Localization within Bharti building using WiFi

CSL838 Class Project: Progress Report-II

Naveen Kumar Tiwari
2013MCS2566
mcs132566@iitd.ac.in

Ravi Kamal Choudhary
2013MCS2576
mcs132576@iitd.ac.in

Ujjwal Kumar Gupta
2013MCS2588
mcs132588@iitd.ac.in

May 8, 2015

1 Objectives

Increasing availability of Wireless network infrastructure fueled finding interest in the applications that uses it for solving problems in other fields. Some of these may be activity recognition, survillence, security fields where it is needed to find location of Personal Digital Assistants (PDA) like mobile phones, tablets, laptops in an indoor environment.

Most of the work done to perform localization of devices is based on making use of Received Signal Strength (RSS). However, these techniques face challanges due to changes in environment (obstacles, persons, walls etc.) and various propagation effects such as reflection of signals against walls, ceiling and floor, diffraction and scattering. These problems cause signal variation and interference in the received signals. This motivated us to design and implement a solution based on machine learning that takes these problems into consideration and provide an effective and efficient solution.

In order to approach our goal, we will use provided Bharti Building Wireless Network infrastructure to locate a laptop on the 4th floor of building. To develop solution with advantages of machine learning, we will first collect data which will be RSS in this case, at different positions on the 4th floor using laptop. In next step, a model will be generated on data that will help us to locate position of the person entering 4th floor, running our application in his laptop and providing RSS to our algorithm. On giving RSS, our algorithm will be able to predict his location.

2 Hardware Requirements

1. **Access points of Bharti Building Wireless Infrastructure** : Used for collecting data and then for localization.
2. **Laptop** : Used for measuring RSS at different positions as data on 4th floor of Bharti Building and testing the solution.

3 Softwares Requirements

1. **Matlab** : Used for analysing data and implement algorithm
2. **Java/C++** : Used for developing front end of the application.

4 Project Milestones

The project will be done in the following phases :

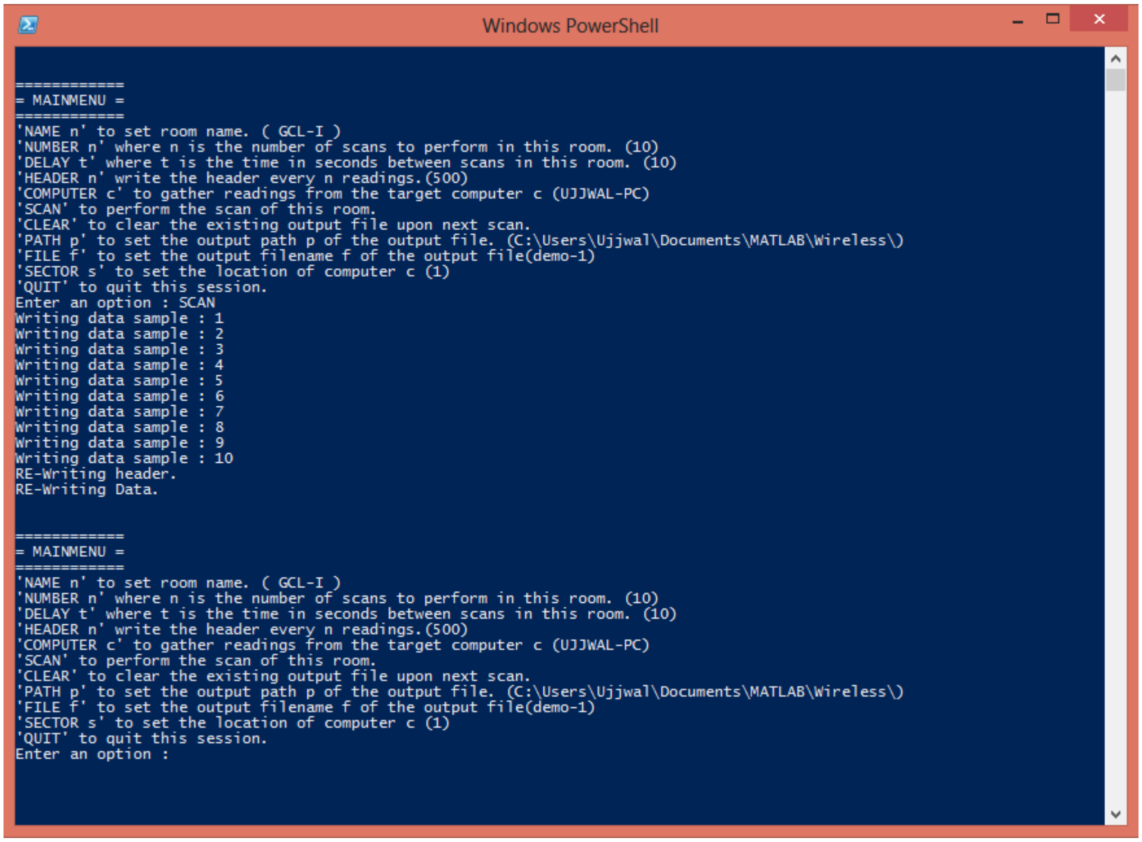
1. **Phase 1** : Data collection (till February 6, 2015).
2. **Phase 2** : Data Analysis and implementation of machine learning based algorithm (till March 10, 2015).
3. **Phase 3** : Interfacing of algorithm with the front end of application (till March 30, 2015).
4. **Phase 4** : Further refinement in the built system, conclusions and possible future works (till April 25, 2015).

5 Data Collection and Data Analysis

In the first phase of this project data collection and data analysis is done. For this purpose, we used existing Bharti Building Wireless Network infrastructure. On doing investigation for access points, it is found that each floor of Bharati building has 2 access points with multiple network access cards installed on it. However, there were some different private wireless networks created by students too which are avoided while taking data.

5.1 Data Collection

For this purpose we developed windows powershell application which specifically reads various acess points signal quality readings at a particular sector and writes them to output file. Windows does not measure signal strength in terms of RSSI values but in terms of signal quality (percentage). However, It is possible to find a linear relationship between siganl quality value and corresponding RSSI value($quality = 2 * (dBm + 100)$ where dBm : $[-100$ to $-50]$ and $dBm = (quality/2) - 100$ where $quality$: $[0$ to $100]$)[1].The interface for data collection is as given in Figure-1.



```
============
= MAINMENU =
============
'NAME n' to set room name. ( GCL-I )
'NUMBER n' where n is the number of scans to perform in this room. (10)
'DELAY t' where t is the time in seconds between scans in this room. (10)
'HEADER n' write the header every n readings.(500)
'COMPUTER c' to gather readings from the target computer c (UJJWAL-PC)
'SCAN' to perform the scan of this room.
'CLEAR' to clear the existing output file upon next scan.
'PATH p' to set the output path p of the output file. (C:\Users\Ujjwal\Documents\MATLAB\Wireless\)
'FILE f' to set the output filename f of the output file(demo-1)
'SECTOR s' to set the location of computer c (1)
'QUIT' to quit this session.
Enter an option : SCAN
Writing data sample : 1
Writing data sample : 2
Writing data sample : 3
Writing data sample : 4
Writing data sample : 5
Writing data sample : 6
Writing data sample : 7
Writing data sample : 8
Writing data sample : 9
Writing data sample : 10
RE-Writing header.
RE-Writing Data.


============
= MAINMENU =
============
'NAME n' to set room name. ( GCL-I )
'NUMBER n' where n is the number of scans to perform in this room. (10)
'DELAY t' where t is the time in seconds between scans in this room. (10)
'HEADER n' write the header every n readings.(500)
'COMPUTER c' to gather readings from the target computer c (UJJWAL-PC)
'SCAN' to perform the scan of this room.
'CLEAR' to clear the existing output file upon next scan.
'PATH p' to set the output path p of the output file. (C:\Users\Ujjwal\Documents\MATLAB\Wireless\)
'FILE f' to set the output filename f of the output file(demo-1)
'SECTOR s' to set the location of computer c (1)
'QUIT' to quit this session.
Enter an option :
```

Figure 1: Screenshot of Data Collection Interface in Windows Powershell

We aimed General Computing Labs (GCLs) and 4th floor corridor for this project. These locations are chosen because GCL-I and GCL-II have similar environments and we can test them differently by dividing them in different sized small areas. We collected signal quality values from each GCL labs and 4th floor corridor by manually dividing each one of them into small areas having almost similar shape called sectors.

1. **Testsite-1 (GCL-I):**

   We divided GCL-I lab into 8 different sectors as per demostrated in Figure-2. We took signal quality readings from different access points installed by IIT Delhi in Bharati building. For each sector 10 different readings were taken with 10 seconds gap in each readings and then mean is taken.

2. **Testsite-2 (GCL-II):**

   We divided GCL-II lab into 20 different sectors as per demostrated in Figure-3. We took signal quality readings from different access points installed by IIT Delhi in

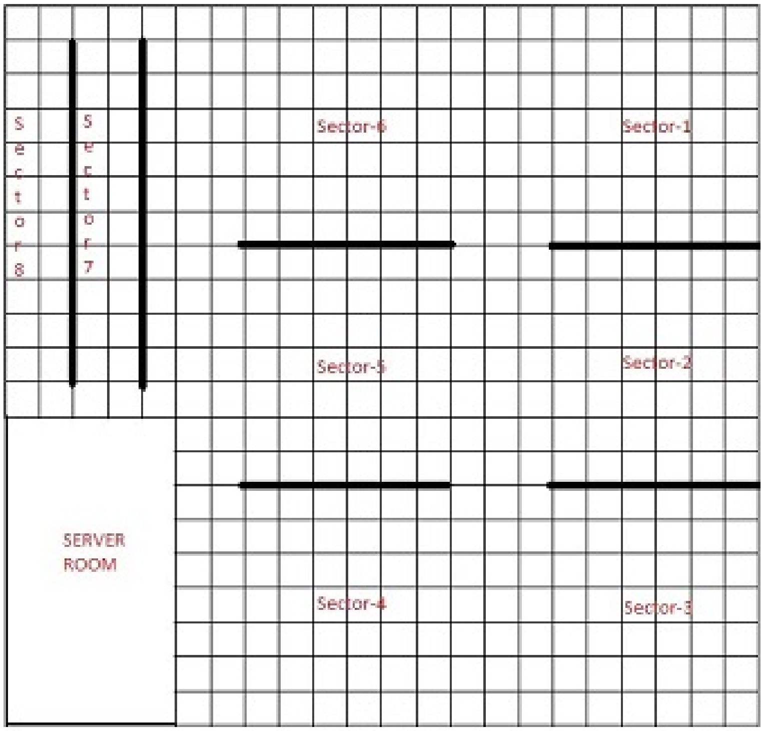


Figure 2: Map: GCL-I, Bharti Building

Bharati building. For each sector 10 different readings were taken with 10 seconds gap in each readings and then mean is taken.

3. **Testsite-3 (4th floor corridor):**

   We divided corridor into 10 different sectors including lift area as per demostrated in Figure-3. We took signal quality readings from different access points installed by IIT Delhi in Bharati building. For each sector 10 different readings were taken with 10 seconds gap in each readings and then mean is taken.

5.2 Data Analysis

We analysed data collected from each test sites by plotting a 3-D graph by keeping each access point as axes. It is easy to see in graphs that for each sector, readings make a cluster. It can be supported by saying each access point has almost fixed distance from each sector. However, some noisy data points are also seen in graphs because of environmental noises in Wifi while collecting data.

Here, each data number x represents corresponding data of sector x.

We have faced some issues with different hardware configuration (laptops), which we have normalized before consideration. These are listed below:

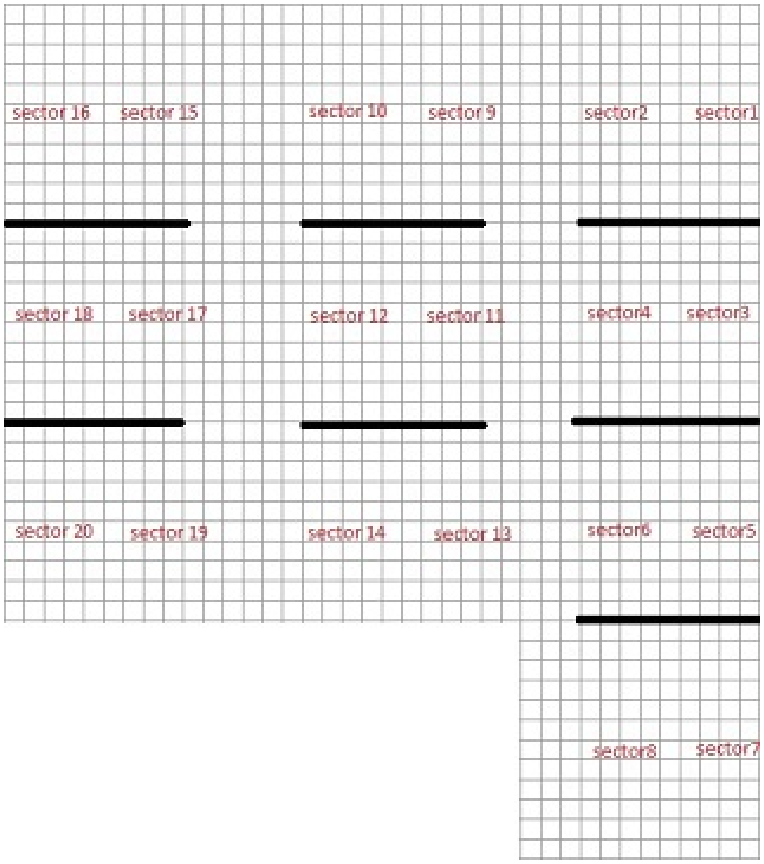


Figure 3: Map: GCL-II, Bharti Building

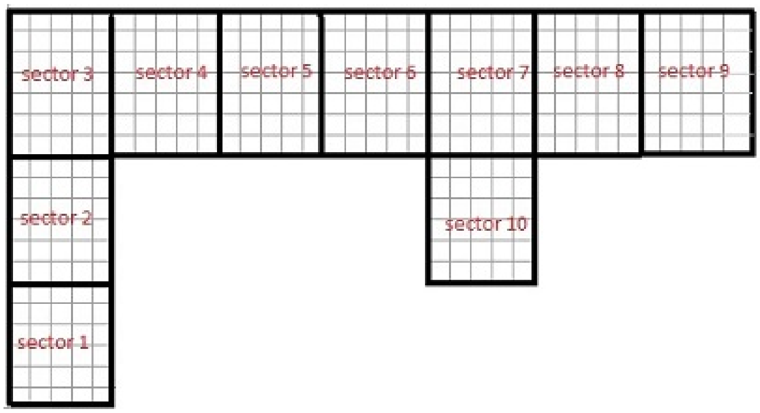


Figure 4: Map: 4th floor corridor

5.2.1 Temporal Variation

While taking readings, we have observed that a little change in the surrounding's environment causes RSS signal strength to vary in a considerable amount around 10-15 dBm in

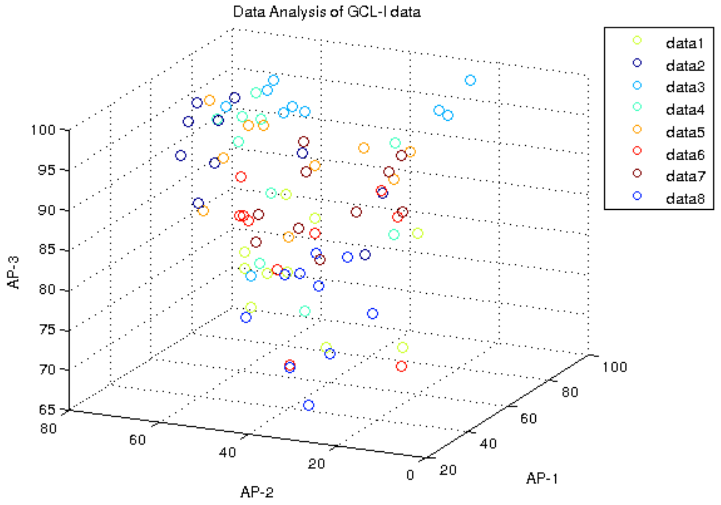


Figure 5: 3D-graph plot for GCL-I

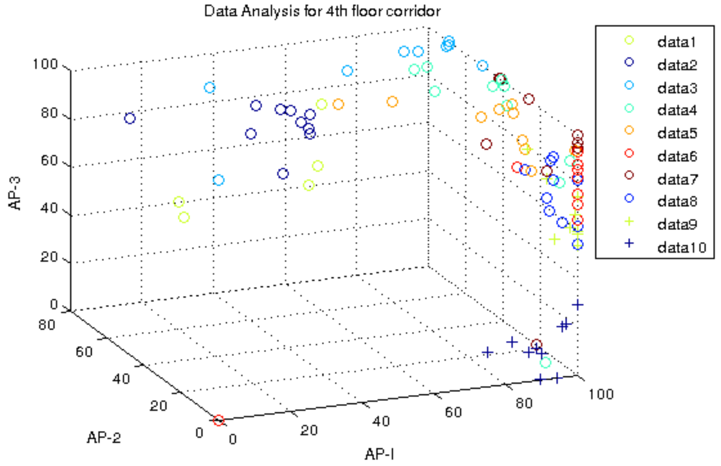


Figure 6: 3D-graph plot for 4th floor corridor

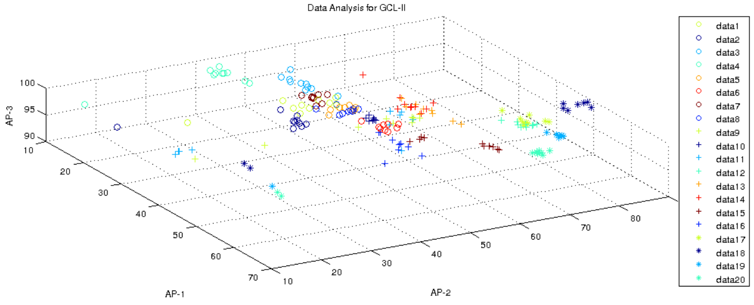


Figure 7: 3D-graph plot for GCL-II

seconds at a particular location. We have also observed that difference in the set of RSS values recorded at one location with different orientations of laptop is sometimes quite large.

5.2.2 Variation

We have also observed that the RSS vector (a vector of RSS values of all the AP's) at a point at time t1 in one sector is equal to a different RSS vector at another at time t2 in some other sector. This is the observed spatial variation in the collected RSS values. This is probably due to change temperature, humidity, obstacles and other physical causes. This creates ambiguity in data that is very hard to learn by ML algorithms and may leads to significant errors.

6 Implementation: Machine Learning Algorithm

We've implemented various machine learning algorithms based on their respective accuracies. For evaluation of these methods, we use accuracy as parameter. For this we first trained the model with our database which we created from the readings of sectors then we tested it against test data which was unseen by model and reported accuracy as testing accuracy. Training accuracy is calculated by classifying training data using model.

6.1 Quadratic Discriminative Analysis

Quadratic classifiers are more general version of linear classifiers. In Linear Discriminative Analysis it is assumed that the measurements from each class are normally distributed, whereas in QDA there is no assumption of covariance of each class being identical. When the normality assumption is true, the best possible test for the hypothesis that a given measurement is from a given class is the likelihood ratio test. Let us assume there are two groups ($y \in \{0,1\}$) and means are μ_0, μ_1 respectively and covariances are defined as $\Sigma_{y=0}$ and $\Sigma_{y=1}$ respectively. Then the like hood ratio for some threshold t, given by

$$\frac{\sqrt{2\Pi|\Sigma_{y=0}}exp(-\frac{1}{2}(x-\mu_1^T)\Sigma_{y=1}^{-1}(x-\mu_1))}{\sqrt{2\Pi|\Sigma_{y=1}}exp(-\frac{1}{2}(x-\mu_0^T)\Sigma_{y=0}^{-1}(x-\mu_0))} < t \tag{1}$$

After some rearrangement, it can be shown that the resulting separating surface between the classes is a quadratic.

Location	Training Accuracy	Testing Accuracy
Testsite 1	81.25	70.7
Testsite 2	25.5	14.3
Testsite 3	47	36.8

6.2 Decision Tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. It is a flowchart-like structure in which each internal node represents a ”test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represents classification rules. This type of analysis involves a tree and a closely related influence diagram as decision support tool.

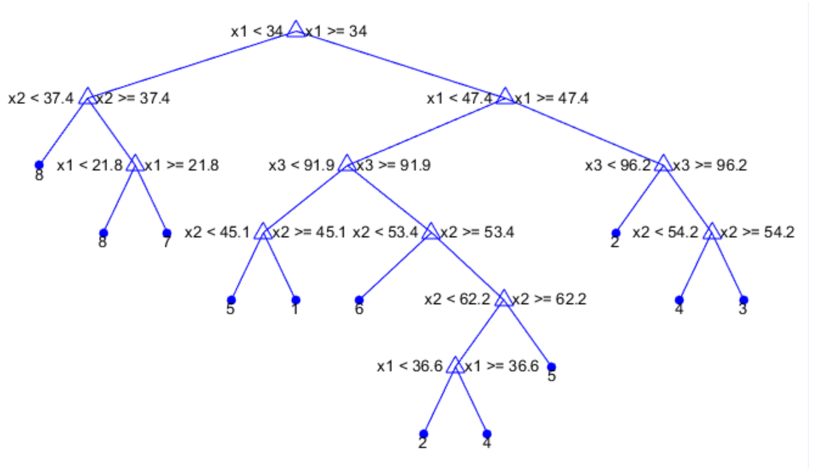


Figure 8: Decision Tree GCL-I

In above figure, x1, x2 and x3 are three different access points signal qualities. Leaf nodes are the predicted sectors and internal nodes are conditions on which decisions are made.

Location	**Training Accuracy**	**Testing Accuracy**
Testsite 1	86.5	72.2
Testsite 2	59.5	46.3
Testsite 3	83	70.6

6.3 Support Vector Machine

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM [6] training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. So it is a linear classifier where margin (with that can be increased without hitting data point) is maximum. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

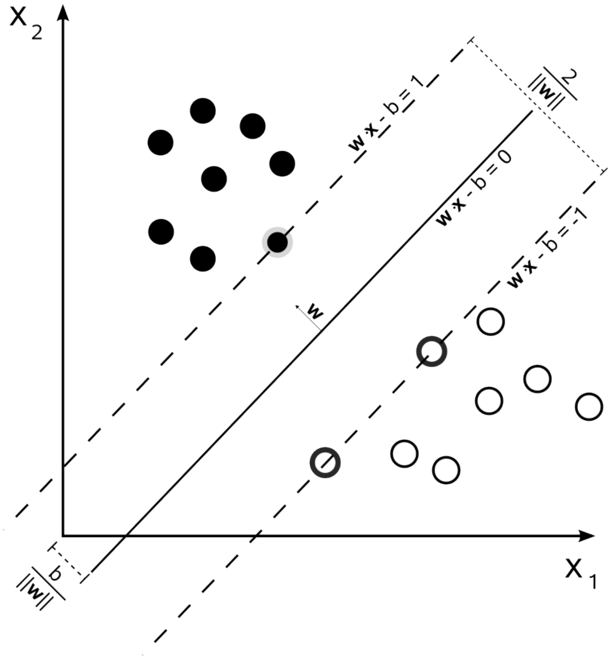


Figure 9: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors[6]

SVM Formulation

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}||w||^2 + C\,\Sigma_{i=1}^{n}\xi_i, \qquad (2) \\ \text{s.t. } \forall i, \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \\ \forall i, \quad & \xi_i \geq 0 \end{aligned}$$

Where ξ is penalty vector, w is a weight vector, y is labelling vector, x is input vector, C is cost and b is bias.

Location	Training Accuracy	Testing Accuracy
Testsite 1	97.5	86.4
Testsite 2	89	78.25
Testsite 3	99	85.3

6.4 k-mean Clustering

k-mean clustering is an algorithm of vector quantization. This method try to make the k different cluster with a mean, which serves as the prototype of the cluster. Once cluster have been formed, all the test points are mapped to different clusters based on the distance to different points of the cluster. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum

Given a set of observations $(x_1, x_2, , x_n)$, where each observation is a d-dimensional real vector, k-means clustering [7] aims to partition the n observations into $k(\leq n)$ sets $S = S_1, S_2, , S_k$ so as to minimize the within-cluster sum of squares (WCSS). In other words, its objective is to find:

$$\underset{s}{\arg\min} \quad \Sigma_{i=1}^{k} \Sigma_{x \in S_i} ||x - \mu_i||^2 \tag{3}$$

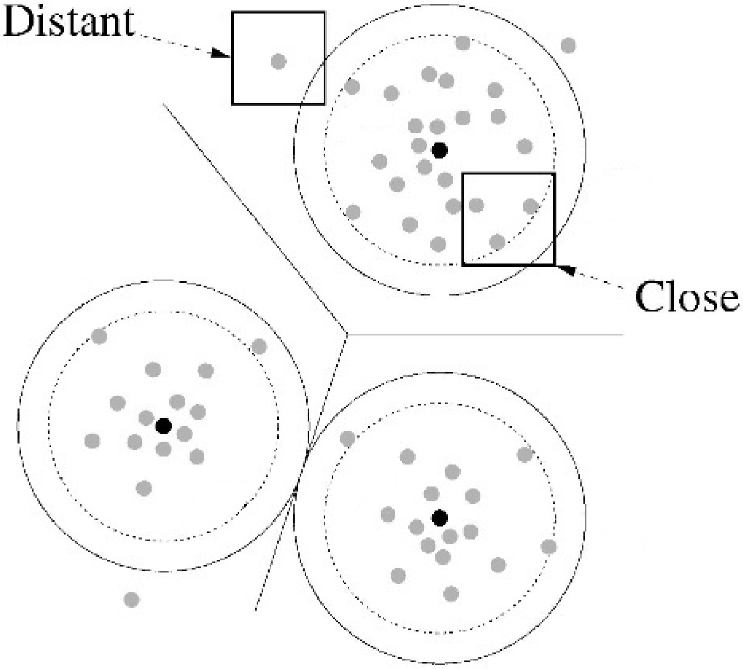


Figure 10: 3 clusters separated by the planes, depicting various points falling in each of them

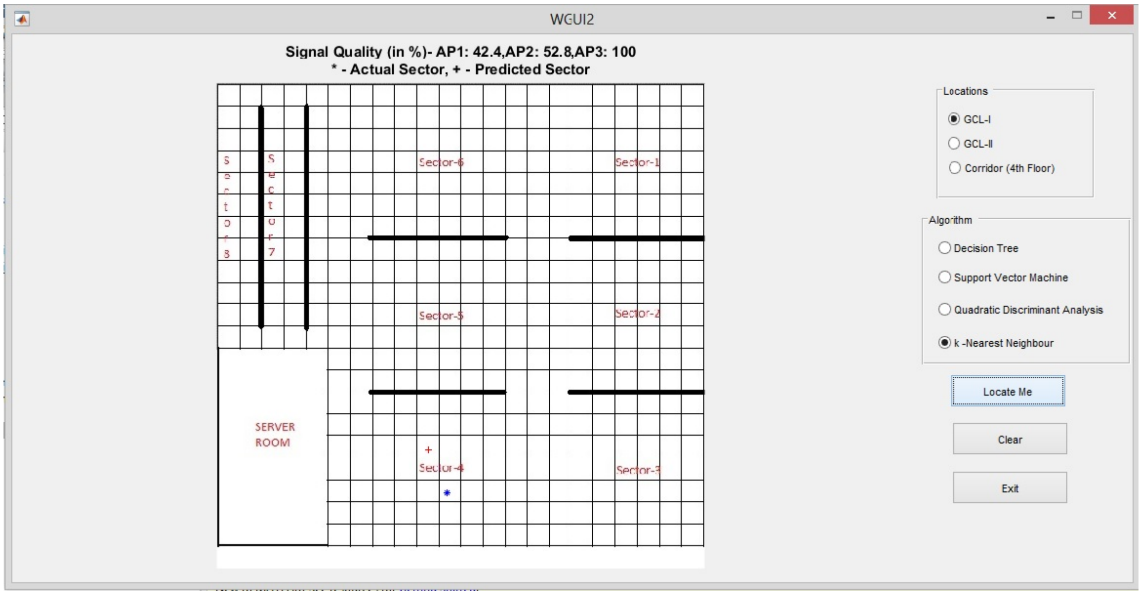
Location	**Training Accuracy**	**Testing Accuracy**
Testsite 1	98.1	89.4
Testsite 2	93	82.5
Testsite 3	99	90.3

7 Observations

- We find out that testsite-1 and testsite-2 having almost same environment but different accuracies because we choose to have small sectors in testsite-2 to make observations in dense but small environment.
- Testsite-1 shows better accuracy than others as there less interference in signals are recorded.
- From testsite-1 and testsite-2 results it can be infer that machine learning methods may work for precision of < 5 meters which is better than GPS navigation system (10-15 m).
- Accuracy can further be improved if number of access points increased as this will increase more parameters to classify the location.
- Usually SVM works better than the QDA and k-mean clustering, but we observed that this is true for more open sites such as corridor, while k mean clustering seems better for congested places. This is similar to various method of localization such as GPS, and other related apps in android.
- Finally, we concluded based on our experiments and result, SVM is better for open spaces while k-mean clustering is suited for congested sectors.

8 User Interface

We have designed a matlab based UI considering data is already collected with proper file name in the same folder. Currently, we have test data for 3 sites, which are marked as radio button in the user interface. We have implemented 4 algorithms viz. Decision Tree, SVM, Quadratic Discriminative Analysis and k-mean clustering which are shown as radio button for user to select. After selecting the test site, user has to choose an algorithm to predict the sector. For better look and feel, we've shown test map where predicted sector will have a dot similar to GPS. Also, and actual sector will have some different dot, which is actually acquired from the test file(No significance in prediction, just to show and compare the accuracy). Currently we have around 90% accuracy with SVM and k-mean clustering. The interface is shown below in figure 11.



Signal Quality (in %)- AP1: 42.4,AP2: 52.8,AP3: 100
* - Actual Sector, + - Predicted Sector

Locations
● GCL-I
○ GCL-II
○ Corridor (4th Floor)

Algorithm
○ Decision Tree
○ Support Vector Machine
○ Quadratic Discriminant Analysis
● k -Nearest Neighbour

Figure 11: User Interface

9 Conclusion

Collected RSS data shows temp oral and spatial variation, which is inherent to wifi localization problem, for which most of the algorithm shows error up to some extant. However, we've find out that SVM and k-mean clustering better than QDA and decision tree. For the real environment we recommend SVM and k-mean clustering. We've also observed that using same device for data collection and localization yields better result.

10 Future Works

- This can be further extended to incorporate 3D map, where we might be able to tell the depth information also.
- This can be further extended to include dynamism which will have same look and feel as GPS, where user's location varies on the map along with his/her movements

References

[1] https://msdn.microsoft.com/en-us/library/windows/desktop/ms706828

[2] http://www.support-vector-machines.org/

[3] P. Bahl, V. N. Padmanabhan. RADAR: An Inbuilding RF-based User Location and Tracking System. In INFOCOM, 2000.

[4] Siddiqi, Sajid, Gaurav S. Sukhatme, and Andrew Howard. ”Experiments in monte-carlo localization using wifi signal strength.” The 11th International conference on advanced robotics, Coimbra, Portugal. 2003.

[5] Biswas, Joydeep, and Manuela Veloso. ”Wifi localization and navigation for autonomous indoor mobile robots.” Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010.

[6] Cortes, Corinna, and Vladimir Vapnik. ”Support-vector networks.” Machine learning 20.3 (1995): 273-297.

[7] Tapas Kanungo, Nathan S. Netanyahu, Angela Y. Wu, ”An Efficient k-Means Clustering Algorithm: Analysis and Implementation”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 7, JULY 2002

[8] Quinlan, J. Ross. ”Induction of decision trees.” Machine learning 1.1 (1986): 81-106.

[9] Cover, Thomas M. ”Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition.” Electronic Computers, IEEE Transactions on 3 (1965): 326-334.