Pinpointing Location using Decision Trees

Subtitle as needed (*paper subtitle*)

Arman Elahi
Department of Computer Science and Engineering
Santa Clara University

***Abstract*—To replace expensive and battery intensive GPS location based services, by mapping routers in an area, a user's location is easily ascertained.** *(Abstract)*

I. Introduction

For students and faculty of Santa Clara University, the campus may seem small. For a newcomer such as a prospective student or a visiting parent, the campus seems much larger. Just being able to articular where your location is an important asset in trying to determine your route to any destination.

Although GPS is the predominant technology for finding location, it does not always provide contextual names for buildings or even any names at all. In addition, keeping GPS drains the battery of mobile units.

To help visitors find their way through campus, I have developed a program to predict location based on routers, the RSSI of each router, and the relative signal strength. The project takes shape in two phases, the first is data aggregation and the second is prediction.

II. Data Aggregation

The first phase of the project was to aggregate as much data as possible about the Signal Strength around Santa Clara University. This entailed creating an Android application to gather this sort of data. Due to commute time and not being easily able to access campus, when I was able to come to campus, I needed to make sure that I gathered more than enough information to predict location. In addition to gathering the BSSID, RSSI, and Cell Signal Strength, I also gathered the Link Speed and SSID. Along with a manually entered location name, I was able to gather about 2k measurements over 3 hours.

A. Data Labelling

The data is kept as a CSV on the phone. Anytime the user hits the plus button, the application starts a Wi-Fi scan to gather results on Wi-Fi signal strength. In addition, the various other fields are also populated.

One of the key decisions in aggregating data was to label points manually or allow the system to gather GPS coordinates for each point and create bounding boxes for locations. In the end, I tried both methods. I had a build which polled GPS coordinates every 3 seconds to get an approximate location and the current signal strength. When examining the logs for this, it became clear that the data was quite noisy. The refresh rate for the GPS was too slow to catch some of the very minute differences in location such as the difference between the San Filipo Lawn and the Campisi Lawn. This caused new Cell Signal strength data to be associated with old GPS locations. By manually labeling points, these minute differences were easily accounted for.

B. Dealing with Border APs

For most access points they are clearly located in one building. But between co-located buildings such as Sobrato and Casa which share a very narrow pathway, you may still be connected to one AP with a good enough signal strength. These areas have more collected data than other places simply to increase the accuracy of the predictor.

It is due to these cases that simply using an access point to determine location was not feasible. These border APs would also catch many measurements during a handoff, so temporarily in a dead zone. To combat this, Cell Signal Strength was added as a dimension. This dimension measured the cell reception from nearby 4G cell towers. Although this varies based on carrier, the coverage amongst major carriers in Silicon Valley was high enough to be overlooked.

C. Ensuring quality measurements

To ensure that we were connected to the closest router possible, the data was collected in low traffic times. These times include 11pm – 12am and 6am – 7am. By collecting at these times, I had more confidence that. The Android application developed for this project shows the recorded values on screen for a quick glance at the data recorded. This ensures the user taking measurements that their data is recorded properly.

III. Data Cleaning

Once the data was aggregated, minor cleaning was done. This cleaning revolved around outliers in dead zones. When no Wi-Fi connection was available, the only feature that was still viable was the cell signal strength. Given the low significance of this variable, it did not appear to be a reliable indicator of location. Entries which indicated an unknown-ssid, signifying

no Wireless connection were dropped. In addition, measurements were taken for only one SSID. All measurements which had an SSID other than SCU-Student were also dropped. Because the initial logs do not contain headers for each column, the column headers were written to the top of the file after all measurements were collected.

IV. Data Preprocessing

Once all the data had been collected as a CSV, there came the need to fit all the data types so that sci-kit learn was able to digest it. Most variables are continuous values such as RSSI, Cell Signal Strength, and Link Speed. These were fit into the decision tree. The large signifier though, the connected router is categorical data. There are no ranges of routers, each router's BSSID is independent of it's neighbors. To transform this categorical data into digestible data, a method called One Hot Encoding was used. Each of the unique routers now corresponded to a dimension. This took the original 6 dimensional vector collected and turned it into a 191 dimensional vector. Each of the added dimensions was a binary dimension indicating the router that gave the indicated measurements in the last 4 columns. This created a very dense last few columns and a very sparse section of columns.

Locations were hashed to numeric values in the training set. A hash table was created to store these values.

In mock runs of the program, the test set is made of a subsection of the entire dataset. This introduces a small amount of error in the case that all points belonging to a certain label were chosen to be part of the test set. Because the test set was made from the original dataset with One Hot Encoding, the data was already in the format needed to apply the algorithm. The actual execution of fitting was easily done using existing libraries in sci-kit learn.

V. Results

The results are quite promising even with the limited amount of data. To actually cover the campus in a three dimensional space, it would have been important to take measurements from levels other than the ground floor. This was not a primary concern of the dataset, as the eventual goal of the project was to find relative building location. The measurements in Table 1 are taken as the average value of fitting the model and predicting the labels over 10 runs.

TABLE I. Error Percentage Based on Size of Training Set

% Used as Training Set	Average Error %	Standard Deviation
10	51.44	4.50
20	33.56	2.50
30	26.34	3.05
40	22.05	2.57
50	17.24	1.46
60	14.54	1.81
70	13.22	1.55
80	12.47	1.99
90	9.84	2.10

A. Analysis

From the data in Table 1, it is clear to see that the higher the percentage used as the training set, the lower the average error is. When factoring in the Standard Deviation though, it seems as though using 70% of the data as the training set produces the best results. To achieve an accuracy of at least 87%, there must be more than 1,750 measurements in the training set.

As discussed before, there is some error introduced by the sampling method. The relative size of areas though is accounted for in the dataset. Although not perfectly proportional, the relative number of measurements per Location is roughly proportional to the number of measurements in the dataset. Larger buildings are accompanied by more measurements and thus reflect the real probability that a measurements would have been taken in that vicinity.

The use of a decision tree was very purposeful. Due to the low dimensionality of the data, the tree is not particularly deep. This allows the decision tree to have a tolerable amount of loss. Most labels are provided by the router. Of the 2500 records gathered, there were 186 unique routers, only 45 of which had multiple locations associated with them. 75% of locations were determined by router alone. Only 25% of all access points were on the border of two buildings.

B. Improvements

Some obvious improvements to the methodology are apparent. First, having a true test and training set is a priority. Although the deviations from the average were quite low, only about 2%, the standard deviation can be lowered by having a more consistent training set. Second, using GPS to create bounding boxes for collecting the data would have expedited the process and given much more accurate results. Lastly, Santa Clara broadcasts 3 different SSIDs, each of which reside on a different Wi-Fi channel, factoring this data would help to lower the error percentage for the 25% of border access points.

VI. Conclusions

Through the results gathered for small sets of data, it is clear that the connected routers is a good indication of relative location. The RSSI provides a good secondary layer to back up the label provided through just routers.

Some next steps for this project would be to factor in angle of arrival into the calculation. In addition, setting up a service for visitors to pinpoint location would also be interesting. Another interesting point would be to assemble k-partite graphs in order to route users across a Campus Area Network.

References

[1] Q. Fu, H.Gao, F.Shang, W.Su, Q. Wang, "A Location Estimation Algorithm based on RSSI Vector Similarity Degree," *International Journal of Distributed Sensor Networks*, vol. 10.8, August.

[2] R. Gatej, S. Lehmann, P. Sapiezynski, A. Stopczynksi, "Tracking Human Mobility using WiFi Signals," 10.1371/journal.pone.0130824