Unit 2 - Real Time Location System Case Study

Kelly Carter, Swee K Chew, Volodymyr Orlov, Anjli Solsi

Introduction

A real-time location system (RTLS) is one of several technologies used to identify and track the location of an object or person in real time. With global positioning systems (GPS) not performing accurately or reliably inside buildings, real-time location systems have been honed and specified for indoor usage. RTLS utilize continuous communication between a device and an object enabled by wireless local area networks (LANs) or indoor positioning systems (IPS) with WiFi signals from network access points. The growth and prevalence of WiFi has expanded the desire for these systems across all industries. The need and application of RTLS ranges from retail, healthcare, supply chain, logistics to numerous more industries to protect and track assets [1].

This type of system is developed based on the connection between a device and access points, measuring the signal strength and further details as the device moves around. For this case study, the RTLS data will be analyzed using the k-nearest neighbors (KNN) clustering method to determine locations and to determine potential issues with decisions made regarding the use, and non-use, of the data. The offline reference data is used as the training set to predict the location of the test/online data. Weighted and unweighted KNN algorithms are utilized to predict the device locations. Upon implementation of an alternative approach, the various predictions are compared and discussed.

In the analysis presented in the textbook, Nolan and Lang made the decision to keep the access point with MAC address 00:0f:a3:39:e1:c0 and to exclude the data corresponding to MAC address 00:0f:a3:39:dd:cd. For the purpose of this case study, we will be (1) conducting a more thorough analysis to determine which two of the MAC addresses that have the same access point should be used and whether using data for both MAC addresses results in more, or less, accurate prediction of location; and 2) implementing an alternative prediction method which applies weights on the received signal strength based on the distance.

Data

For this case study, the dataset is from the University of Mannheim and contains signal strengths that were measured using a handheld device on a specific floor in a building. This offline data is used as the training dataset to build the prediction model. The raw data contains the following fields, which will be transformed and cleaned to be suitable for analysis.

The *id* variable indicates the MAC address of the scanning device. The *degree* variable indicates the orientation of the user carrying the scanning device in unit degrees. The *time* variable is a timestamp in milliseconds since midnight of January 1, 1970, UTC. A new variable *raw time* is created to hold the value of the original time divided by 1,000. Then, it is converted into a usable time format of year-month-day and hour-minutes-seconds and assigned to the original *time* variable. The *pos* variable indicates the physical coordinates of the scanning device and originally has three values that represent the x, y, and z coordinates. The data is divided, and a column created for each coordinate as *pos X*, *pos Y*, and *pos Z*. The Z-coordinate is removed for the purpose of this analysis, based on the method of measurement collection. The end of the datapoint, *MACofResponse1* and *MACofResponseN* consists of the MAC address of a responding peer with the corresponding values for signal strength in dBm, channel frequency, and device mode of operation. These are parsed into respective columns and variables *mac*, *signal*, *channel*, and *type*. Multiple devices can respond to the scanning device, which is why the additional data is provided for the responding device. The *type* variable can either be a 3 for an access point or a 1 for a device in adhoc mode. For this analysis, only type 3 data points are kept for a focus on just access points.

Exploratory Data Analysis

To remove extraneous data points, all non-type 3 data points are removed. Then, orientation points are normalized, time variables are translated, and unused variables are removed to create the final data set that will be used for this analysis. The final format of this data set can be seen in Table 1.1.

time	posX	posY	orientation	mac	signal	channel	rawTime
2/11/2006 1:31	0	0	0	00:14:bf:b1:97:8a	-38	2437000000	1.10E+12
2/11/2006 1:31	0	0	0	00:14:bf:b1:97:90	-56	2427000000	1.10E+12
2/11/2006 1:31	0	0	0	00:0f:a3:39:e1:c0	-53	2462000000	1.10E+12
2/11/2006 1:31	0	0	0	00:14:bf:b1:97:8d	-65	2442000000	1.10E+12
2/11/2006 1:31	0	0	0	00:14:bf:b1:97:81	-65	2422000000	1.10E+12
2/11/2006 1:31	0	0	0	00:14:bf:3b:c7:c6	-66	2432000000	1.10E+12

Table 1.1: Data Frame Used for Analysis

As signal strength was recorded at 8 different orientations, it is expected that there should only be 8 unique orientations present in the data frame; however, 203 unique orientations are found. The cumulative distribution function (CDF) of orientation variable in Figure 1.1 shows that the values are clustered around the 8 expected angles. It should be noted that values near 0 or 360 refer to the same angle.

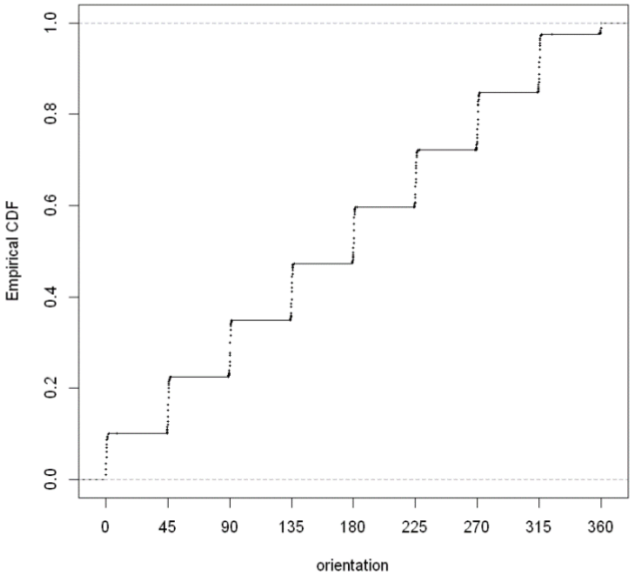

Figure 1.1: CDF or Orientation for the Hand-Held Device

To ensure only 8 unique orientations are represented in the data frame, we determine for each value what is the closest of the 8 orientations and round it to that value. Figure 1.2 shows the comparison of the original angle values to rounded values. This demonstrates the variability in the act of measuring (1).

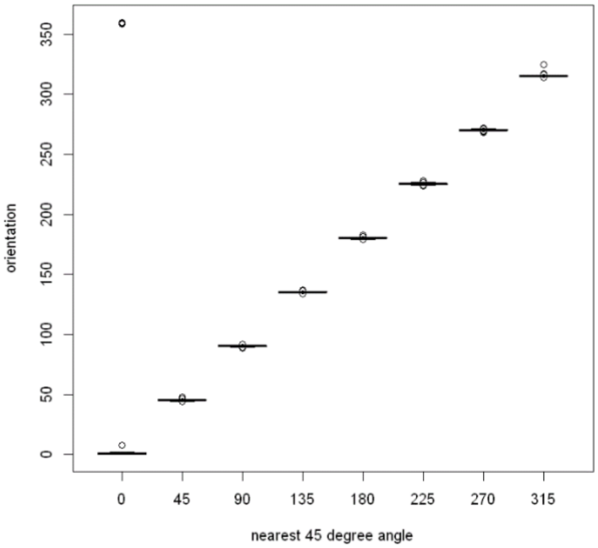

Figure 1.2: Boxplots of Orientations for the Hand-Held Drive

We observe that data frame has 12 unique MAC addresses and 8 unique channels which contradicts the assumed one-to-one relationship. Further analysis is required on the MAC addresses is required to determine why this result is obtained when the building plan only shows 6 access points. Table 1.2 shows the frequency of MAC addresses.

MAC Address	Count
00:04:0e:5c:23:fc	418
00:0f:a3:39:dd:cd	145,619
00:0f:a3:39:e0:4b	43,508
00:0f:a3:39:e1:c0	145,862
00:0f:a3:39:e2:10	19,162
00:14:bf:3b:c7:c6	126,529
00:14:bf:b1:97:81	120,339
00:14:bf:b1:97:8a	132,962
00:14:bf:b1:97:8d	121,325
00:14:bf:b1:97:90	122,315
00:30:bd:f8:7f:c5	301
00:e0:63:82:8b:a9	103

Table 1.2: Frequency of MAC Addresses

As there are 146,080 possible signal recordings as each MAC address location (a combination of 166 grid points, 8 orientations, and 110 replications), we need to remove any addresses that do not have sufficiently large counts. We see in Table 1.2 that 5 access points do not have sufficient counts. Further analysis confirms a one-to-one relationship between the remaining 7 MAC addresses and channels. Thus, the channel variable can be removed from the data frame.

Next, we explore the positioning of the hand-held device by examining the x and y coordinates (*posX* and *posY)*. We find there to be 476 unique (x,y) combinations of which 310 are empty. All empty positions are removed from the data frame resulting in a data frame the 166 unique locations where a hand-held device has been observed. Table 1.3 shows the first 8 locations along with their respective counts.

posX	0	1	2	0	1	2	0	1
posY	0	0	0	1	1	1	2	2
count	5505	5505	5506	5524	5543	5558	5503	5564

Table 1.3: Location Counts of First 8 (X,Y) Positions

We find that there are about 5,500 recordings at each of the first 8 locations. This is in line with the maximum possible signal readings of 6,160 (a combination of 8 orientations, 7 access points, and 110 replications). Figure 1.3 shows the signal count for all 166 unique observed location points is roughly 5,500.

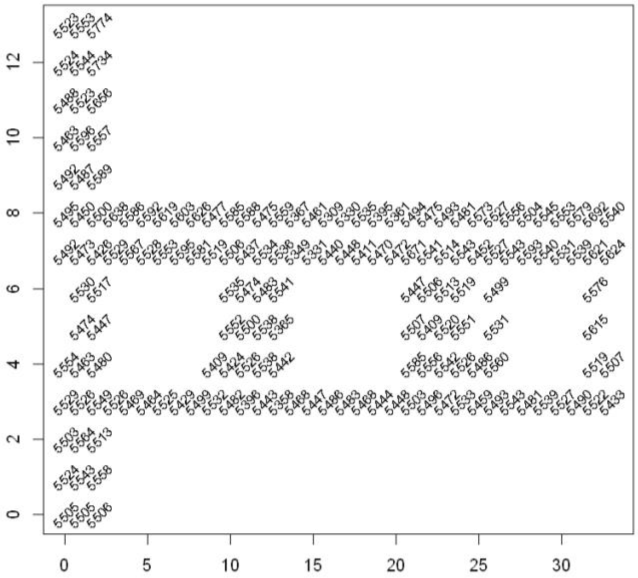

Figure 1.3: Signal Counts of Observed Location Points

Now that analysis on location, addresses, and orientation is completed, a final data set incorporating our findings is created and focus is turned toward our variable of interest, signal strength. Questions of interest that will guide further exploratory analysis include whether signal strength is affected by location, orientation, and access points and a look at the relationship signal strength and the distance between an access point and the hand-held device.

Signal strength readings have negative values where a higher value represents higher signal strength and a lower value represents weaker signal strength (1). The summary statistics of signal show the minimum signal strength is -98 and the maximum signal strength is -25. We additionally find the mean of signal strength is -60 and median is -59.

To answer our first question of interest we want to examine the distribution of signal strength at different orientations and access points. Using a fixed location point, we observe how signal strength changes at 6 of the access point changes for each of the 8 angles in Figure 1.4. For this figure, we observe that signal strength varies with orientation for both close and far access points.

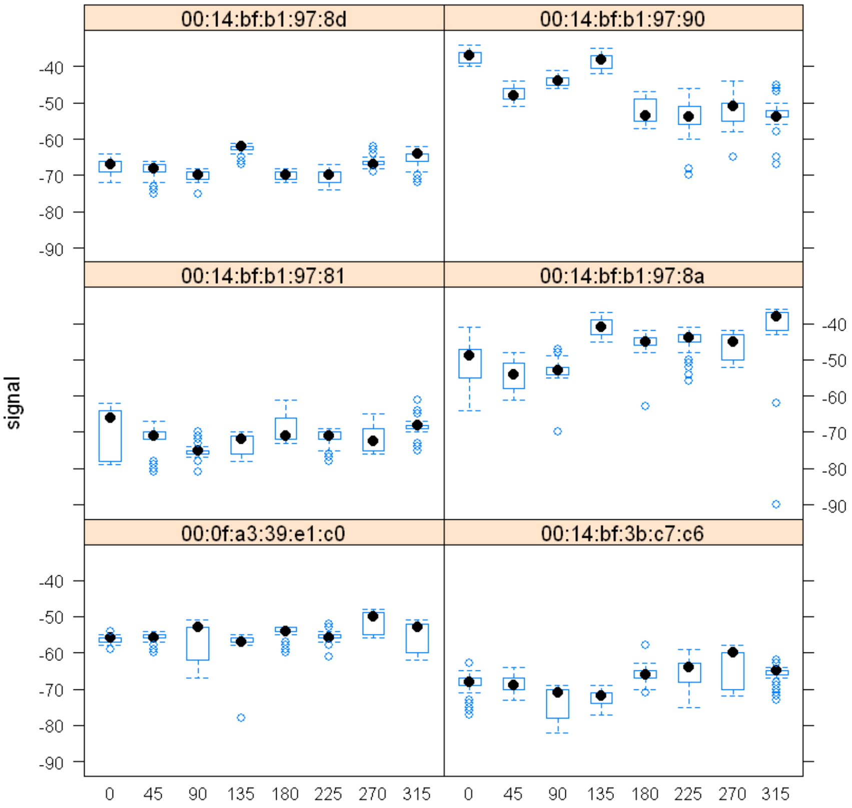

Figure 1.4: Signal Strength Boxplots by Angle at Each Access Point

Figure 1.5 further explores relationship between signal strength and angle. Again, using a fixed location, we examine the distributions of signal strength for different angles and access points by producing density curves. While many of the resulting distributions appear normal, there are several indications of skewness and bimodality. Additionally, the center of the distributions appears to vary with angle and access point indicating a dependency between signal strength and angle.

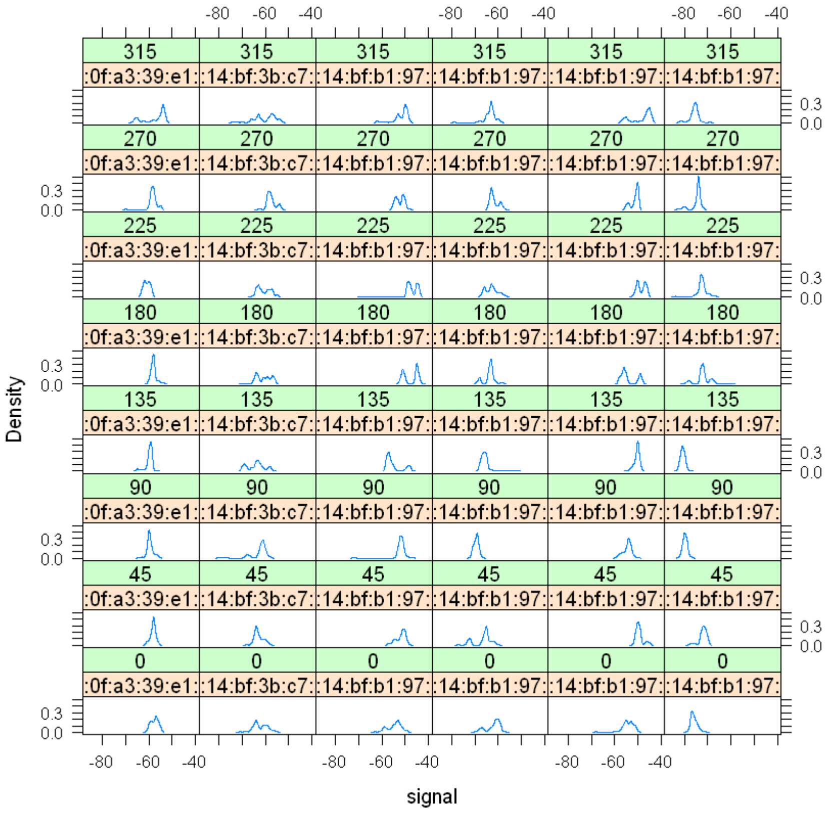

Figure 1.5: Signal Strength Distribution by Angle at Each Access Point

In order to examine signal strength at all possible combinations of location, orientation, and access points we use summary statistics. The boxplots in Figure 1.6 examine the relationship between the standard deviation and mean of all signal strength readings for each location-orientation-access point combination. We see from this figure that the variability of signal strength readings appears to increase as the mean signal strength increases.

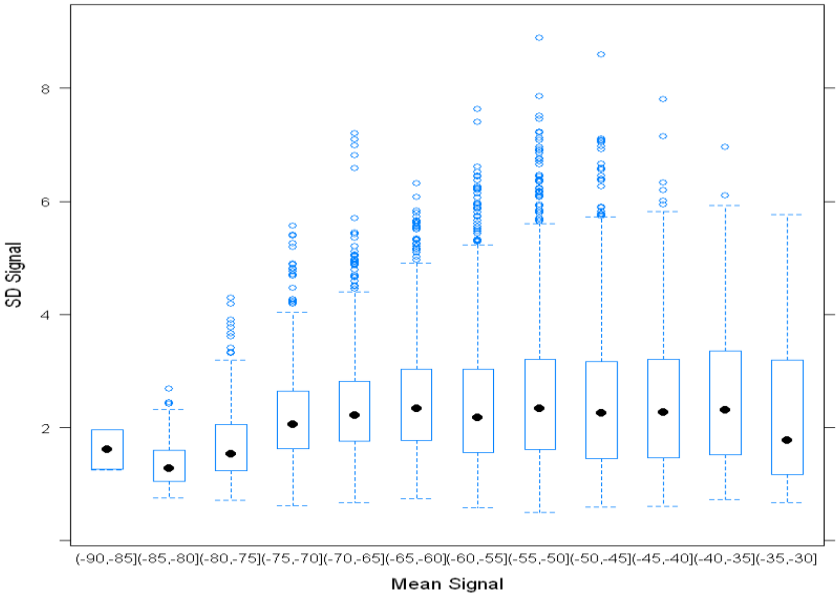

Figure 1.6: SD of Signal Strength vs. Mean Signal Strength

To examine this skewedness, we compare the mean and median of the previously created sample statistics. Figure 1.7 show a plot of the difference between the mean and median versus the number of observations. We find that in most cases these differences are close to zero with a typical deviation of 1 to 2 dBm.

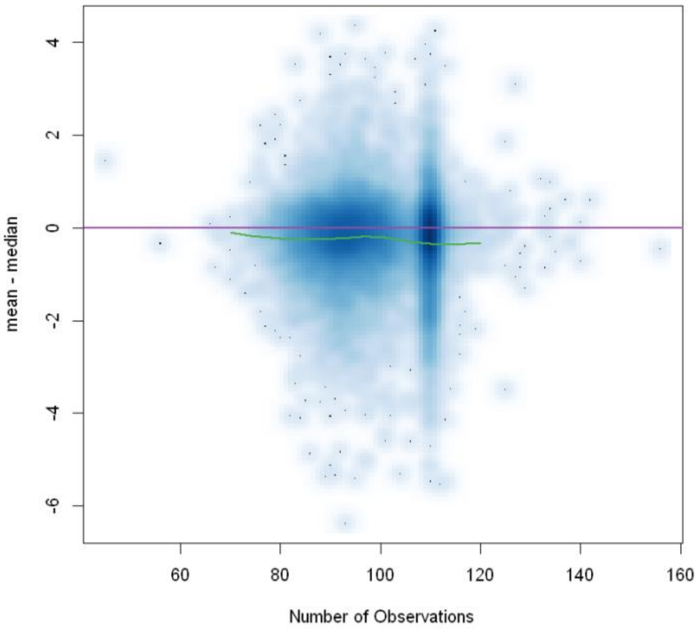

Figure 1.7: Mean vs. Median Signal Strength

Now that we've gained insight on the relationship between location, orientation, and access points versus signal strength, we move to examining the relationship between distance and signal strength. One way to do this is to create a contour plot, also known as a heat map. A contour plot visualizes the relationship between signal strength and distance between the hand-held device and access points. We choose to examine the summary statistic of a fixed combination of orientation and MAC address as we've done in previous analysis.

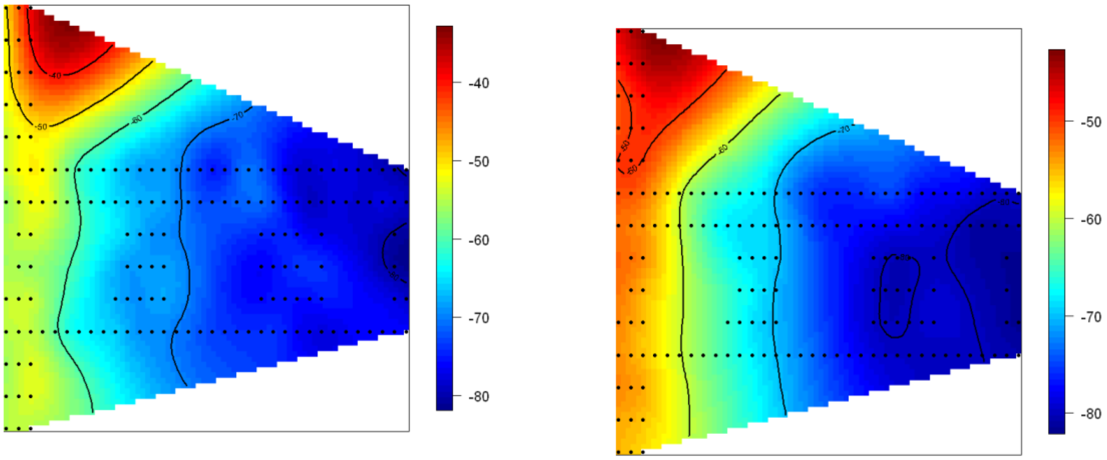

Figure 1.8: Heat Map of First Access Point

Figure 1.8 shows contour maps of the first access point at 0 and 135 degrees respectively. We can see that this access point corresponds to the access in the top left corner. Thus, these heat maps allow us to connect access points to MAC addresses. Further analysis shows two access points have similar heat maps, meaning they represent the same access point. One of these is dropped from the data frame. (This is additionally the reason previous figures only include 6 access points.)

As the final part of this portion on the analysis, we'll further examine the relationship between distance and signal strength. Figure 1.9 shows scatterplots of distance versus signal strength for each combination of orientation and access point. The shape is consistent across combinations and shows a clear negative correlation between distance and signal strength.

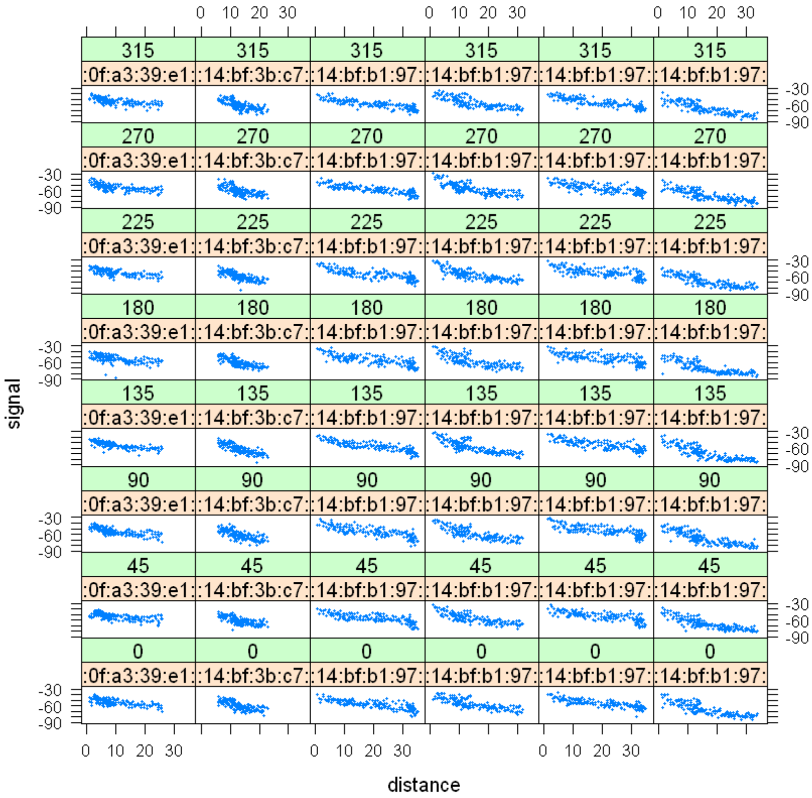

Figure 1.9: Scatterplots of Distance versus Signal Strength

Methods

Analysis of MAC Addresses

For this analysis, we utilized the code provided by Professor Slater and modified accordingly to address the following questions.

The k-nearest neighbors approach is used to determine locations by:
1) eliminating MAC address 00:0f:a3:39:dd:cd
2) eliminating MAC address 00:0f:a3:39:e1:c0
3) including both of the above two MAC addresses

For each method, an 11-fold cross-validation is used to find the optimal k value with the lowest sum of squared errors. Then, the resulting k is used to determine locations using the online dataset and to plot a floor map that shows the test locations and their respective predicted locations. At last, a comparison is made of the sums of squared errors of the three methods above.

Alternative Prediction Method

Our alternative prediction method is based on the weighted averaging method described in the assignment. The basic idea of the method is to use a simple weighted mean when taking an average of the X and Y coordinates of the nearest neighbors to calculate the X and Y coordinate of the point from the online dataset. This allows for the closer points to have a larger contribution to the k-nearest neighbor location calculation, as opposed to the points that are further away. The weight is calculated as:

$$w = \frac{1}{d} \qquad (1)$$

where w is weight of a neighbor and d is overall distance of the neighbor to the point.

Figure 2.1 below shows how the function (1) behaves on a range of values from 5 to 50.

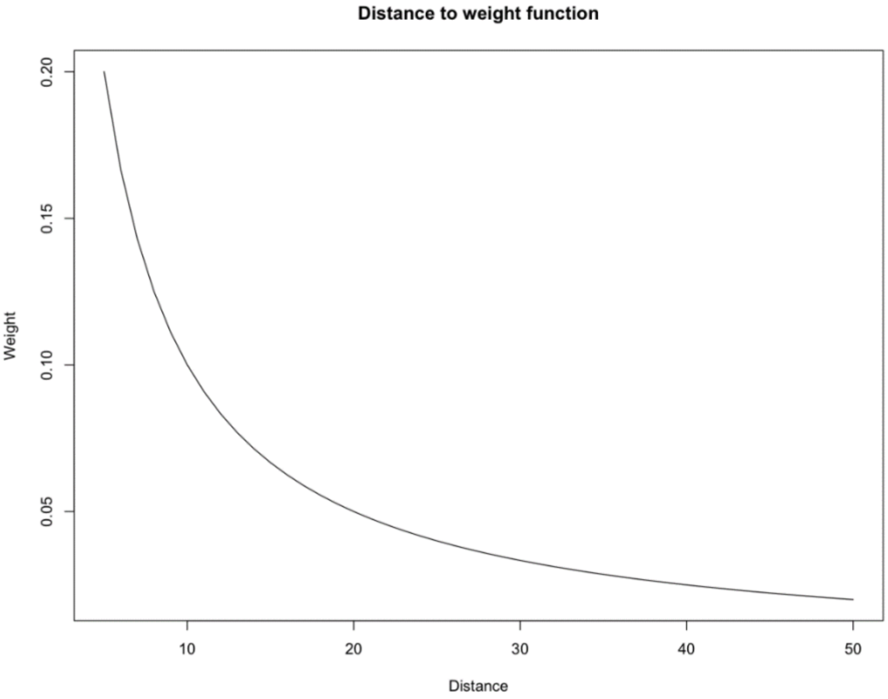

Figure 2.1 Example of function (1) behavior on specified range of values 5 to 50

When calculating X and Y coordinates of the point, we do a simple linear combination of neighbors weighted by their corresponding weights:

$$pos_x = w_1 \times x_1 + w_2 \times x_2 + \cdots + w_k \times x_k \quad (2)$$
$$pos_y = w_1 \times y_1 + w_2 \times y_2 + \cdots + w_k \times y_k$$

where k is the maximum number of neighbors.

The weight of the top k neighbors is normalized using formula (3).

$$w_{normalized} = \frac{w}{\sum_{i=1}^{k} w_i} \quad (3)$$

To implement our method, the *findNN* and *predXY* functions had to be redefined.

Results

Analysis of MAC addresses

Figure 2.2 shows the elbow curve using an 11-fold cross validation to determine the optimal number of k (number of nearest neighbors) without using the data related to MAC address 00:0f:a3:39:dd:cd. Using the suggested k value of 7 (with the lowest sum of squared errors), we then estimate the locations of the online dataset and map the floor plan with predicted and actual locations shown in Figure 2.3.

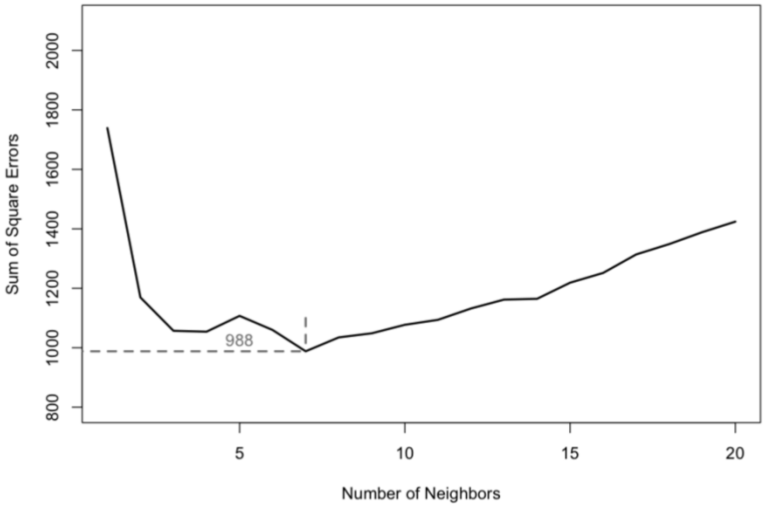

Figure 2.2: Cross Validated Selection of k (excluding MAC address 00:0f:a3:39:dd:cd). The sums of square errors are obtained via cross-validation of the offline data.

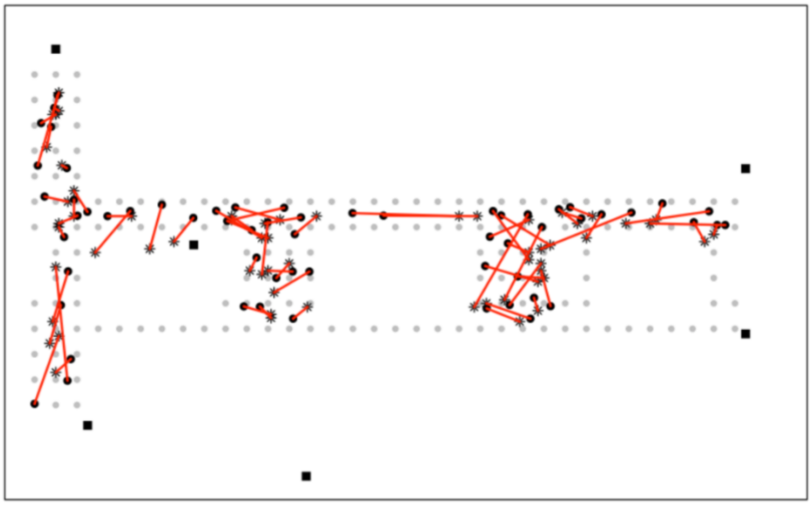

Figure 2.3: Floor Plan with Predicted and Actual Locations (excluding MAC address 00:0f:a3:39:dd:cd). The red line segments connect the test locations (black dots) to their predicted locations (asterisks).

Similarly, Figure 2.4 and Figure 2.5 show the elbow curve and the floor error map, respectively, without using the data related to MAC address 00:0f:a3:39:e1:c0. Just by visually looking at the two floor maps in Figure 2.3 and Figure 2.5, we cannot tell if one performs better than the other so we will look at the calculated sums of squared errors later in the section.

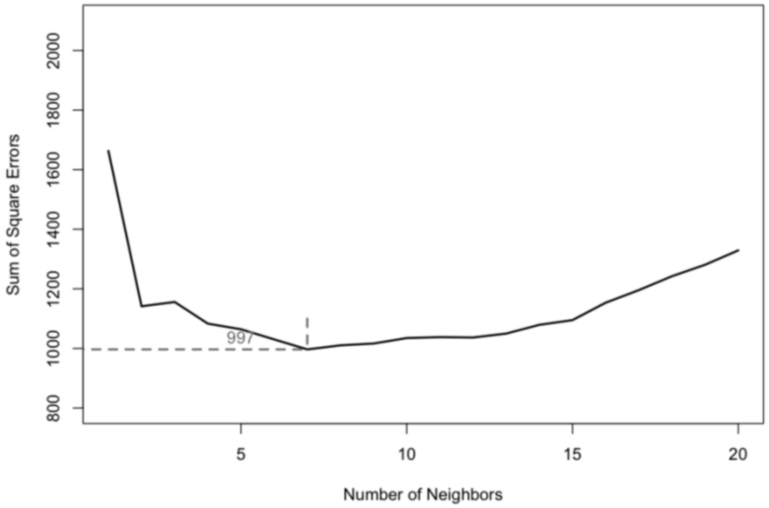

Figure 2.4: Cross Validated Selection of k (excluding MAC address 00:0f:a3:39:e1:c0). The sums of square errors are obtained via cross-validation of the offline data.

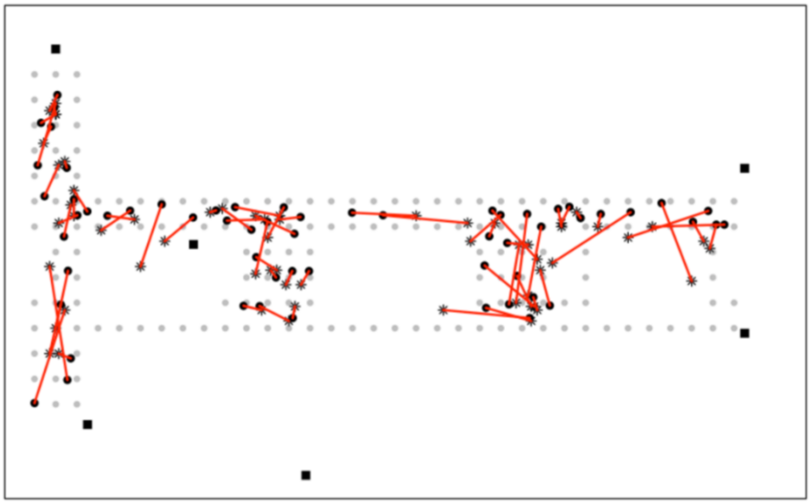

Figure 2.5: Floor Plan with Predicted and Actual Locations (excluding MAC address 00:0f:a3:39:e1:c0). The red line segments connect the test locations (black dots) to their predicted locations (asterisks).

Lastly, Figure 2.6 and 2.7 show the resulting elbow curve and floor error map, respectively, when including all 7 MAC addresses in the offline data. The floor error map seems to be performing the best among the three floor maps with shortest line segments.

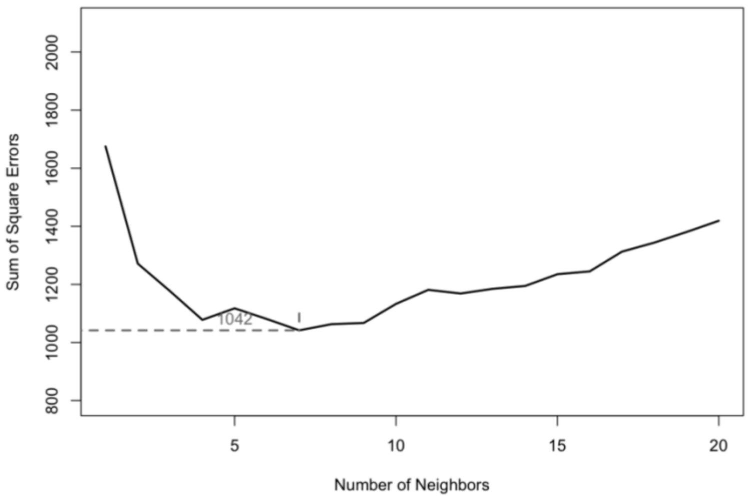

Figure 2.6: Cross Validated Selection of k (with all 7 MAC addresses). The sums of square errors are obtained via cross-validation of the offline data.

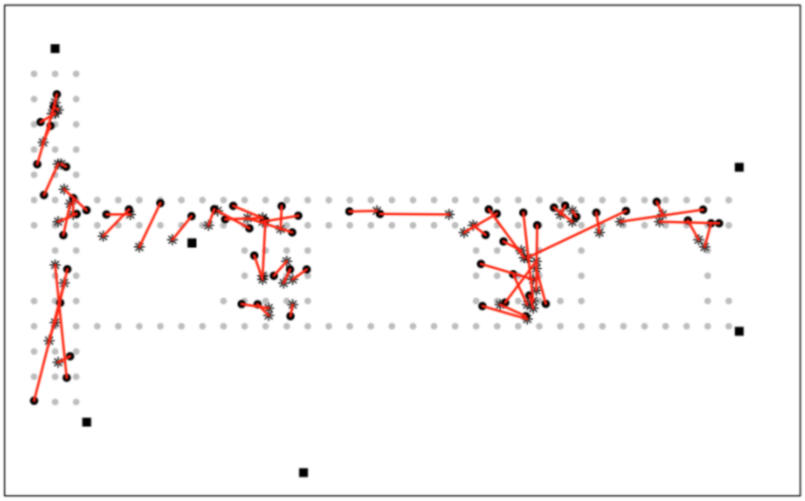

Figure 2.7: Floor Plan with Predicted and Actual Locations (with all 7 MAC addresses). The red line segments connect the test locations (black dots) to their predicted locations (asterisks).

As shown in the summary table (Table 2.1) below, excluding the access point with MAC address 00:0f:a3:39:e1:c0 has a lower sum of squared error than eliminating MAC address 00:0f:a3:39:dd:cd, the approach used by Nolan and Lang. Thus, keeping the access point with MAC address 00:0f:a3:39:dd:cd yields a more accurate prediction of location and should be used for RTLS.

Method	Sum of Squared Errors
Eliminate MAC address 00:0f:a3:39:dd:cd	273.680
Eliminate MAC address 00:0f:a3:39:e1:c0	258.075
Include both MAC addresses	221.444

Table 2.1: Summary Table of Sums of Squared Errors

In addition, using data for both MAC addresses results in lowest sum of squared errors and thus, it yields the best prediction of location. We believe that in this particular case, keeping both MAC addresses with the same access point increases the number of data points in the training data for determining locations using the k-nearest neighbors' approach, which in return, increases the prediction accuracy.

Alternative Prediction Method

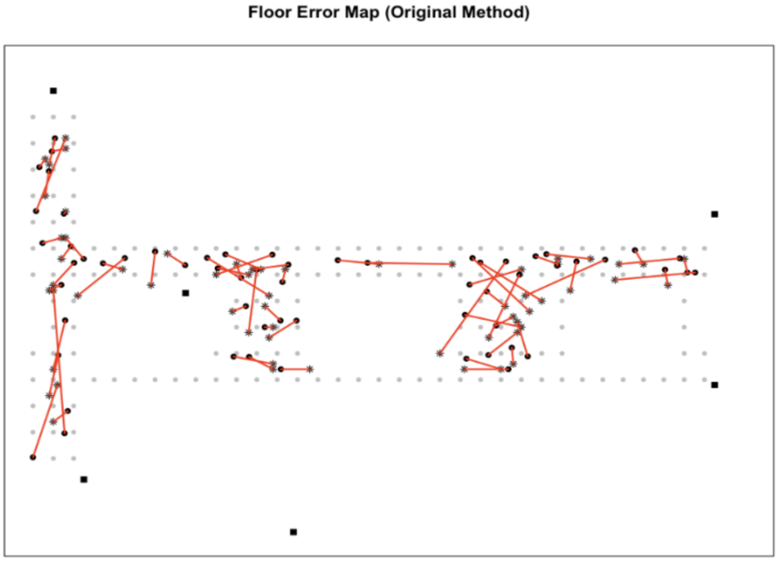

Figure 2.8: Floor Plan with Predicted and Actual Locations utilizing the original prediction method. The red line segments connect the test locations (black dots) to their predicted locations (asterisks).

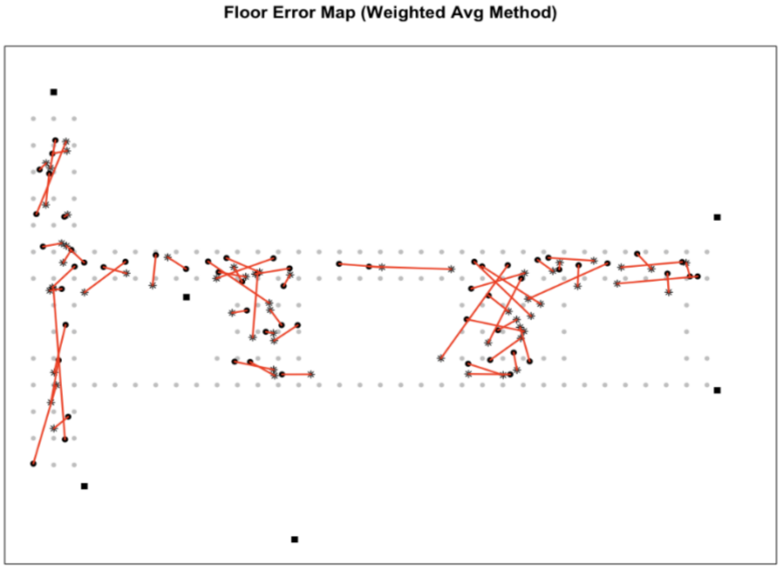

Figure 2.9: Floor Plan with Predicted and Actual Locations utilizing the alternate prediction, weighted average, method. The red line segments connect the test locations (black dots) to their predicted locations (asterisks).

As seen in Figures 2.8 and 2.9 above, both floor plans are very similar visually. It is hard to judge upon relative performance of the new method compared to the original method by visually looking at the floor plans. To quantify the difference, we decided to calculate the Sum of Squared Error (SSE) using the test dataset. The SSE for the original method was 275.50, while the SSE for the weighted average method was 273.12. Our method results in smaller SSE error and thus is slightly better.

A range of k values were input to determine the appropriate number of neighbors that would provide accurate classification, also being the lowest error value for the loss function. To estimate the optimal k, we are using 11-fold validation.

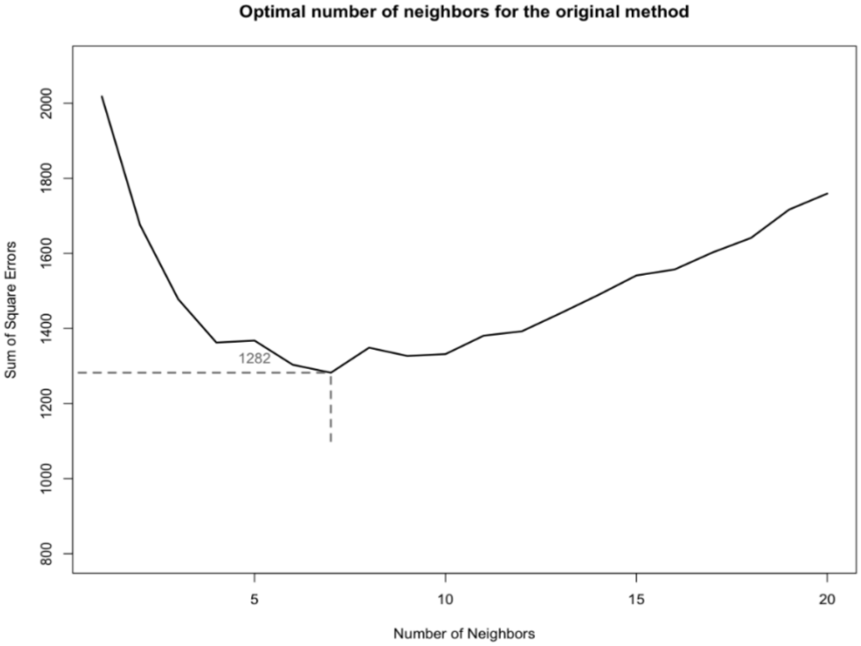

Figure 2.10 Elbow plot identifying the optimal number of neighbors utilizing the original prediction method.

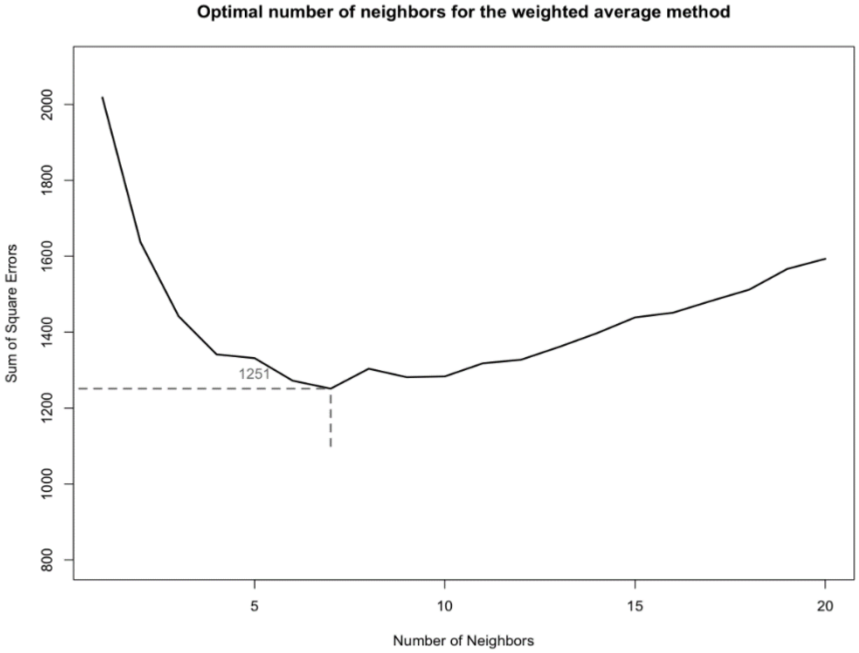

Figure 2.11 Elbow plot identifying the optimal number of neighbors utilizing the alternate prediction, weighted average, method.

The weighted KNN method uses the same concepts, while applying a penalty to neighbors that are further away from the access point. Based on the analysis, it seems that our alternate method has the same optional value, 7, for the number of nearest neighbors, k, as the original method.

Conclusion

Based on the results of the analyses, the RTLS technology seems to work well to predict the locations of the hand-held device with unknown positions. However, thorough assessment is needed before deciding to exclude data points. The results show that the MAC address that was being eliminated, performed better than the other MAC address that was kept in the data to build the model. Additionally, using data for both MAC addresses resulted in the lowest sum of squared errors, yielding the best prediction of locations. For this analysis, keeping both MAC addresses with the same access point increases the number of data points used in the training, which in turn, increases the prediction accuracy.

For the alternative method, based on the comparison of the two floor error maps, it is hard to visually judge relative performance. The SSE for the original method is 275.50, while the SSE for the weighted average method is 273.12. Our alternative weighted method results in a smaller SSE error, and thus, it performs slightly better than non-weighted method.

Additionally, it seems that the k value of 7 is being selected for all the cross-validations that we performed in this case study. Therefore, k = 7 seems to be appropriate to use for the k-nearest neighbors (KNN) clustering method for this specific dataset.

Sources

[1] D. Lang and D. Nolan, Data Science in R: A Case Studies Approach to Computation Reasoning and Problem Solving. New York, New York: CRC Press.
[2] Deborah Nolan and Duncan Temple Lang, “Case Studies in Data Science with R”. University of California, Berkeley and University of California, Davis. 2015. http://www.rdatasciencecases.org

Code

Analysis of MAC addresses

Method 1

```
# Modified to drop Mac id 00:0f:a3:39:dd:cd
orig_offline = offlineSummary

offlineSummary = subset(offlineSummary, mac != subMacs[2])


# Modified to change the XY positions calculation with k=7
estXYk7 = predXY(newSignals = onlineSummary[ , 6:11],
                 newAngles = onlineSummary[ , 4],
                 offlineSummary, numAngles = 3, k = 7)


# Modified to change the parameter to use the updated XY positions with k=7
floorErrorMap(estXYk7, onlineSummary[ , c("posX","posY")],
              trainPoints = trainPoints, AP = AP)
calcError(estXYk7, actualXY)
```

Method 2

```
# Modified to drop Mac id 00:0f:a3:39:e1:c0
orig_offline = offlineSummary
```

```
offlineSummary = subset(offlineSummary, mac != subMacs[1])

# Modified to update the Mac id
trainPoints = offlineSummary[ offlineSummary$angle == 0 &
                              offlineSummary$mac == "00:0f:a3:39:dd:cd"
,c("posX", "posY")]


# up to 20 neighbors, 11 folds
# Modified to exclude Mac id 00:0f:a3:39:e1:c0
offline = offline[ offline$mac != "00:0f:a3:39:e1:c0", ]

keepVars = c("posXY", "posX","posY", "orientation", "angle")

onlineCVSummary = reshapeSS(offline, keepVars = keepVars,
                            sampleAngle = TRUE)


# Modified to change the XY positions calculation with k=7
estXYk7 = predXY(newSignals = onlineSummary[ , 6:11],
                 newAngles = onlineSummary[ , 4],
                 offlineSummary, numAngles = 3, k = 7)


# Modified to change the parameter to use the updated XY positions with k=7
floorErrorMap(estXYk7, onlineSummary[ , c("posX","posY")],
              trainPoints = trainPoints, AP = AP)

calcError(estXYk7, actualXY)
```

Method 3

```
# Modified to keep all the Mac ids
orig_offline = offlineSummary

offlineSummary = subset(offlineSummary)


# Modified to add both Mac ids with the same access point
# signal strength vs distance
AP = matrix( c( 7.5, 6.3, 7.5, 6.3, 2.5, -.8, 12.8, -2.8,
                1, 14, 33.5, 9.3, 33.5, 2.8),
            ncol = 2, byrow = TRUE,
            dimnames = list(subMacs, c("x", "y") ))


# Modified to update the ncol to be 7
keepVars = c("posXY", "posX","posY", "orientation", "angle")
byLoc = with(online,
             by(online, list(posXY),
                function(x) {
                  ans = x[1, keepVars]
                  avgSS = tapply(x$signal, x$mac, mean)
                  y = matrix(avgSS, nrow = 1, ncol = 7,
                        dimnames = list(ans$posXY, names(avgSS)))
                  cbind(ans, y)
                }))
onlineSummary = do.call("rbind", byLoc)


# Modified to change the XY positions calculation with k=7 and the number
of columns used (6:12)
estXYk7 = predXY(newSignals = onlineSummary[ , 6:12],
                 newAngles = onlineSummary[ , 4],
                 offlineSummary, numAngles = 3, k = 7)


# Modified to change the parameter to use the updated XY positions with k=7
floorErrorMap(estXYk7, onlineSummary[ , c("posX","posY")],
              trainPoints = trainPoints, AP = AP)

calcError(estXYk7, actualXY)
```

Alternative Prediction Method

```
# Our findNN function.
findNN = function(newSignal, trainSubset) {
  diffs = apply(trainSubset[ , 4:9], 1,
                function(x) x - newSignal)
  dists = apply(diffs, 2, function(x) sqrt(sum(x^2)) )
  closest = order(dists)
  result = trainSubset[closest, 1:3 ]
  result$weight = 1 / dists[closest]
  return(result)
}

# Our weighted mean function
weightedMean = function(x, w, k) {
    normalized_w = w[1:k] / sum(w[1:k])
    result <- (sapply(x[ , 1:2], function(x) sum(x[1:k] * normalized_w)))
    return (result)
}

# Original mean function
regularMean = function(x, w, k) {
    result <- (sapply(x[ , 1:2], function(x) mean(x[1:k])))
    return (result)
}

# predict X-Y based on the the neasest k neighbors (default 3)
predXY = function(newSignals, newAngles, trainData, meanFunc,
                  numAngles = 1, k = 3){

  closeXY = list(length = nrow(newSignals))
  for (i in 1:nrow(newSignals)) {
    trainSS = selectTrain(newAngles[i], trainData, m = numAngles)
    closeXY[[i]] =
      findNN(newSignal = as.numeric(newSignals[i, ]), trainSS)
  }

  estXY = lapply(closeXY,
                 function(x) meanFunc(x[ , 2:3], x[, 4], k))
  estXY = do.call("rbind", estXY)
  return(estXY)
}

# Display Distance to Weight function
x_distance <- seq(5, 50, by = 1)
y_distance_transformed <- 1 / x_distance
plot(x_distance, y_distance_transformed, type="n", main="Distance to weight
function",
     xlab="Distance", ylab="Weight")
lines(x_distance, y_distance_transformed, type="l")

# Original method
estXYk5Orig = predXY(newSignals = onlineSummary[ , 6:11],
                 newAngles = onlineSummary[ , 4],
                 trainData = offlineSummary,
                 meanFunc = regularMean,
                 numAngles = 3,
                 k = 5)

floorErrorMap("Floor Error Map (Original Method)", estXYk5Orig,
onlineSummary[ , c("posX","posY")],
              trainPoints = trainPoints, AP = AP)

# Weighted Average method
estXYk5Weighted = predXY(newSignals = onlineSummary[ , 6:11],
                 newAngles = onlineSummary[ , 4],
                 trainData = offlineSummary,
                 meanFunc = weightedMean,
                 numAngles = 3,
                 k = 5)

floorErrorMap("Floor Error Map (Weighted Avg Method)", estXYk5Weighted,
onlineSummary[ , c("posX","posY")],
              trainPoints = trainPoints, AP = AP)
```