Predicting Location via a Statistical Indoor Positioning System

Project Group 38

Names: YILONG LI(SID: 25772051), CHING MAN HAZEL MAK(SID: 25765738), ZHENG CHANG

Introduction

As the technology developes, Wireless is one of the most important tools in our life. In this project, we will build a model to predict location to develop indoor positioning systems (IPS). IPS is a solution to locate people inside a building by using signal strength.

In order to predict the location, we need a banch of data where the signal strength between hand-held device such as labtop, routers, and cell phone are measured in the building. Then, we build a model to examine the signals strength within a building at the University of Mannheim. Based on the model, we can predict location with these data to develop a statistical IPS .

Background

We use k-nearest neighbors to predict location (x,y), and then use cross-validation to find a best k. In order to measure distance, we come up Euclidean distance to estimate the distance between two sets of signal strengths. Then we find k closest training points , and then we estimate new observation's position by an aggregate of the (x, y) positions of the k training points. Then, cross-validation can help us to determine the best k. Cross-validation is a statistical model to evaluate by dividng two grops; one is using train a model and other used to valid the model. The technique is assessing how the outputs generalize an independent data set. Finally, after we find a best K from offline.txt, we apply the data to online data, which is true data. Then we assessw how the best k fits with the true data.

Processsing and Cleaning Data to Build a Representation for Analysis

Task 1

First, we read offline.final.trace.txt into R. We found that the data is extremely big.

```
## [1] "# timestamp=2006-02-11 08:31:58"
## [2] "# usec=250"
## [3] "# minReadings=110"
## [4] "t=1139643118358;id=00:02:2D:21:0F:33;pos=0.0,0.0,0.0;degree=0.0;00:14:bf:b1:97:8a=-38,2437000000,3;00:14:bf:b1:97:90=-56,2427000000,3;00:
## [5] "t=1139643118744;id=00:02:2D:21:0F:33;pos=0.0,0.0,0.0;degree=0.0;00:14:bf:b1:97:8a=-38,2437000000,3;00:0f:a3:39:e1:c0=-54,2462000000,3;00:
## [6] "t=1139643119002;id=00:02:2D:21:0F:33;pos=0.0,0.0,0.0;degree=0.0;00:14:bf:b1:97:8a=-38,2437000000,3;00:0f:a3:39:e1:c0=-54,2462000000,3;00:
```

When we executed head(txt) function, we found that some characters are not necessary. That is "#" character. Therefore, We discard the sentences starting with the comment character "#". We also found that the variable of the â€œnameâ€ is separated by an "=" character, some values that contain multiple values are seperated by "," , and also some variables are seperated by ";" character. Therefore, We use regular expression to process the raw data.

In the function of processline of MAC1, the elements, except the first 10, give the information of the signals received at the same time, same postion and the same hand- held device. We extract these variables and create a 4 columns matrix, indicting each particular mac address, signal value, channel and type for each signal that detected. The number of signals that detected will then be the number of rows. For MAC2, we only extract the mac address, physical coordinates and the time for the scanning point and create a matrix in the first 10 elements. These information are indicated by 6 variables, so we will have 6 columns.

Finally, We combine these information with MAT1 and create and a matrix, so that the the new matrix will give information for each signal detected from the same scannning point.

Task 2

1. We found that the type of time, posX, posY, posZ, orientation and signal varable is all character, also they are all numbers. It is a good idea converting them to numeric because it will help us to analyze data in next section. Therefore, We converted them all into the type of numeric.

2. According to the data, we found that the variable of time is measured by milliseconds. It was not readable if we want to see the pattern of some variables, for example, two macs was recording at the same time and location. Therefore, we frist converted milliseconds to seconds by dividing 100, and then, we used POSIXt, POSIXct to convert seconds to readable dates plus time, e.g. 2006-02-10 23:31:58

3. Accounding to documentation, we need to drop all records that correspond to adhoc devices, and not the access points. We also found the all of posZ are zero ,and scanMac has only one value. Therefore, we discarded these 2. Right now, we only keep "time","posX", "posY", "posZ", "orientation", "signal" as variables.

4. Accounding to documentation, we also need to round the values for orientation to the nearest 45 degrees, but keep the original values too.

5. After we narrowed down the Mac numbers(by unique function), we found 12 Mac numbers. Rereading the documentation, we need only 6 Mac numbers, so we looked up the MAC addresses at http://coffer.com/mac find/, and we only found 5 MAC addresses which are all Linksys/Cisco's, there was no Mac numbers matching with Lancom L-54g routers. Therefore, we had to find the sixth mac number.

12 Mac numbers

First, we check the 12 Mac numbers by listing a table.

00:04:0e:5c:23:fc : 418

00:0f:a3:39:dd:cd : 14145619

00:0f:a3:39:e0:4b : 43508

00:0f:a3:39:e1:c0 : 145862

00:0f:a3:39:e2:10 : 19162

00:14:bf:3b:c7:c6 : 126529

00:14:bf:b1:97:81 : 120339

00:14:bf:b1:97:8a : 132962

00:14:bf:b1:97:8d : 121325

00:14:bf:b1:97:90 : 122315

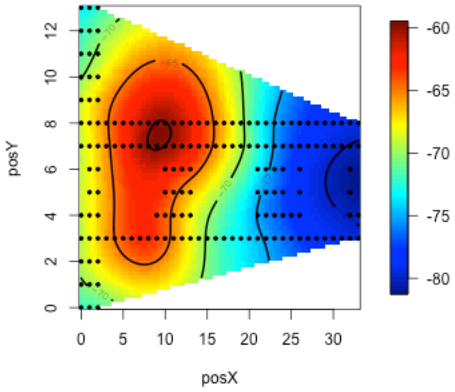
00:30:bd:f8:7f:c5 : 301

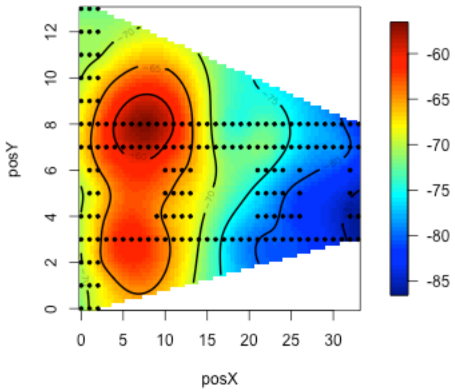
00:e0:63:82:8b:a9 : 103

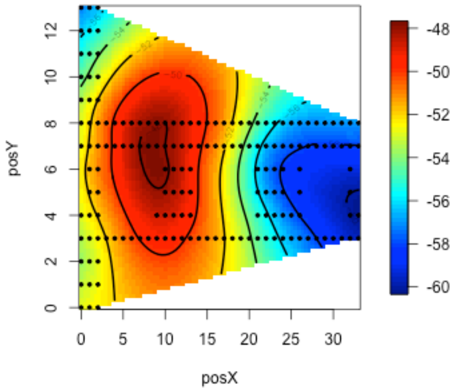
According to the table, there are 12 mac addresses. However, some mac addresses counts are extremely low since there are signals recorded at 166 grid points, 110 replications, and 8 orientations. Therefore, the count of the each addresses should be around 166 * 110 * 8, which is 146080. Hence, we will select "00:0f:a3:39:e1:c0", "00:0f:a3:39:dd:cd", "00:14:bf:b1:97:8a", "00:14:bf:3b:c7:c6", "00:14:bf:b1:97:90","00:14:bf:b1:97:8d", "00:14:bf:b1:97:81" as macAddress. But, it still have 7 Mac numbers there. We only need 6!!

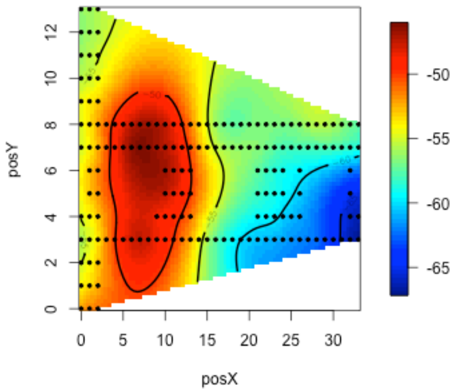
Mapping 7 Mac numbers

After we selected 7 Mac numbers, we want to know where the 7 macs located on. So we were using the heat plot to map all 7 access points. We found that there are 2 access points at almost the same location. (see 4 plots below)









The first two plots are 00:0f:a3:39:dd:cd with angle degree 0 and 90, The second two plots are 00:0f:a3:39:e1:c0 with angle degree0 and 90. The dark red color is the location of access point. The location points are mapped also, which helped us to see every locations related to the floor plan.

Dropping 00:0f:a3:39:dd:cd

We want to know why they are at same location. So, we were tring to see how the time variable connect to the 2 access points. By doing this, we created a data frame with only 2 variable time and mac numbers.

```
##                       time               mac
## 7    2006-02-10 23:31:58 00:0f:a3:39:dd:cd
## 3    2006-02-10 23:31:58 00:0f:a3:39:e1:c0
## 698  2006-02-10 23:32:56 00:0f:a3:39:dd:cd
## 697  2006-02-10 23:32:56 00:0f:a3:39:e1:c0
## 1367 2006-02-10 23:34:25 00:0f:a3:39:dd:cd
## 1363 2006-02-10 23:34:25 00:0f:a3:39:e1:c0
```

From these 6 rows, it seems like 2 access points were recording at the same time. To prove that, we explored length of each mac and time(unique). We found these two are almost evenly separate. It may have some empty records or repeated records.

the length of 00:0f:a3:39:dd:cd is 1328

```
## [1] 1328    2
```

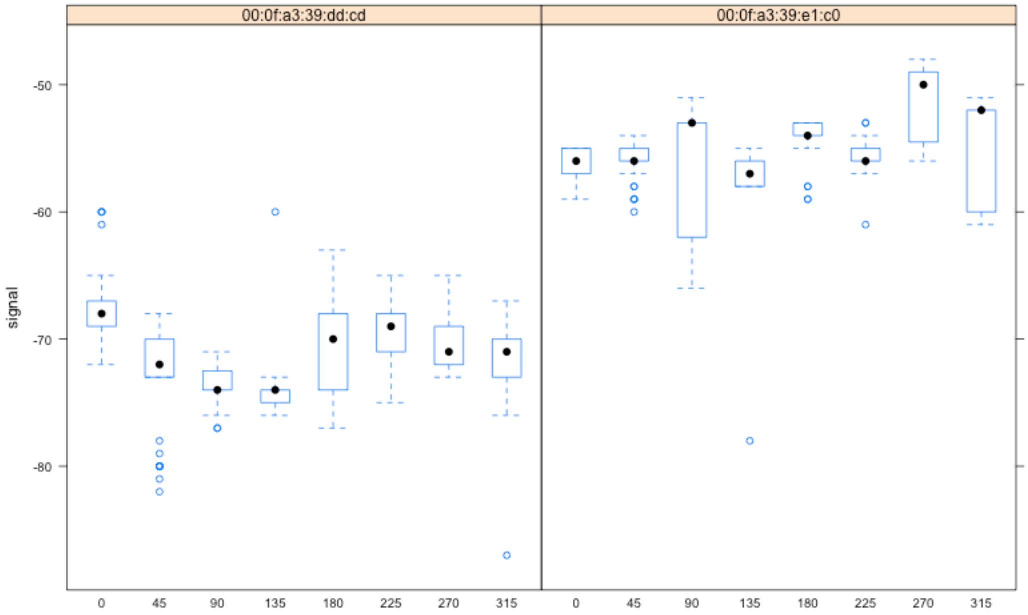
the length of 00:0f:a3:39:e1:c0 is 1328

```
## [1] 1328    2
```

the length of time is 1342

```
## [1] 1342
```

Right now, we known that there were 2 devices at same location. However, we still need to drop one to reach 6 Macs' goal. By doing this step, we need to find which device has a stable signals which will less affect our result in the next section. We created two signal strengths at a location where are far from this access point(2, 12)

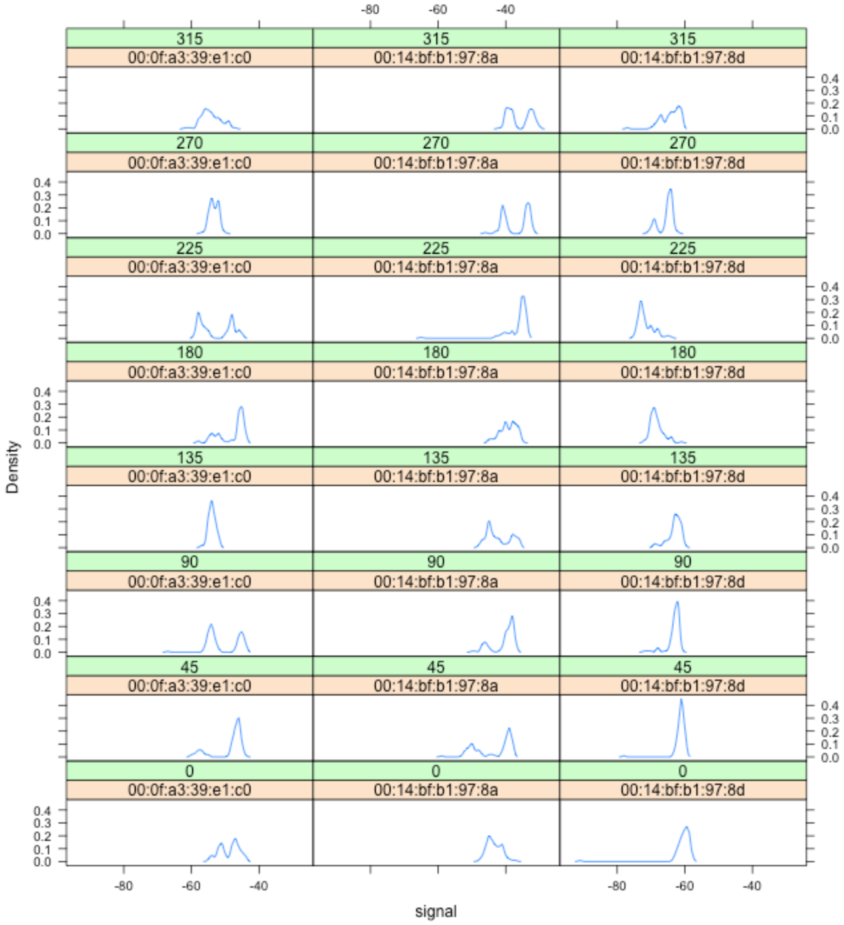


The two graphs show that the relationships between orientations and signal strength. We used this two graphs to choose which macAddress is stable. According to the graph, the macAddress "00:0f:a3:39:dd:cd" has lower mean of the signal. On the other hand, the "00:0f:a3:39:e1:c0" has higher mean of the signal. Therefore, the signal strength of "00:0f:a3:39:e1:c0" is stronger than "00:0f:a3:39:dd:cd" signal strength.

In addition, the variance of "00:0f:a3:39:dd:cd" is higher than the variance of "00:0f:a3:39:e1:c0". Hence, "00:0f:a3:39:e1:c0" is more stable than "00:0f:a3:39:dd:cd". Therefore, we drop "00:0f:a3:39:dd:cd".

Visualization of the Signal Strength Analysis

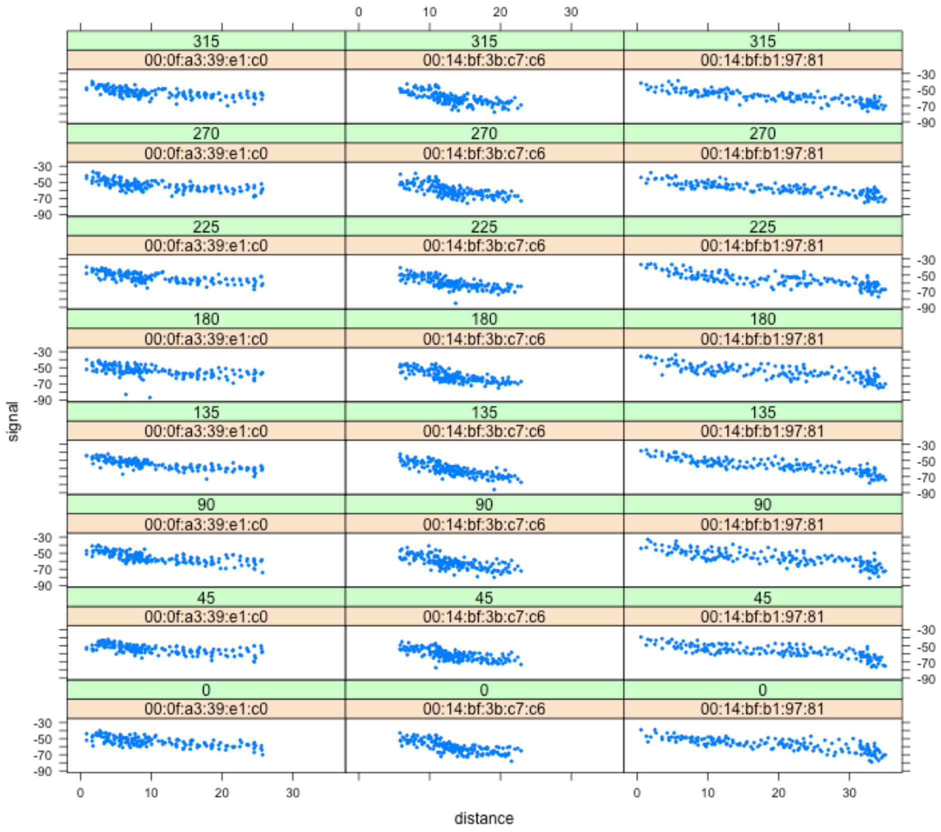
Signal Strength Distribution at some locations



We choose macAddress '00:0f:a3:39:e1:c0',"00:14:bf:b1:97:8a" and "00:14:bf:b1:97:8d" at position (2,2) to estimate the signal strength. The point(2,2) is close to "00:14:bf:b1:97:8a", and "00:14:bf:b1:97:8d" is far from the point(2,2). Therefore, we mark "00:14:bf:b1:97:8a" as close point, "00:0f:a3:39:e1:c0" as a middle point, "00:14:bf:b1:97:8d" as farrest point. Accoding to the graph, "00:14:bf:b1:97:8a" has the strongest sginal strength, and "00:14:bf:b1:97:8d" has the weakest signal strength. Therefore, it proofs that signal strength behaves diffrently at all locations. Far position has a weaker signal and closed position has a stronger signal strength.

Furthermore, the graph also showed that the orientation does not affect sginal strength significantly. Although each density curve is slightly different, the signal strength is almost same at each diffrent angle. Therefore, the level of the signal strength does not depend on the orientation, and it does depend on distance. The location and access point will affect the distribution, but the orientation does not affect the distribution significantly.

Signal Strength and the Distance



We chooseed "00:0f:a3:39:e1:c0", "00:14:bf:3b:c7:c6", and "00:14:bf:b1:97:81" macAddresses to estimates the signal strength. As we known, "00:0f:a3:39:e1:c0" and "00:14:bf:3b:c7:c6" are located at the inside of the room. On the other hand, "00:14:bf:b1:97:81" is located at the outside the room. According to the data, the three macAddresses have strong signal when the distance is close to zero. However, their sigals are weak as the the distance increasing, especially "00:0f:a3:39:e1:c0" and "00:14:bf:3b:c7:c6". When the distance is above 23, "00:14:bf:3b:c7:c6" does not have any signal record. 00:0f:a3:39:e1:c0" does not have signal for " when the distance is over 27. However, "00:14:bf:b1:97:81" has a less signals even though those are extremely low. Therefore, the wall of buildings affect the signal strength, and it shows that the wall of the building can add significant noise and screen to cut the signal strength measurements. In addition, human acitivity also affects the signal strength.

Furthermore, the graph shows that the relationship between the signal and the distance to the access point for each of the 3 access points and 8 orientations of the device. According to the graph, the shape of each signal strength is consistent across panels showing curvature in the relationship. Each sginal is very strong when the distance is close to zero. However, the each of the strength of the signal is decreasing as the distance is increasing. For those graph, although there are some signals when distance is more than 25, they are extremely weak, and they all tend to zero when the distance is close to 0. In conclusion, the level of strength signal does not depend on the orientation. It does depends on the distance and physical characteristics of a building and human activity. Therefore, the shape of signal strength is consistent.

Nearest Neighbor and Cross-validation Methods to Predict Location

Creating a function to predict the locations of the devices

We have the signal strengths from 6 macs address of all the 60 online data and we want to predict all the 60 locations. In this case, we created a function, which is called predXY(). This function has 4 parameters. NewSig is a data frame containing the signal strengths that is used for prediction, NewAngles is the vector showing the angles of the predicting positions. A data frame of offline's summary will be passed into the parameter trainData. And k is the number of closest positions we used for prediction.

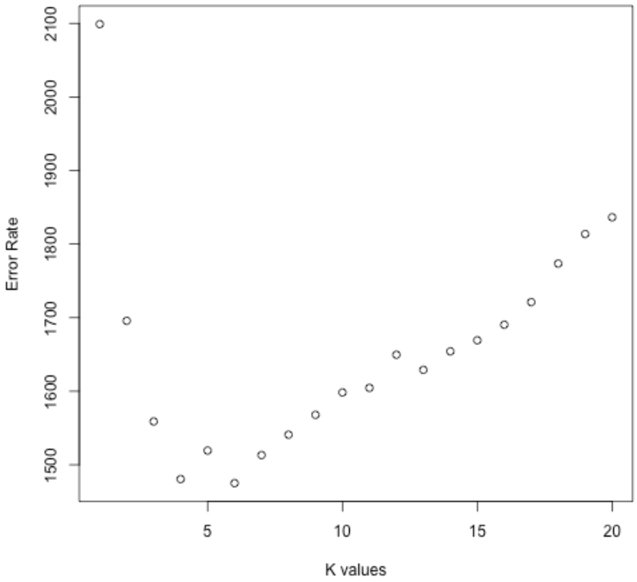
In this function, the "trainData" will be modified to data frames with signal strengths from 6 macs of all the positions on the same angle. The NewAngles will match the data of the same angles from the TrainData. That means, the new orientation with particular 6 signal strengths will match with the signal strengths of the training data on the same angle and produce the closest k positions of (X,Y). The mean of the values of X and Y will be the predicted positions.

Finding the best k and cross validation

To try this function and find out the best k, we compare the 166 known positions of the offline data and the predicted offline positions. We squared all the differences of actual X from predicted X and the differences of actual Y from predicted Y, then sum them up. This will be the error value of each prediction. But we want to know all the accuracy of k from 1:20 and find the best one. Before we use the cross- validation, we need to first reshape our offline data, so that it will have 6 columns of signal strengths from each position and the randomly selected angles. For cross- validation, we reordered the 166 positions randomly and put them into a matrix with 33 rows and 5 columns as we have chosen 5 as the number of folds. Then the 33 positions will be predicted first, and the information of other positions except that 33 positions will be used as the training data. In this case, we predicted all the locations using different values of k. And using the function for computing errors to calculate the error sum of each k. Observing the graph of the errors of different k values, we can see that the errors for the first four k values decrease sharply, and 6 is the k with the smallest error.

Processing and predicting the online data

After we have read in the online data, we used the functions we created in the previous parts to process the lines and clean the data. We realized that there are totally 60 different combinations of posX ad posY. Then we made a list of information showing the signals strengths between the 6 macs and each particular position. It was converted to a data frame, which was used for the prediction. With this data frame, the predXY() function, the training data and the best k that we have chosen, the online positions are then predicted.



Conclusions

Using 6 as the value of k to predict the positions in the test data(online data), we found out that the error is 348.1992, using the method we mentioned above. The error is quite low, comparing to using other k values, like k =1,2 or 3.