. We then used the data to construct a regularization term Ω_b for the network. This regularization term looks like a basin-shape function returning near-zero values to points within a region and large values for points outside of that region. In other words, it "punishes" the results that fall out of the given region.

$$\Omega_b = \alpha_b(\text{sigmoid}(-x + x_{min}) + \text{sigmoid}(x - x_{max}) + \text{sigmoid}(-y + y_{min}) + \text{sigmoid}(y - y_{max}))$$

where

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-\lambda x)}$$

is the logistic regression function, α_b is the weight factor for this term and λ is the factor controlling the steepness of basin side.

For an input $\mathbf{x}$ and target value $\mathbf{t}$, we have $\mathbf{t}' = y(\mathbf{x})$ where y is the network function and $\mathbf{t}'$ is the network output. The error is given by $\epsilon = ||\mathbf{t} - \mathbf{t}'||$. We measured the mean value of ϵ on our dataset as ϵ_m. We then created a circular region centered on $\mathbf{t}'$, with radius ϵ_m. This region is the output from tier 1. In the case this region does not contain enough landmarks, we enlarge the radius until either we have enough landmarks in the region or an upper bound is met.

C. Tier 2 - MLP Network and Comparison Network

Within the region identified by Tier 1, we used a multilayer perceptron (MLP) network to do a more precise estimation that focuses only on that region. An MLP network is a variant of neural networks that consist of multiple layers of units (aka neurons). Each neuron is a computation unit with multiple inputs and single output. It takes a weighted sum of the previous layer's output, applies an activation function and returns the result as its output. Figure 4 shows a general example of two layers MLP, in which x_i are inputs to the network, z_i are outputs from hidden layers and y_i represent outputs from the network. We have

$$z_j = h_{\text{h}}(\sum_i^n w_{ij} x_i)$$
$$y_k = h_{\text{o}}(\sum_j^m w_{jk} z_j)$$

where h_{h} is the hidden layer activation function and h_{o} is the output layer activation function. The weight factors w_{ij} are the network parameters we need to determine.

We used a two layer MLP network with a logistic activation function in the hidden layer , and a linear function in the output layer. This structure had been proven capable of simulating arbitrary functions with enough neurons [15]. We chose the size of neurons in the hidden layer to be twice as much as input size after observing that the target value distribution follows a relative simple model in a local area. We then used standard techniques including error propagation and Levenberg-Marquardt method to train the algorithm.

To deal with aforementioned variance in the dataset, we added a regularization term Ω_v to mitigate its impact. In [16] Bishop

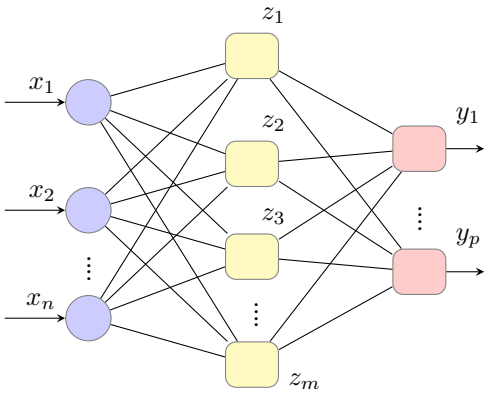

Fig. 4: Two-layer MLP network

gives an overview of such techniques in general, and we applied it to our special case.

$$\Omega_v = \sum_{i=1}^{n}\sum_{k=1}^{m}\sum_{t=1}^{N} J_{kti}^2$$

where α_v is the weight factor for this term and

$$J_{kti} = \frac{\partial y_k}{\partial x_{ti}}$$

is the $[k, i]$ element of the Jacobian Matrix of the network near data point $\mathbf{x}_t$. J_{tki} can be evaluated effectively using numerical differentiation.

Within the region in which the target IP resides, we use a regularization term based on logistic sigmoid function to constrain the estimated value to this region.

$$\Omega_r = \alpha \cdot \text{sigmoid}(||x - x_{\text{center}}|| - \epsilon)$$

where α and sigmoid is the same as described in previous section, x_{center} is the center of the target region and ϵ is the radius.

The training process of MLP is a non-linear optimization process. This means some results may represent local minima rather than reaching a global minimum. We mitigate this by repeating the training process multiple times and choose the one that perform best on the entire dataset.

To compare performance of MLP network with other regression models, we developed a comparison network. It is a framework that allow us to easily compare estimation performance between different models. In the comparison network, each regression model is implemented as a module. The framework copies input to different modules simultaneously and collects estimation output from them. It then compares these output values with target values, calculates errors and generates reports. As a comparison to MLP network, we again used RBF network in the second tier.

IV. Evaluation

In this section, we report the the estimation accuracy on each layer of the network. The metrics we have chosen for evaluation are also used by previous researchers ([3][1]), we believe these evaluations are able to provide a good comparison between our method and previous efforts.

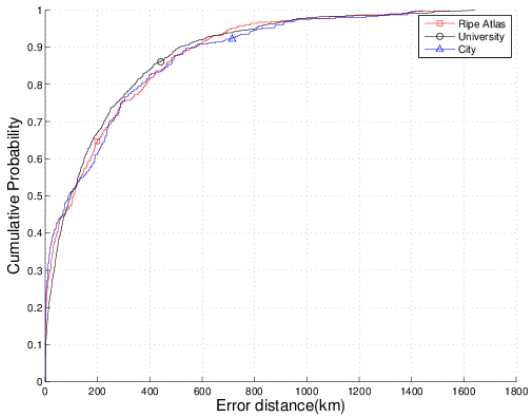

Fig. 5: Error Distribution in Different Dataset

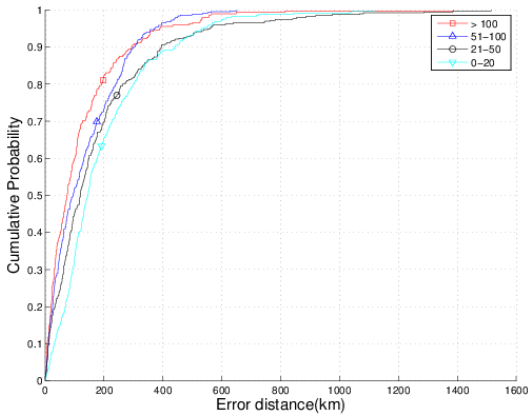

Fig. 6: Error related to Landmark Density

A. Tier 1 Estimation Results

Preference to Network Type: Figure 5 shows the cumulative distribution function (CDF) of error in estimation results with Radial-basis function network in Tier 1. With the curiosity to know whether our method will have preference to network type, e.g., academic, residence and commercial networks. We know that any preference to network type should lead to difference in their cdfs. However, the figure does not show significant difference between the three cdfs. This reveals that our method has no obvious preference to academic or commercial networks. The median errors in all three datasets are around 110 km and over 85% errors are less than 300 km.

Landmark Density Impact: We used 300km as a region radius within which 70% of the landmarks have at least 50 landmarks in surroundings. There are more landmarks in eastern US and on the West coast and fewer in the mid-west. We are also interested in whether our estimation results are impacted by landmark densities in an area. Figure 6 separates the landmarks into 4 bins and shows their cdfs. Not surprisingly, the errors are larger in regions with fewer landmarks. However, the difference is small. This illustrates that the radial-basis function network is relatively resilient to landmark density variance.

B. Tier 2 Estimation Results

To test the performance of tier 2, we constructed 1547 sub datasets. Each sub dataset consists of one centered landmark along with all other landmarks within a radius of 300 km. We used all surrounding landmarks to estimate the center landmark's location and obtain an error distance for each sub dataset. We empirically chose the MLP hidden layer size to be 60% of the input data size, and repeatedly trained the scheme for 50 times. The input dataset was randomly split into three portions: 80% as a training set, 10% as a test set and 10% as a validation set. The network with the smallest sum of squared error value on the training set is used to conduct final estimation. Figure 7 shows the error distribution. It can be seen that the median error of distance is 5.1 km, and in over 80% cases the error is less than 10 km. We repeated the same experiment using RBF network with the error distribution results shown in Figure 7. It can be seen RBF gives a slightly better result than MLP with a median error of 4.1 km. However, it has a longer tail and in the worst case the error distance reaches over 60 km. On the contrast, MLP has the worst case of around 35 km.

We were also interested in how MLP performance varies with landmark density. We showed such relationship in Figure 8. The x axis denotes the landmark count in each sub dataset, and the y axis denotes the error distance in km. The blue spots show the error distance of each region. We also studied how the median error changed with landmark density. We separated the data into 30 bins based on their landmark count (0-300) and obtained the median error in each bin. The red line in the figure shows this trend. It can be seen that the median error becomes slightly smaller with the increase of landmark density. In half of the regions where more than 100 landmarks are present, MLP has achieved a median error of 3.7 km, while in regions where less than 50 landmarks are present, the median error is around 6 km. We also showed how RBF performance varied with different landmark densities in Figure 9. It can be noticed that RBF's results are not as stable as MLP in different regions. We can notice two obvious peaks near x-point 130 and 180 reach around 40 km, while in MLP the median error is stably around 5 km. We also notice that the median error of RBF does not show a decreasing trend with the increasing of landmark density. Actually when counting the median error in regions with more than 100 landmarks, we had an even slightly higher result of 4.4 km. These results show us that in regions with higher landmark densities (greater than 100), MLP has better performance than RBF. Overall, these results justify our hypothesis that using only local landmarks can lead to a higher accuracy of estimation results.

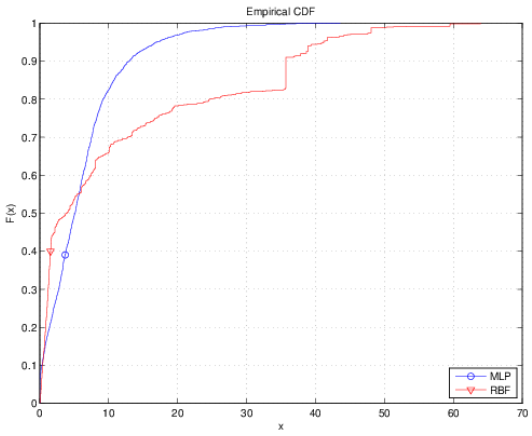

Fig. 7: MLP and RBF Error Distribution for Tier 2

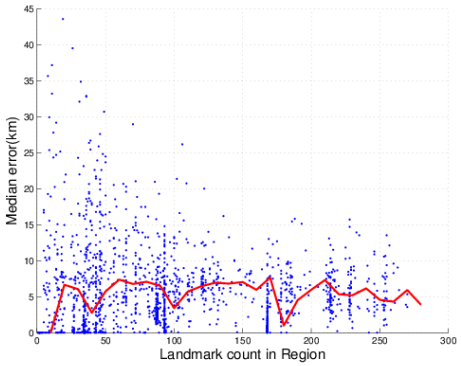

Fig. 8: MLP Error Related to Landmark Density

V. Related Work

Delay model based methods: A noticeable category of previous work in IP geolocation [8][3][4][17] was based on a delay model. These methods start with the assumption that network latency has a direct relationship with physical distances between nodes. The physical distances are then mapped to a Euclidean space in latitude and longitude. Mathematical models can then be applied to this space and the locations of unknown nodes are estimated based on other known nodes. These methods give promising results with a median error of around 30 km. However, they suffer from the inaccuracy due to the incorrect or simple distance-to-latency assumption.

Measurement similarity based methods: We were inspired by the work from Wang et al. [1], who proposed to determine a big area that target IP resides in, then collect landmarks in that area using data-mining techniques. However, we used different datasets to conduct experiments. In [5] Youn et al. described a method based on gradient-descent and forced direction. Eriksson et al. [18] introduced their method based on Bayesian approach to increase IP geolocation estimation accuracy. These methods are good examples showing data-based methods' potential in IP geolocation. Craig et al. [2] give a solution to the last-meter problem, allowing the location of an IP address to be mapped to a physical street address.

IP-Geo Databases: These databases provide a quick way to address IP geolocation problem. However, this method also suffers from problems such as outdated data, incomplete coverage and inaccurate information. For example, Huffaker et al. [19] conduct a survey on available IP databases in market. Their results show that IP databases have a median error (88 km∼727 km) higher than state-of-the-art IP geolocation methods and thus is incapable of being used to do IP geolocation alone. Nevertheless, IP databases has their advantages such as wide coverage and fast access speed. We can combine two methods by use IP databases to look for landmark candidates in a specific region, then update IP database entries regularly using our method.

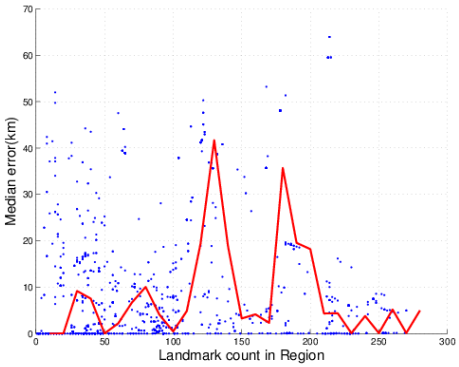

Fig. 9: RBF Error Related to Landmark Density

VI. Future Work

We are pleased with the ability of our algorithm to achieve accuracy with fewer landmarks, but we would like to see if we can simplify our approach. By systematically quantifying the differential contribution of each aspect of the current algorithm, we may be able to achieve a similar level of accuracy with less complexity. For example, neural networks are powerful tool that worked well for us on this problem. However, we are interested in further exploring the underlying properties of the problem that enable this result. We would like to conduct further experiments to compare different methods on the same data set.

In this study, we did not focus on cellular networks. Cellular networks may have different properties not represented in the dataset we collected. We are interested in collecting data specific to cellular networks and evaluating the performance of our methods in that environment.

Our method assumes that geograpically adjacent IP addresses will also be adjacent on network topology. This assumption has been justified by our research based on U.S. network. We are interested in testing whether this assumption holds in other areas of the world and would look to expand our testing in regions such as Europe and Asia.

VII. Conclusion

In this paper, we have shown a novel approach that accurately geolocates arbitrary IP address with fixed number of landmarks. Subsequently, we created a two-tier neural network system using a large dataset covering academic, commercial and residential networks. We also built a system that regularly updates measurement results and trains the schemes to keep our estimation result up-to-date. We tested our system on this dataset extensively, showing a similar accuracy as previous state of the art, with a fixed landmark dataset.

References

[1] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang, "Towards Street-level Client-independent IP Geolocation," in *NSDI '11*. Berkeley, CA, USA: USENIX Association, 2011, pp. 27–27. [Online]. Available: http://dl.acm.org/citation.cfm?id=1972457.1972494

[2] C. A. Shue, N. Paul, and C. R. Taylor, "From an IP Address to a Street Address: Using Wireless Signals to Locate a Target," in *Presented as part of the 7th USENIX Workshop on Offensive Technologies*. Berkeley, CA: USENIX, 2013. [Online]. Available: https://www.usenix.org/conference/woot13/workshop-program/presentation/Shue

[3] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards IP Geolocation Using Delay and Topology Measurements," in *IMC '06*. New York, NY, USA: ACM, 2006, pp. 71–84. [Online]. Available: http://doi.acm.org/10.1145/1177080.1177090

[4] B. Wong, I. Stoyanov, and E. G. Sirer, "Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts," in *NSDI '07*. Berkeley, CA, USA: USENIX Association, 2007, pp. 23–23. [Online]. Available: http://dl.acm.org/citation.cfm?id=1973430.1973453

[5] I. Youn, B. Mark, and D. Richards, "Statistical geolocation of internet hosts," in *ICCCN 2009*, Aug 2009, pp. 1–6.

[6] RIPE NCC, "RIPE Atlas Network," https://atlas.ripe.net/.

[7] Z. Hu, J. Heidemann, and Y. Pradkin, "Towards Geolocation of Millions of IP Addresses," in *IMC '12*. New York, NY, USA: ACM, 2012, pp. 123–130. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398790

[8] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based Geolocation of Internet Hosts," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1219–1232, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.886332

[9] Wikipedia, "Lists of American institutions of higher education," 2014. [Online]. Available: https://en.wikipedia.org/wiki/Lists_of_American_institutions_of_higher_education

[10] Google Inc., "Google Websearch API," July 2014. [Online]. Available: https://developers.google.com/web-search/

[11] National Atlas, "Cities and Towns of the United States," October 2013. [Online]. Available: http://nationalatlas.gov/mld/citiesx.html

[12] MaxMind, "GeoLite City Database," http://dev.maxmind.com/geoip/legacy/geolite/.

[13] R. Lippmann, "Pattern classification using neural networks," *Communications Magazine, IEEE*, vol. 27, no. 11, pp. 47–50, Nov 1989.

[14] Wikipedia, "Extreme points of the United States," https://en.wikipedia.org/wiki/List_of_extreme_points_of_the_United_States.

[15] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989. [Online]. Available: http://dx.doi.org/10.1007/BF02551274

[16] C. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.

[17] Z. Dong, R. D. Perera, R. Chandramouli, and K. Subbalakshmi, "Network measurement based modeling and optimization for {IP} geolocation," *Computer Networks*, vol. 56, no. 1, pp. 85 – 98, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611003173

[18] B. Eriksson, P. Barford, J. Sommers, and R. Nowak, "A Learning-based Approach for IP Geolocation," in *PAM '10*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 171–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1889324.1889342

[19] B. Huffaker, M. Fomenkov, and k. claffy, "Geocompare: a comparison of public and commercial geolocation databases - Technical Report ," Cooperative Association for Internet Data Analysis (CAIDA), Tech. Rep., May 2011.