Analysis of Indoor Localization using phase differences in LTE signals

Vishal Nayak
Computer Science
Stony Brook University
Stony Brook, USA
vnayak@cs.stonybrook.edu

Ashish Chaudhry
Computer Science
Stony Brook University
Stony Brook, USA
ashchaudhary@cs.stonybrook.edu

***Abstract*— Long Term Evolution (LTE) is a standard for wireless communication of high-speed data for mobile phones and data terminals. The goal of LTE was to increase the speed and capacity of wireless data networks [1]. The signals are transmitted from the source with the same phase. When the signals reach the destination, the phase of signals will be different due to the channel characteristics. The signals are transmitted at specific frequency bands using many sub-carriers. If various samples of signals are captured from many different sub-carriers from a particular location, it represents the fingerprint of the location. If many such fingerprints can be recorded and modelled using machine learning algorithms, then this information can be used to predict the location of the device from the signal data.**

***Index Terms*—Localization, LTE, Weka, confusion matrix**

I. INTRODUCTION

The localization of devices using phase difference have been done in the past, using RFID tags [2]. But in this report we capture the analysis of localization using phase differences of the received LTE signals from various locations. The focus is mainly on the analysis of the indoor locations. We used RTL-SDR to capture the signals. These devices are based on Realtek RTL2832U and can be used as cheap SDR, since the chip allows transferring the raw I/Q samples to the host, which is officially used for DAB/DAB+/FM demodulation [3]. The signals received by the device was later provided as input for LTE Tracker. LTE-Tracker is open source software that continuously searchers for LTE cells on a particular frequency and then tracks, in real-time, all found cells [4]. The phase information of the signals from the subcarriers are captured from the application and are used as an instance of the fingerprint of the location. The analysis of a deluge of such fingerprints is done using Weka. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization [5].

This paper captures the method adopted for capturing the data, explaining in detail the steps to install the tools and capture the tool, characterization of locations used to capture the data, models developed and the observations. This paper also explains about the features and implementation details of the tool developed to ease the analysis in future.

II. DATA COLLECTION

LTE downlink works in different bandwidths like 1.4MHz, 3MHz, 5MHz, 10MHz, and 20MHz. In the RTL-SDR device mentioned above, can only tune up to 2MHz. We chose 1.4MHz for our analysis. In the LTE signals, both frequency and time are multiplexed using orthogonal frequency-division multiplexing (OFDM). OFDM is a method of encoding digital data on multiple carrier frequencies [6]. In each frequency band, there will be 6 Resource Blocks (RB) each of width 180 KHz. This sums up just above 1MB. Extra bands (guard bands) will be present. Each RB will have 12 Sub-Carriers (SC) each 15 KHz apart. With 1.4M, there will be 72 subcarriers. Signal strength and the phase of the signal in each subcarrier (SC) represents the signature of location. Both the signal strength and the phase can be captured using LTE-Tracker, but in this case only phase information is captured. The frequency used for analysis is 739MHz. LTE-Tracker detects various cell towers. Each cell tower is identified by cell ID and has 3 ports: 2 MIMO (Multiple Input Multiple Output) ports and one pilot channel sink. The collected data will have 72 phase values one pertaining to each subcarrier belonging to a port. So from 3 ports, the data collected will be a tuple of 216 values at any particular instant.

12 different locations were chosen for analysis and they are all from within an apartment: hall, kitchen, bedroom, washroom, veranda, portico, etc. Out of these, few were chosen such that the distance between them are same but they differ with respect to the obstructions to the signal. Example, for locations say [A, B, C], distance between [A, B] and [A, C] are same, but between A and B there is a clear path for signal whereas locations A and C is separated by a wall. This provides a direction for the analysis of the dependence of the distance on the localization. From each location 1000 tuples were collected. This data collection was performed on 5 different days.

III. Tools and Application Developed

Two tools are developed for analysis.

A. Raw Data Parser

Weka tool is used for analysis and the input for Weka tool should be in ".arff" format [5]. RawDataParser converts the data captured from the LTE-Tracker into the format which Weka accepts.

B. Indoor Localization Application

The percentage split analysis (as you'll see later) could not be performed by the Weka Explorer application distributed along with weka.jar containing all the machine learning algorithms. This was the motivation to come up with an application to do this job.

This is a GUI application which enables the user to select the input file of choice for analysis along with the percentage of random input data to be used and the number of iterations to be run. This application also saves the model created for each iteration which can be used later on to re-run the data to verify the results. This application is built using the weka.jar as the backbone.

Below is the snapshot for the application developed that will automate every task for all the data collection and analysis like building the confusion matrix and giving the precision.

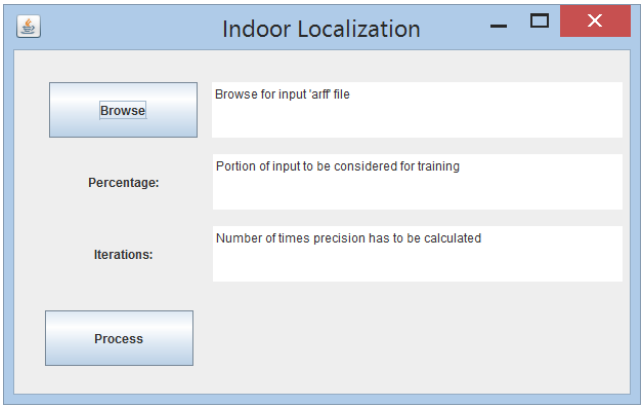

Fig 1. Indoor Localization application UI

IV. Analysis & Observations

The classifier used for analysis is Naive Bayes, a classifier of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features [7].

A. Approach 1:Analysis of the data captured on a single day.

From the data for 1 day (12000 tuples), extract the data randomly from the input and use it to build the model. From the rest of the data, test the created model. This was done with the following combinations: (90% train data, 10% test data), (80% train data, 20% test data), (70% train data, 30% test data), (60% train data, 40% test data), (50% train data, 50% test data), (40% train data, 60% test data), (30% train data, 70% test data), (20% train data, 80% test data), (10% train data, 90% test data).

For each combination of test and train data percentages, a confusion matrix was created and the precision values were collected.

This effort was run for 50 iterations (The Indoor Localization App is developed to handle this). The precision values were collected for each iteration.

The output files of the application run will also print the classification summary, kappa statistics, class complexity, root mean squared error, confusion matrix, fMeasure, precision, recall value and many more each run of the model. Finally, after all the iterations the output will also contain the mean, median and standard deviations for all the precision values.

The median of the precision values are plotted against the percentage of train data used for the creation of model. The standard deviations for each set from various iterations is represented as the error bar on the plot. A regression line is plotted for all the values. This provides the clear idea as to how the precision is varying with different sets of training information for the creation of data model.

- Observation :

Day 1: The precision values ranged from 80% to 82%, showing a slow increase in precision with the increase in the percentage of train data.

Day 2: The precision averaged around 53%.

Day 3: The precision averaged around 63%.

Day 4: The precision averaged around 67%.

Day 5: The precision averaged around 63%.

It was not known why the results were remarkable for day 1. In order to stress test the results, another approach was employed.

B. Approach II:Analysis of the data captured from 5 different days

All the analysis is similar to what is done with data from single day. The only difference is that the data set is enormously huge compared to the ones used for day 1. Also, the goal was to analyze the behavior of the models if data is collected with sufficient time durations between each data capture, along with different weather conditions.

- Observation :

The precision values drastically dropped to 30% to 33%. The increase in the precision on the regression line was still be able to be observed, but it was insignificant.

One of the analysis that we did was whether we would be able use LTE signals in indoor localization by distinguishing a point (A) from a point (B) within a particular distance range

(say 2 meter) in the same room to a point (C) which is also at the same distance from point A but in some other room or behind some doors so that the propagation to this point is not the same as the above two points A and B. We initially had the intuition that because signal will follow Log-distance path loss model, so the signal at point A and B would be nearly same and the machine learning tool (Weka) using Naive Bayes algorithm would be a bit confused within these two locations while between point A and C the algorithm would be able to distinguish and not get confused as the points will be receiving different signals. As mentioned out of the 12 locations that we took the data. There 3 set of points that were considered for this analysis. [A,B,C] as [7,5,6], [7,8,6], [9,10,8]. We took random percentage (10%, 20%, 30%, 40%, 50%, and 60%) of data at these locations and run Weka on the data set taking these as training set and passing the test set as the combination of any two locations to confuse the algorithm. Here we are presenting the Heat map for 40% random data (Heat Map for Naive base confusion matrix 40%.PNG) for location set (9, 10, 8). And also we are portraying three confusion matrix, one for the whole 12 locations for 40% random data and other two are for the set of location (9,10) and (9,8).

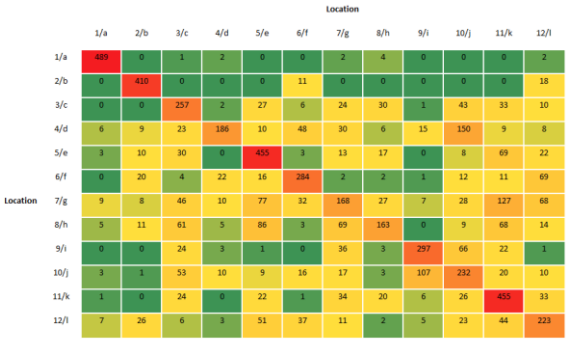

Fig 2. Heat Matrix for 40% Random Data

Classifier output

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.978     0.006      0.935     0.978     0.956      0.998    1
                 0.934     0.015      0.828     0.934     0.878      0.993    2
                 0.594     0.049      0.486     0.594     0.534      0.916    3
                 0.372     0.01       0.765     0.372     0.501      0.915    4
                 0.722     0.055      0.603     0.722     0.658      0.942    5
                 0.641     0.028      0.644     0.641     0.643      0.948    6
                 0.277     0.044      0.414     0.277     0.332      0.781    7
                 0.33      0.021      0.588     0.33      0.423      0.832    8
                 0.656     0.025      0.677     0.656     0.666      0.959    9
                 0.482     0.066      0.389     0.482     0.43       0.904    10
                 0.732     0.074      0.53      0.732     0.615      0.931    11
                 0.509     0.046      0.467     0.509     0.487      0.87     12
Weighted Avg.    0.599     0.038      0.606     0.599     0.589      0.913

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   <-- classified as
 489   0   1   2   0   0   2   4   0   0   0   2 |   a = 1
   0 410   0   0   0  11   0   0   0   0   0  18 |   b = 2
   0   0 257   2  27   6  24  30   1  43  33  10 |   c = 3
   6   9  23 186  10  48  30   6  15 150   9   8 |   d = 4
   3  10  30   0 455   3  13  17   0   8  69  22 |   e = 5
   0  20   4  22  16 284   2   2   1  12  11  69 |   f = 6
   9   8  46  10  77  32 168  27   7  28 127  68 |   g = 7
   5  11  61   5  86   3  69 163   0   9  68  14 |   h = 8
   0   0  24   3   1   0  36   3 297  66  22   1 |   i = 9
   3   1  53  10   9  16  17   3 107 232  20  10 |   j = 10
   1   0  24   0  22   1  34  20   6  26 455  33 |   k = 11
   7  26   6   3  51  37  11   2   5  23  44 223 |   l = 12
```

Fig 3. Naive base confusion matrix and data 40%

The figure above is showing the confusion matrix for 40 % data when we have given the whole data as training set. We have used Naïve Bayes algorithm and the machine learning tool Weka again to get this confusion matrix.

Below two images shows the Location 8 and 9 confusion matrix and data and location 9 and 10 confusion matrix and data as collected from the Naïve Bayes algorithm.

Classifier output

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0.006      0          0          0          ?        1
                 0         0.011      0          0          0          ?        2
                 0         0.109      0          0          0          ?        3
                 0         0.007      0          0          0          ?        4
                 0         0.084      0          0          0          ?        5
                 0         0.004      0          0          0          ?        6
                 0         0.126      0          0          0          ?        7
                 0.306     0.004      0.99       0.306      0.467      0.903    8
                 0.628     0.002      0.996      0.628      0.77       0.987    9
                 0         0.088      0          0          0          ?        10
                 0         0.09       0          0          0          ?        11
                 0         0.019      0          0          0          ?        12
Weighted Avg.    0.453     0.003      0.993      0.453      0.605      0.942

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   <-- classified as
   0   0   0   0   0   0   0   0   0   0   0   0 |   a = 1
   0   0   0   0   0   0   0   0   0   0   0   0 |   b = 2
   0   0   0   0   0   0   0   0   0   0   0   0 |   c = 3
   0   0   0   0   0   0   0   0   0   0   0   0 |   d = 4
   0   0   0   0   0   0   0   0   0   0   0   0 |   e = 5
   0   0   0   0   0   0   0   0   0   0   0   0 |   f = 6
   0   0   0   0   0   0   0   0   0   0   0   0 |   g = 7
  10  25 193  10 198   8 205 393   3  28 168  45 |   h = 8
   4   0  65   6   1   1  94   4 678 181  45   0 |   i = 9
   0   0   0   0   0   0   0   0   0   0   0   0 |   j = 10
   0   0   0   0   0   0   0   0   0   0   0   0 |   k = 11
   0   0   0   0   0   0   0   0   0   0   0   0 |   l = 12
```

Fig 4. Location 8 and 9 confusion matrix and data

Classifier output

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0.003      0          0          0          ?        1
                 0         0          0          0          0          ?        2
                 0         0.086      0          0          0          ?        3
                 0         0.015      0          0          0          ?        4
                 0         0.012      0          0          0          ?        5
                 0         0.021      0          0          0          ?        6
                 0         0.06       0          0          0          ?        7
                 0         0.004      0          0          0          ?        8
                 0.628     0.202      0.744      0.628      0.681      0.773    9
                 0.511     0.168      0.765      0.511      0.612      0.751    10
                 0         0.037      0          0          0          ?        11
                 0         0.009      0          0          0          ?        12
Weighted Avg.    0.568     0.185      0.755      0.568      0.646      0.761

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   <-- classified as
   0   0   0   0   0   0   0   0   0   0   0   0 |   a = 1
   0   0   0   0   0   0   0   0   0   0   0   0 |   b = 2
   0   0   0   0   0   0   0   0   0   0   0   0 |   c = 3
   0   0   0   0   0   0   0   0   0   0   0   0 |   d = 4
   0   0   0   0   0   0   0   0   0   0   0   0 |   e = 5
   0   0   0   0   0   0   0   0   0   0   0   0 |   f = 6
   0   0   0   0   0   0   0   0   0   0   0   0 |   g = 7
   0   0   0   0   0   0   0   0   0   0   0   0 |   h = 8
   4   0  65   6   1   1  94   4 678 181  45   0 |   i = 9
   3   1 126  27  25  46  39   6 233 588  38  19 |   j = 10
   0   0   0   0   0   0   0   0   0   0   0   0 |   k = 11
   0   0   0   0   0   0   0   0   0   0   0   0 |   l = 12
```

Fig 5. Location 9 and 10 confusion matrix and data

From the above figures we can clearly see that our intuition was correct and the algorithm is getting confused between the

points A and B (locations 9 and 10, i and j respectively in the confusion matrix) which are in the same room and is able to distinguish more clearly the points A and C (locations 9 and 8, i and h respectively in the confusion matrix) which are in two different rooms although at the same distance as the above two points. And this result was seen for all the three location sets we used in our experiments. So as result we can say that indeed the LTE signals can be used for distinguishing such points in the cases of indoor localization.

V. Plot showing Median and standard deviation of Precision vs % of Data

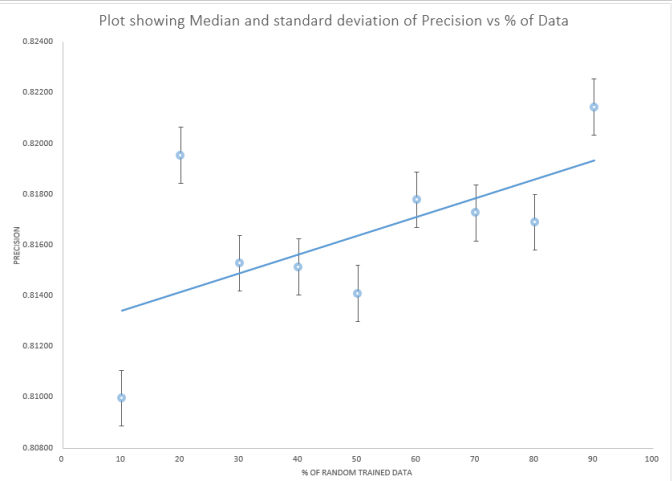

Fig 6. Plot showing Median and standard deviation of Precision vs % of Data

Plots that we got after collection of data from all the 12 locations over a period of 5 days in adverse climatic conditions such rainy, windy, night, foggy and normal. The result was not as good as we expected but when ran on the data for a particular day we get precisions in the range of 60's. But for all the 5 days the precision was in lower 30's. Here is the plot for the same.

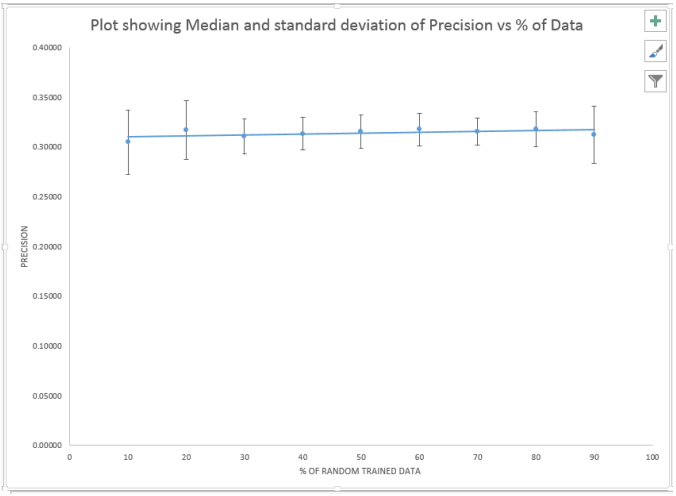

Fig 7. Plot showing Median and standard deviation of Precision vs % of Data for 5 days.

Below is given the confusion matrix for the first day data for 10% training data and also the confusion matrix for all the 5 days that we have collected.

```
=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
 1079    0    6    8    1    5    0    8    0    0    0    1 |    a = 1
    0  873    0    1    0   31    0    0    0    0    0   48 |    b = 2
    3    0  520    9   82   28   68   25    4  102   99    3 |    c = 3
   16   29   77  430   20  118   58    8   47  330   20    6 |    d = 4
    5   26   36    2 1143   19   11   39    0   15  155   43 |    e = 5
    0   72    4   48   52  656    5    8    1   28   32   91 |    f = 6
    8   23   97   15  235  115  288  128   20   78  304   56 |    g = 7
    9   24  169    7  190    7  132  385    4   34  150   36 |    h = 8
    0    0   79    4    2    1   38   12  582  230   31    0 |    i = 9
    2    1  119   21   30   55   16    3  216  545   22    2 |    j = 10
    2    0  101    0   75   18   33   68   16   58 1042    3 |    k = 11
   12   78   18   16  174  184   25   13    8   37  115  325 |    l = 12
```

Fig 8. Confusion matrix for 10% training data for 1st day.

```
=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
 1627  665   23  115   24   75   93  185  132  175   83 1304 |    a = 1
  128  677  657  428   75  262  117  225 1344  289  125  186 |    b = 2
  136   47 1525  466  923  102   84  273  719   10   55  162 |    c = 3
   69  124  173 1376  611    5  291  461  833  122  204  197 |    d = 4
   19   21  258  445 2754    6   45  264  471  106   14  129 |    e = 5
  765  106  290  345 1066  417  339  538  396   29   42  172 |    f = 6
   76   88  239  924 1038  130  436  464  751   58   90  226 |    g = 7
   14   49  209  431 1469    7  218  527 1018  350   73  145 |    h = 8
  133   11    7  675  728    0    3  658 1820  231  120   98 |    i = 9
  105   53   97  571  454   29  161  339 1232 1089  260  116 |    j = 10
  202   52  281  549  574  190  197  236  600  263 1271   74 |    k = 11
   28   91  379  454 1073   22  123  516 1039  289  106  352 |    l = 12
```

Fig 9. Confusion matrix for 10% training data for 5 days.

Acknowledgment

We would like to express our gratitude towards Professor Samir Das of Computer Science Department, Stony Brook University, and Ayon Chakraborty under whose esteemed guidance we were able to successfully implement this project. They continually and persuasively pushed us towards research. We would also like to thank Wings Lab of Computer Science Department of Stony Brook University, for providing us the resources and access to the computer labs required for the project.

References

[1] http://en.wikipedia.org/wiki/LTE_%28telecommunication%29.

[2] http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5467268.

[3] http://sdr.osmocom.org/trac/wiki/rtl-sdr.

[4] http://www.evrytania.com/lte-tools/lte-tracker.

[5] http://www.cs.waikato.ac.nz/~ml/weka/.

[6] http://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing.

[7] http://en.wikipedia.org/wiki/Naive_Bayes_classifier