INFRASTRUCTURE SIDE POSITIONING OF CELLULAR BAND DEVICES

Nidhi Mendiratta Neeharika Kondipati

{nmendiratta, vkondipati}@cs.stonybrook.edu

Abstract

The main aim of our project is to build a scalable infrastructure to handle the location prediction mechanism in an indoor environment. It includes building a central repository which had all the signal measurements from near by towers which would be picked up by an unsupervised and semi supervised algorithm treating it as a stream of data and accordingly train and test the model on same.

I. INTRODUCTION

The process of finding the spatial location of nodes in a wireless network is commonly known as localization. Wireless sensor network localization is an important area that attracted significant research interest. This interest is expected to grow further with the proliferation of wireless sensor network applications. Our project is one such attempt to build a scalable infrastructure to handle location prediction mechanism. Our main focus is on cellular band devices and also to implement location prediction using semi supervised and unsupervised learning techniques.

The set of RSS values that are collected for each position in the map from various towers or base stations is called the fingerprint for that location. The idea of matching observations of RSS to the map of the previously measured RSS values is known as fingerprinting and is a widely used concept as a ranging technique for localization.

A global positioning system (GPS) can be used to obtain location information. But it does not work indoors. While today's more sensitive GPS chips can sometimes get a fix (receive signals from enough satellites to determine a location) inside a building, the resulting location is typically not accurate enough to be useful. The signals from the satellites are attenuated and scattered by roofs, walls and other objects. Besides, the error range of many GPS chips can be larger than the indoor space itself (small grocery store). The limitation of GPS has motivated researchers to develop algorithms to infer location using cheap hardware by leveraging network connectivity, signal strength, and angle-of arrival information. No single solution works perfectly in all environments. For that reason devices may support more than one positioning solution and switch between them as needed. Today's mobile phones use GPS (when it's turned on) outdoors but may switch to Wi-FI positioning (when it's turned on) when the signal is weak, such as when an individual goes indoors.

Indoor positioning is in demand for a variety of uses. While the goal of indoor positioning for some users, notably hospitals and malls, is to provide navigation aid, others want to use indoor positioning to better market to customers, provide just-in-time information via audio for tours, offer video or augmented reality experiences or connect people of interest in proximity to one another.

In our project, we chose the approach of semi supervised and unsupervised approach because collecting a sufficient amount of 'labeled' data, however, requires effort on the part of the operator which is very cost intensive. Therefore we minimized this cost by taking very less or no labelled data and train our model accordingly.

II. LITERATURE SURVEY

Related work in the field of user location and tracking are as follows :

Radar : An Inbuilding RF based user location and tracking system

This paper was the first of its kind to pick up the problem of indoor localization. There was lot of ongoing research in outdoor localization but for indoor, not much work had been done. This team at Microsoft used radio fingerprinting techniques for predicting the location. The key feature was that it performed better than random selection and strongest base station method. But they faced the huge problem of collecting lot of supervised data which is an ad hoc task.

Network side positioning of cellular band devices with minimal effort :

This paper is the main motivation behind this project. Our main goal is very similar to theirs , the only change is that our main focus is on building a scalable system for similar problem handled in this paper. This paper also focuses on unsupervised and semi-supervised learning techniques and they use the expectation maximization model on signals received from cellular band devices. Instead of picking up random initial parameters, they initialized their model using parameters derived from path loss model and the advantage of not having the need of labelled data makes data collection a much easier process for them.

III. MODELLING APPROACH

For the given problem statement, we started with the major step of collecting data for the same, after which once the central repository was intact, we trained it following both unsupervised as well as semi supervised approaches. For handling the database, we selected mongodb to maintain our repository. This decision was based on the fact that the signals reaching any particular position are not fixed. Therefore we can never completely define the number of signals received, which is why we moved to a non relational database to handle the same.

In Mongo, we created a collection to store all the location values and each of these location values were calculated using Path loss model. Path loss model propagates the idea of reduction in power density while travelling through space. Path loss may be due to many effects, such as free-space loss, refraction, diffraction, reflection and absorption etc.

For our project, we randomly selected the location of transmitters using python's random generator and based on that we calculated the signal strength or rssi values for every location for set of transmitters using the path loss equation :

$$PL = P_{Tx_{dBm}} - P_{Rx_{dBm}} = PL_0 + 10\gamma \log_{10}\frac{d}{d_0} + X_g$$

where,

PL is the total path loss measured in decibels.

P_{Tx} is the power transmitted by the transmitter.

P_{Rx} is the power received at the point.

$PL_0$is the path loss at a reference distance d_0

γ is the path loss component which was taken as 2.5 to generate the vectors.

d is the distance between the transmitter and the point we are calculating the signal strength.

d_0 is the reference distance.

X_g is the normal gaussian random variable with zero mean.

We find the power received at the point from all the transmitters and store it in the form of a vector. For each grid in the total area we generated 20 vectors by varying the standard deviation in the path loss equation.

The output of the pathloss model was stored into the mongo collection by creating a sample vector like :

vector : [-71.32254073649675, -51.94650989675746, -93.22147531402024,-80.33110326107565,-100.41143155448378, -68.23121858044985, -64.40242408397933, -76.18269819511593,-86.49651648287428, -91.41238266000983]

Also ,we made sure that if there was any signal strength not received , in that case we would take the value to be -120 dBm. Also to make sure that enough data is there for the model to train well, we generated 20 such vectors for each location. Therefore for a sample grid size of 5km x 5km , 10 transmitters and 10 grid resolution

Total cells were (5000/10)*(5000/10)

Total vectors were total cells *20.

Therefore using the above approach, we created a corpora of vectors which was used by our model to predict the location.

IV. LEARNING TECHNIQUE

In machine learning, there are different types of learning techniques which can be majorly classified as supervised , unsupervised and semi supervised techniques. Supervised learning technique signifies that along with

data values to be trained, the corresponding target values are also fed in for appropriate training.
For unsupervised method, no target values are passed and the model is trained only on specific set of inputs. In this technique the model learns by finding the structure or relationships between different inputs. For semi supervised learning, the target values are passed only for some values and not with all the inputs.

As already stated, in our problem statement calculating the target values is a very cost intensive task, which is why we chose unsupervised and semi supervised learning techniques. For either of these techniques, instead of algorithm picking random initial parameters, we are modifying our start point by generating the initial mean and variance using path loss model.

V. TRAINING ALGORITHM

We have used Expectation Maximization approach for this project. We used the existing GMM of python as our training algorithm. EM is the soft clustering algorithm unlike k-means which also improves the parameters in rounds based on the errors of the previous models. This algorithm is specifically useful when we have partially labeled data, where there are relatively few training examples confidently assigned to the correct class. We can build the classifiers based on the training examples, and use them to assign the unlabeled points to candidate classes. We have used this to implement the semi supervised model which is explained later.

Apart from the existing EM, we made some changes in the way data is fed into the algorithm. For this we implemented the functionality of training data in batches.

Batch GMM : This approach of training the algorithm in batches is implementing using the warm start functionality in python gmm. Warm start is nothing but a parameter which can be passed to the model. This ensures that for a given batch size , model is trained once and is saved. For the next iteration, model is picked up from the last saved step and trained on the next batch size.

Submodel Approach : This is another modification done, which ensures that instead of taking in scope the entire grid size all at once, the grid is divided into sub models considering the fact that any model trains much better for smaller values. Therefore, this approach helps in determining the exact localization error which as will see in results is much better than taking the entire grid size.In the above scenario, for the entire model (5000x5000 area), the number of classes for a grid resolution of 10 will be (5000/10x5000/10) = 250000. For such a huge number of classes, there will be very low prior because if value of feature vector is sufficiently far from the means, probability will be zero, which in turn will result in a situation of underflow. Therefore instead of training one complete model, we implemented a submodel approach wherein, our entire grid size was divided into submodels and each of those submodels were trained using batch gmm approach. We need a heuristic to map the data point(train/test) to the appropriate sub model. The heuristic is explained and the implementation details are explained in the evaluation section.

The above mentioned modifications will ensure the following :

- Scalable : Since the focus of this project is on building a scalable system to handle larger grid size which implies larger data set.
- Robust : It can handle erroneous situations where the training data vector do not have proper signal values.
- Faster : The approach of training in batches turns out to be much faster than training on all data at once.

VI. EVALUATION

We need a heuristic to map the data point(train/test) to the appropriate sub model. In the real scenario, continuous stream of training data points are to be trained with our model. For the purpose of this project as stated earlier we have considered the path loss model to generate vectors. These vectors are mapped to the respective sub models to be trained using heuristics. Also the same heuristic is used to map the test data point to the respective sub model to get the predicted location. Before going into the details of the heuristics used, we will see in detail how sub models are created by partitioning the total area (REM). If the total area is 1000m*1000m and the area partition is n , then we will have n*n submodels each with area of 1000/n*1000/n square meters. For this experiment we have considered a uniform distribution of submodels with same area ignoring the irregularities in the terrain. For data point to submodel mapping we have implemented the following heuristics

Naive heuristics: We find the transmitter from which the given point receives the maximum strength and map the data point to the sub model to which the transmitter is closest. Using this heuristic we achieved a localization error of 190m for 1 km * 1 km area , 20m grid resolution, area partition = 5 and 10 transmitters (For the above configuration we had 40k training points and 10k test data points)

Triangulation heuristics: We find the top 3 transmitters from which the given point receives highest signal strength and mark them as anchors for triangulation method. Using the path loss model we calculate the distance between our point and the 3 transmitters, then locate the point using the triangulation method. Using this heuristic we achieved a localization error of 160m for 1 km * 1 km area , 20m grid resolution, area partition = 5 and 10 transmitters. We can observe an improvement of 30m localization accuracy.(For the above configuration we had 40k training points and 10k test data points).

We have a flask REST API to predict the location of the user using the test data point of rssi vector. For the purpose of evaluation we inserted the test data along with the training data into the mongo server. We take the data points from the server and use that submodel for location prediction which was mapped using the heuristics mentioned above. Actual GMM predicts the class for the data point which has the highest log probability. To avoid the errors due to hard clustering we used the log probability vector to find an intermediate location. Instead of selecting the location with highest probability, our model considers all the probability outputs and then calculates the final predicted location by multiplying probability with the location.

The sub model approach was applied to both un supervised and semi supervised model. In case of semi supervised model we considered few training points with labels and used them as anchors in the Expectation Maximization model. We incrementally improved the amount of labeled data and observed that the localization error decreases as the model learns better from the anchors.

VII. RESULTS

In this section we will discuss the results we achieved by varying each of the parameters and plotting the localization error for it. Various parameters involved in a configuration constitute number of transmitters, grid resolution of the area, total area into consideration, partition of the area for the sub model formation.

Area vs Localization error (Fig 1):

- Number of transmitters = 10
- Grid resolution = 20
- Area partition = 5

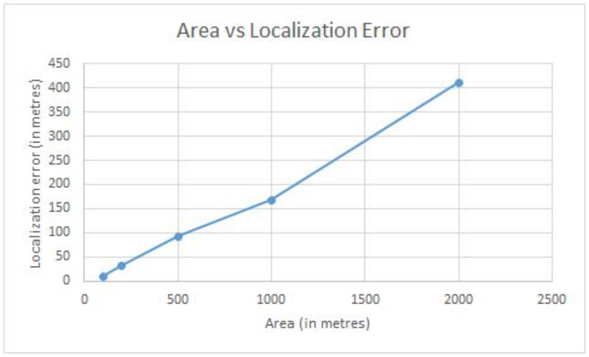

Fig 1. Area versus Localization error

We observed that for smaller area the localization error is low, this is the motivation behind the sub model approach. Dividing the total area into many sub areas each being trained helps to avoid underflow (means tending to zero).

Number of transmitters vs Localization error (Fig 2):

- Total area = 1000m (square grid of 1 km * 1 km)
- Grid resolution = 20
- Area partition = 5

As we increased the number of transmitters, the localization error has decreased which is expected because the information of the feature vector is being increased. Even though the positions of the transmitters were randomly chosen in each case, we can observe a decrease in localization error. Usually in an ideal case the positions of the transmitters also affects the localization error.

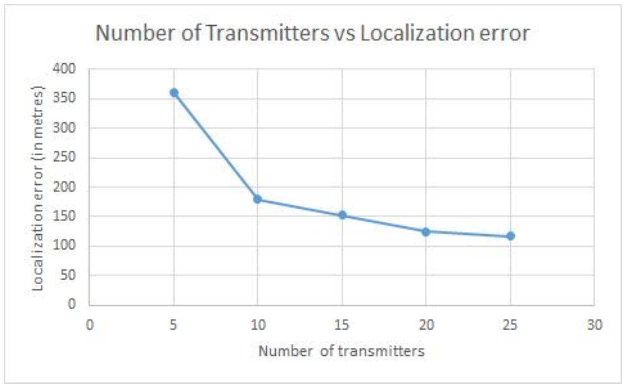

Fig 2. Number of transmitters versus Localization error

Grid Resolution vs Localization error (Fig 3):

- Total area = 1000m (square grid of 1 km * 1 km)
- Number of transmitters = 10 (positions of the transmitters are kept constant)
- Area partition = 5

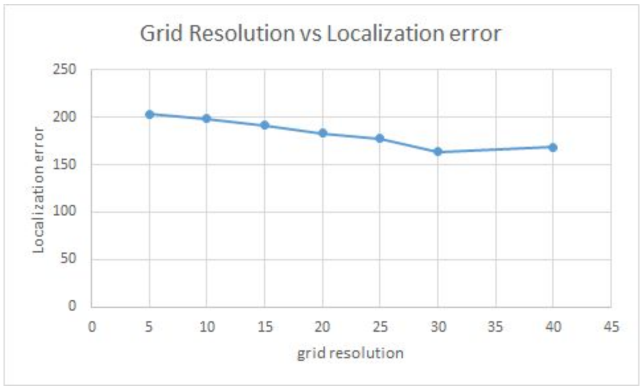

Fig 3. Grid Resolution versus Localization error

We did not observe a perfect pattern in case of grid resolution. Although we can say that as the localization error decreases with increase in grid resolution, after a particular threshold the trend seems to change and the error increases.

Area partition vs Localization error (Fig 4):

- Total area = 1000m (square grid of 1 km * 1 km)
- Number of transmitters = 10 (positions of the transmitters are kept constant)
- Grid Resolution = 20

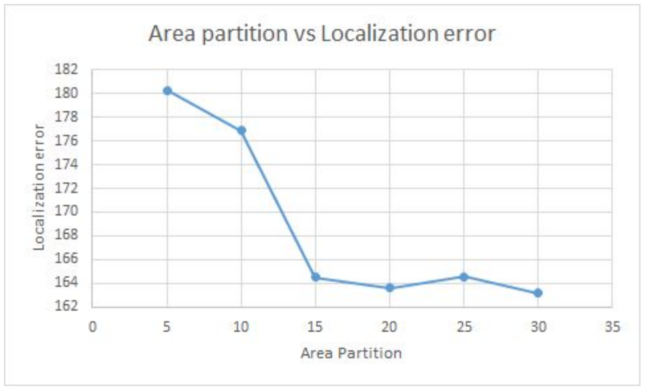

Fig 4.Area partition versus Localization error

Similar to the trend observed in case of grid resolution, we did not observe a monotonic trend for localization error versus area partition. We observed steep decrease in the error from area partition 10 to 15. Too few partitions has the problem of underflow i.e more classes to classify in each model, also too many partitions challenges the performance of heuristics. We can say that a area partition of 20 can be said an optimal value for this configuration.

By varying the amount of labeled data, the localization error is plotted for the semi supervised model. The following configuration is used for this experiment.

- Total area = 1 km * 1 km
- Number of transmitters = 10
- Grid resolution = 20
- Area partition = 5
- Total train points = 40k data points
- Total test points = 10k data points

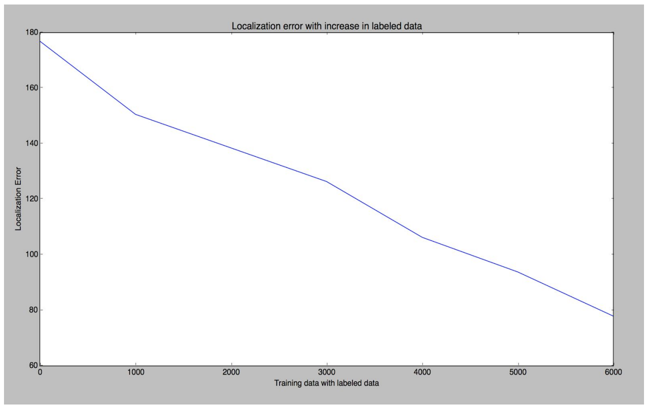

Fig 5.Amount of labeled data versus Localization error

We have compared our model with the performance of other models and made the following observations.

Configuration used to compare the models:

- Total area = 1000m (square grid of 1 km * 1 km)
- Number of transmitters = 10 (positions of the transmitters are kept constant)
- Grid Resolution = 20
- Area partition = 5

Observed Localization error for each of the models:

- Gaussian Naive Bayes : 108 m
- GMM(without batches) : 464m
- **GMM batches+submodel : 167m**

We also observed the time taken to train the complete model was much higher than the time taken to train the model in batches with sub models. This is because we do not have to classify too many classes at a time in case of submodels (i.e for a submodel of area 200m * 200m and grid resolution 20m we have 100 classes). If we consider a total area of 1 km * 1 km it is easy to train 25 models with 100 classes each than to train a model with 2500 classes at a time.

References

Network-side Positioning of Cellular-band Devices with Minimal Effort
Ayon Chakraborty, Luis E. Ortiz and Samir R. Das Computer Science Department, Stony Brook University, Stony Brook, New York.

RADAR: An InBuilding RF-based User Location and Tracking System
Paramvir Bahl and Venkata N. Padmanabhan

Sklearn Library
Used the library for gaussian mixture model Expectation Maximization, github link

Python Localization Library
Used the library for calculating triangulation heuristic, link

Acknowledgments

We gratefully acknowledge the support of Prof. Ayon Chakraborty, course instructor for his support during our project.