MSDS 7333 Fall 2020: Case Study 1

Jayson Barker, Brandon Croom, Shane Weinstock

9/7/2020

Business Understanding

Organizations measure productivity in various ways depending upon the metrics they want to track. They may look at shipping times for materials, optimization of workflows within the organization, and many are focused on cost minimization. One such way organizations can help within this optimization is through leveraging real-time location systems (RTLS). Real-time location systems allow organizations to monitor assets, movement of goods, and workers near real-time through the workflows that exist within the organization.

The evaluation of RTLS as it relates to an organization's WiFi usage is one such method to analyze productivity. More specifically, the distribution of various wifi-enabled RTLS devices across a facility may allow clustering methods to be leveraged in order to predict the position of assets based on previous patterns. In this particular evaluation, data were provided for offline and online data across a variety of access points. The building layout, access point configuration, online measurements (black dots), and offline measurements (gray dots) are shown below in Figure 1.

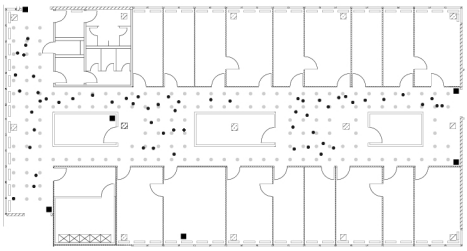


Figure 1: Building Floorplan

In order to perform the analysis k-Nearest Neighbor (KNN) algorithm will be used. KNN is an algorithm that can be used as both a classifier and a regressor. In this analysis KNN will be used as a classifier. In the classifier configuration, KNN attempts to map items that are close to each other together into clusters of other similar items. KNN is a lazy algorithm and large datasets do impact the speed of the KNN results. This algorithm works well in the context being discussed as the access points can be thought of as anchors for the individual clusters and the reader measurements can be grouped to those access points closest to the reader.

The objectives for this case study are:

- evaluate the data to determine the optimal access point configuration to include in the data
- evaluate both a regular unweighted KNN and weighted K-Nearest Neighbor (KNN) methodology to perform predictions based on the offline data set to predict locations for the online data set

Data Engineering

In evaluating the data that has been made available it is critical to understand the data file structure. There are two data files to work with: online.final.trace.txt and offline.final.trace.txt. The offline file contains information relating to the offline measurements that were taken. The online file contains information relating to the online measurements that were taken. For the purposes of this analysis, the offline file makes up the training dataset while the online file is our test dataset. Both files are structured with the data fields noted below:

Data Field	Data Description
t	Time stamp in milliseconds
Id	Router MAC Address
Pos	Router Location (Comma separated positions for x, y, z)
Degree	Direction scanning device measured in degrees that was carried by the researcher
MAC	MAC address of either the router, or scanning device combined with corresponding values for signal strength (dBm), the mode in which it was operating(adhoc scanner = 1, access router = 3), and its corresponding channel frequency
Signal	Received Signal Strength in DbM

As noted above, the data fields have compressed information into single columns. For example, the Pos column contains information for X,Y, and Z coordinates in comma delimited format instead of being broken out into individual fields. The same is true for the MAC column. The MAC column contains information on the router MAC address, signal strength, operating mode and channel. In order to handle all of this information dataframes were constructed for both the online and offline datasets. Each field was split out into a column within the dataframe in order to make analysis much easier.

In reviewing the data in raw form a few observations were identified:

- The Z position (i.e. elevation) was zero for the data points. We will ignore this column in the analysis
- The scan angles are not consistent and need to be made consistent. We will round these to the nearest 45 degree angle
- There are some MAC addresses with only a few data points. A threshold of approximately 1 million will be set. Any MAC addresses with less than this threshold will be removed

In relation to this analysis the following features will also be dropped from the dataset as they do not impact the analysis:

- scanMac - this is the MAC address of the scanner
- channel - this is the channel where the reading was acquired
- type - this is the device type

Multiple columns will be created to assist in further analysis

- medSignal - this will be the median signal strength captured
- avgSignal - this will be the average signal strength captured
- sdSignal - this will be the standard deviation of the captured signal strength
- iqrSignal - this is the interquartile range of the captured signal strength
- posXY - this is a combination of the posX, posY fields into a single value. Coordinates are separated by the '-' character

- rawTime - this is a raw time value

In addition to the above the angle column was standardized to the nearest 45 degree angle. Due to multiple differences in the angle readings, this standardization is necessary to ensure consistency.

For purposes of this analysis these rules were applied to both the online and offline datasets.

As part of the analysis the determination of certain access points to keep or remove was needed. Specifically, evaluation of whether to individually keep both access points that ended in C0 and CD or to keep only one of these points. This need resulted in three distinct dataframes for each file as follows:

Dataframe	Description
offlineSummary	The offline dataframe containing only the C0 access point
offlineSummary2	The offline dataframe containing only the CD access point
offlineSummaryall	The offline dataframe containing both C0 and CD
onlineSummary	The online dataframe containing only the C0 access point
onlineSummary2	The online dataframe containing only the CD access point
offlineSummaryall	The online dataframe containing both C0 and CD

All of the dataframes have the same structure. For illustrative purposes a view of the offlineSummaryall dataframe showing the available features is below:

```
##             Length Class  Mode
## time        7968   POSIXt numeric
## posX        7968   -none- numeric
## posY        7968   -none- numeric
## orientation 7968   -none- numeric
## mac         7968   -none- character
## signal      7968   -none- numeric
## rawTime     7968   -none- numeric
## angle       7968   -none- numeric
## posXY       7968   -none- character
## medSignal   7968   -none- numeric
## avgSignal   7968   -none- numeric
## num         7968   -none- numeric
## sdSignal    7968   -none- numeric
## iqrSignal   7968   -none- numeric
```

In preparation for data modeling, the data has also been flattened to create a dataframe with a single positional entry (X, Y coordinate) for each MAC address. This will help to reduce the analysis record count down to approximately 1328 points per MAC address. this breakdown is shown below:

```
##                      time posX posY orientation               mac signal
## 7      2006-02-11 02:31:58    0    0         0.0 00:0f:a3:39:dd:cd    -75
## 24512  2006-02-11 03:03:44    0    1         0.7 00:0f:a3:39:dd:cd    -73
## 221777 2006-02-11 06:46:02    0   10         0.9 00:0f:a3:39:dd:cd    -69
## 229191 2006-02-11 06:54:05    0   11         0.4 00:0f:a3:39:dd:cd    -72
## 236493 2006-02-11 07:02:32    0   12         0.2 00:0f:a3:39:dd:cd    -72
## 243546 2006-02-11 07:14:12    0   13         0.1 00:0f:a3:39:dd:cd    -69
##             rawTime angle posXY medSignal avgSignal num sdSignal iqrSignal
## 7      1.139643e+12     0   0-0       -67 -68.12613 111 3.159739       4.5
## 24512  1.139645e+12     0   0-1       -70 -70.33636 110 2.420392       3.0
## 221777 1.139658e+12     0  0-10       -67 -69.24324 111 4.407053       6.5
```

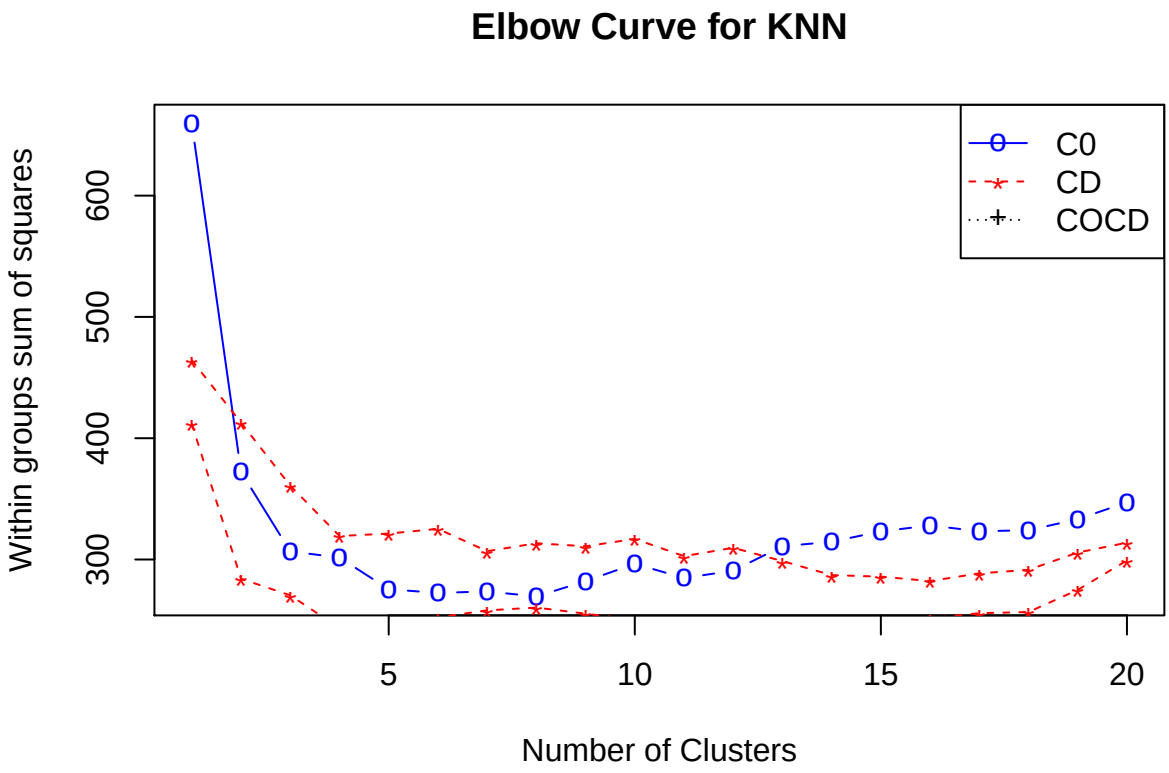
```
## 229191 1.139659e+12     0  0-11       -73 -74.30631 111 5.396620       7.0
## 236493 1.139659e+12     0  0-12       -70 -69.57895 114 2.372015       3.0
## 243546 1.139660e+12     0  0-13       -74 -73.58716 109 3.493939       4.0
```

```
##                 mac count
## 1 00:0f:a3:39:dd:cd  1328
## 2 00:0f:a3:39:e1:c0  1328
## 3 00:14:bf:3b:c7:c6  1328
## 4 00:14:bf:b1:97:8a  1328
## 5 00:14:bf:b1:97:8d  1328
## 6 00:14:bf:b1:97:90  1328
```

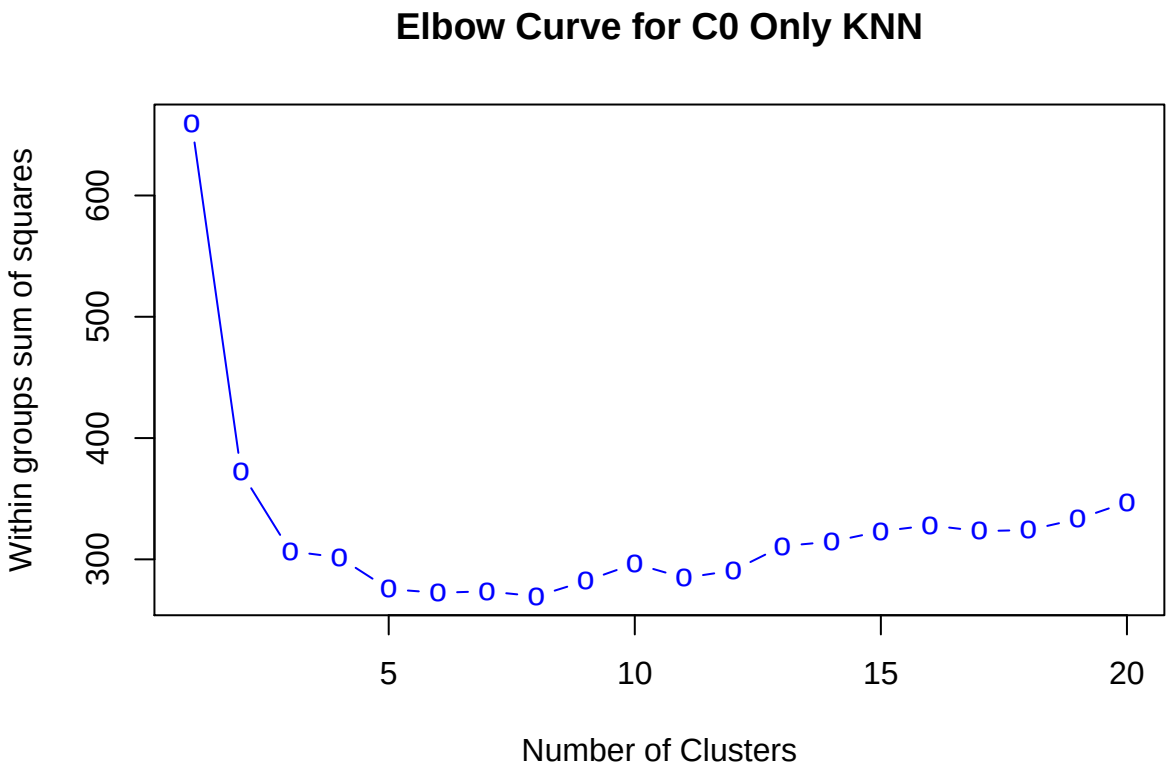
Building the dataframes as described above with flattened data and dataframes containing the different access points allows for providing quick analysis of the data.

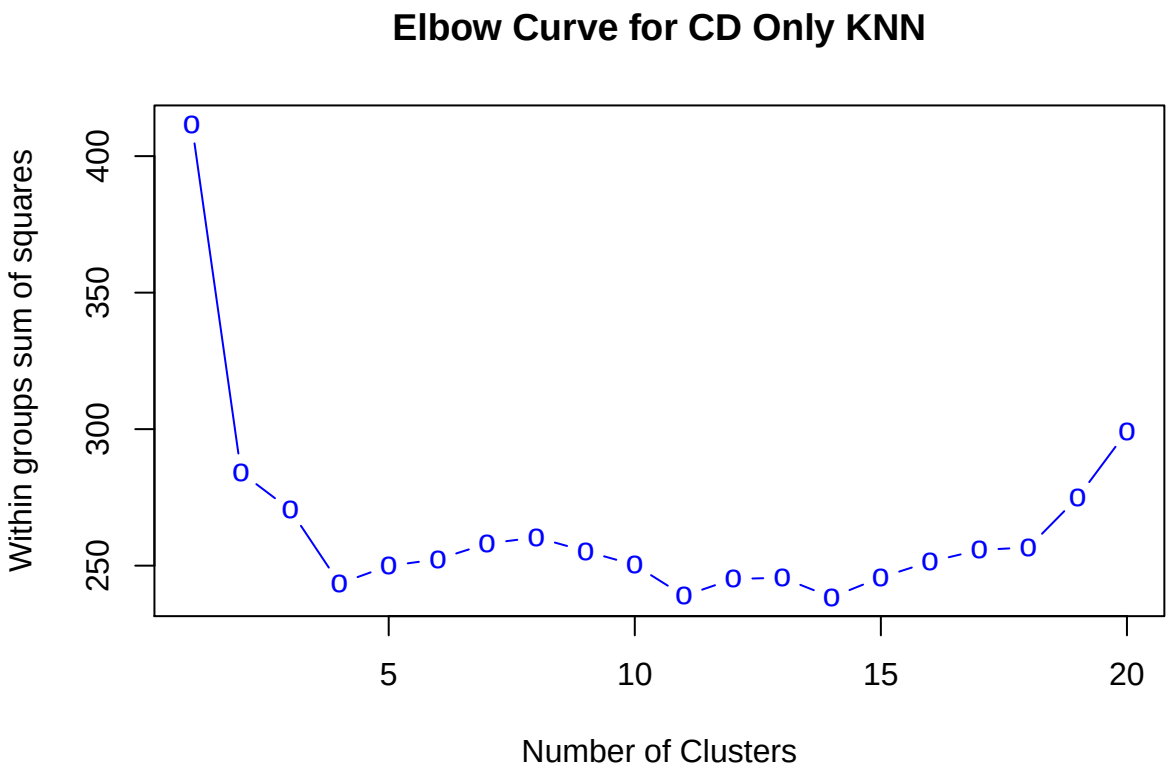
Access Point Analysis

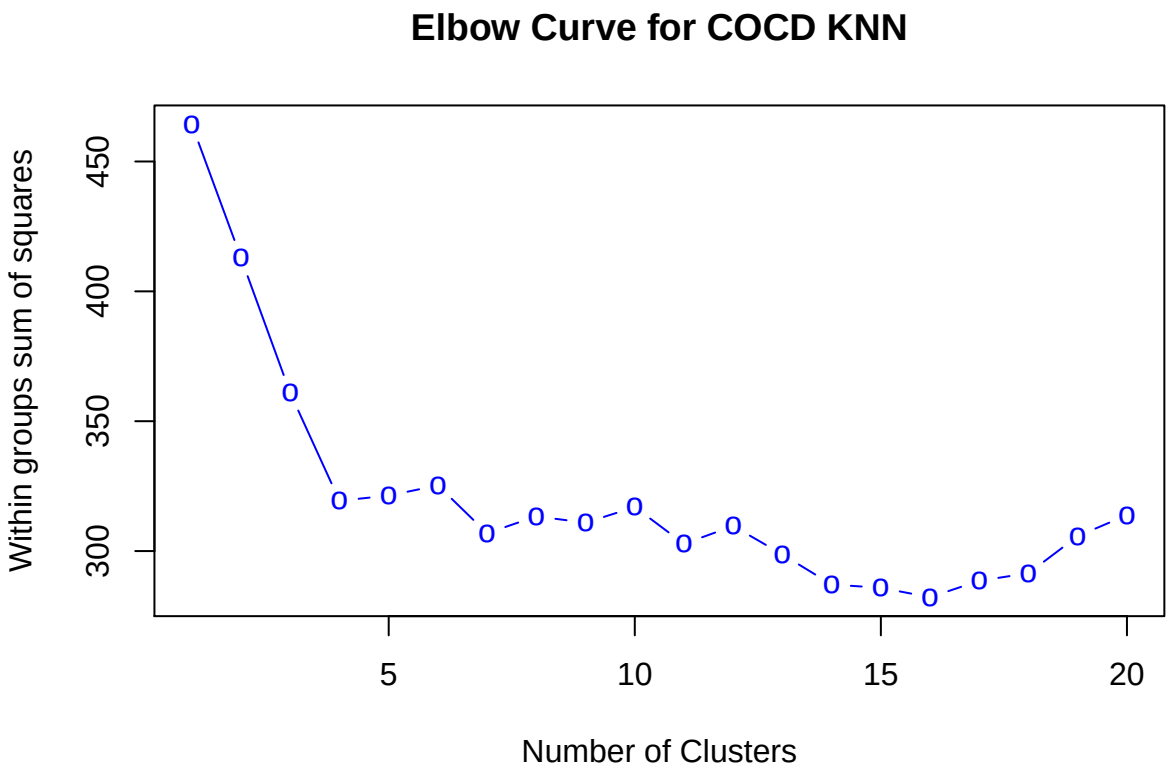
One of the analysis points that must be determined is whether to use the MAC address ending with C0 only, the MAC address ending in CD only, or to keep both in our dataset. In order to determine which approach to take, KNN methodology will be executed on all three configurations. Using this information it can be determined which data configuration will be used based on the error rates output by executing KNN. For each data configuration k will be selected from 1 through 20. This is an arbitrary selection but should provide enough data points to make an analysis. Initially the elbow plots for the three scenarios will be graphed together to assist in determining which scenario best fits.



Viewing the combined elbow plot above it can be seen that for the CD configuration a k=4 looks to be about the optimal solution. For C0, k=8 looks to be optimal. From the combined graph, it's difficult to tell what the optimal K for COCD should be. Evaluating the plots individually will assist in evaluating the scenarios.




Elbow Curve for CD Only KNN
Within groups sum of squares
250
300
350
400
5
10
15
20
Number of Clusters



Evaluating the individual graphs reaffirms the values for C0 and CD as standalone cases; k=8 and k=4 respectively. The combined case gives a k=7 that looks to be optimal. Lets now look at the individual errors at these k values to determine which best meets the analysis needs.

```
## [1] "CO KNN Error at K=5: 269.507800"
```

```
## [1] "CD KNN Error at K=4: 243.430300"
```

```
## [1] "COCD KNN Error at K=7: 306.852137"
```

From the output above it can be seen that the combination with CO and CD has the highest error. This would lead to not keeping both values in the mix. Given CD by itself has the lowest error this seems the most likely option to address.

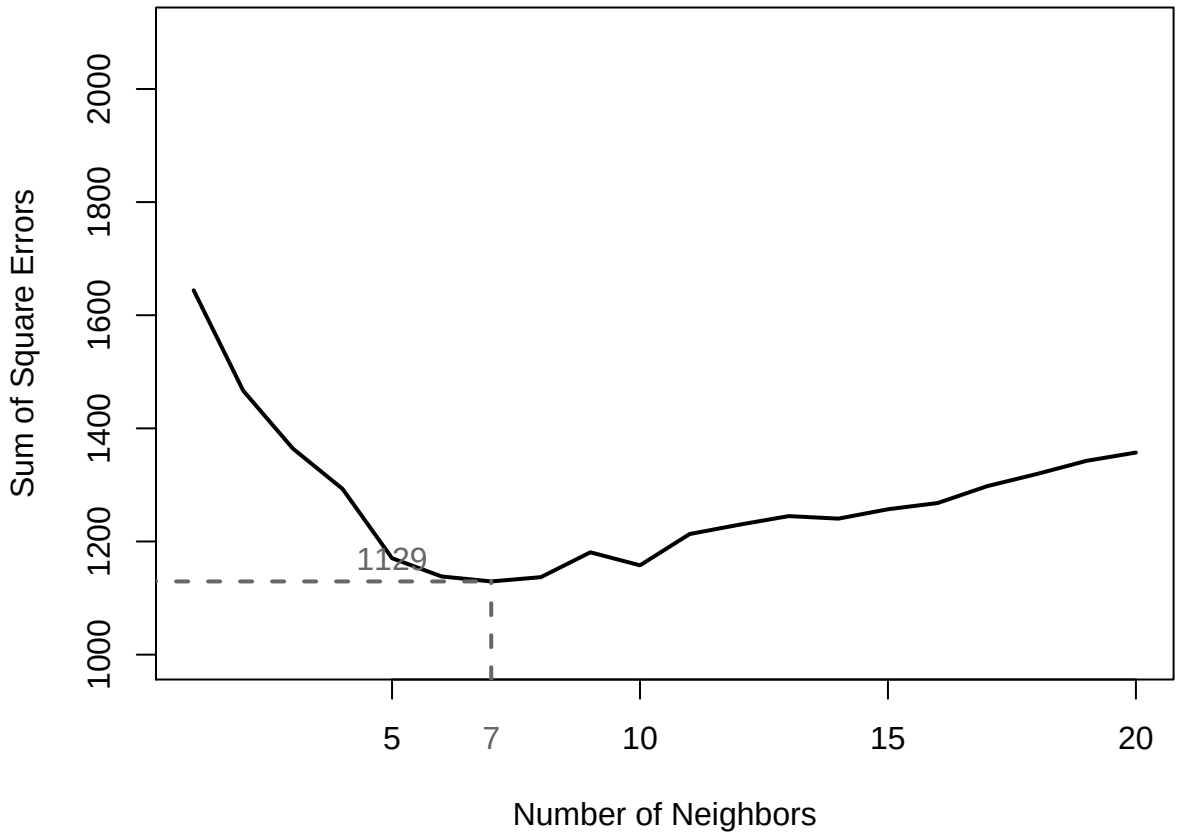
Weighted KNN

Now that it's been assessed that the dataset with just CD looks to be the best another verification will be performed on this dataset using a weighted KNN. Leveraging a weighted KNN should allow for improving the accuracy of predicting locations. In order to perform the weighting, the signal strength column will be used as the weighting factor. Using this methodology allows for stronger, closer signals to have a larger impact on determining location than those with weaker signals. This is different than the simple KNN used above in that locations of closer distance may now have a larger impact to the neighbor grouping that those that are further away.

In order to make the weighted KNN work as expected our current data structures can be leveraged. The structure of the current dataframes is adequate to support the weighted KNN methodology. Some engineering is required within individual functions as part of the analysis; largely these engineering changes are modifications to compute the weight and output distance values based on the weighting calculations.

As part of this weighted KNN analysis cross fold validation will also be leveraged. In this particular run 11 fold validation will be leveraged. As with the previous simple KNN calculations k values from 1 to 20 will be leveraged in order to determine the optimal number of neighbors for this analysis.

```
## Warning in matrix(permuteLocs, ncol = v, nrow = floor(length(permuteLocs)/v)):
## data length [166] is not a sub-multiple or multiple of the number of rows [15]
```



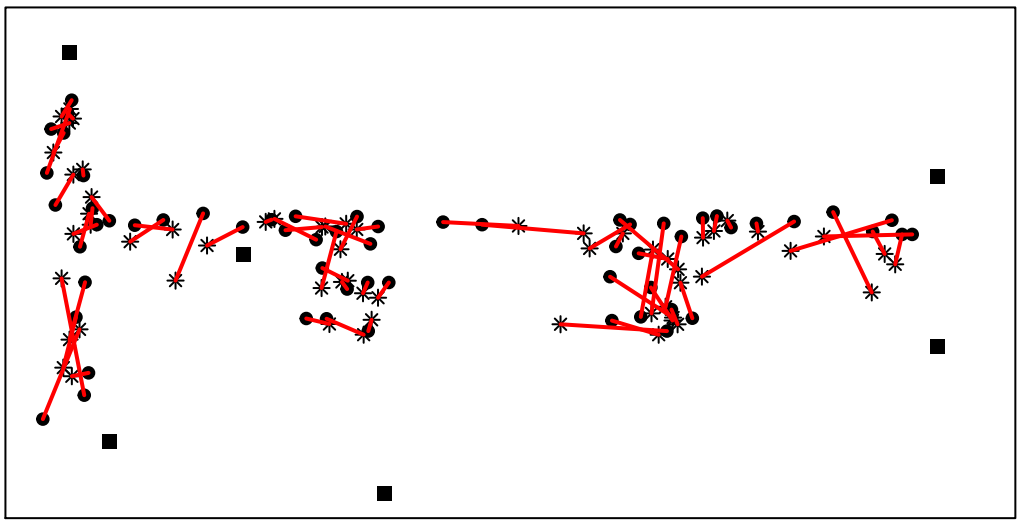
The elbow plot above shows that k=8 seems to be the optimal number of neighbors for this weighted KNN evaluation. Furthermore, evaluating the error present at k=8 will allow for comparison between the simple KNN and weighted KNN.

```
## [1] "CD Weighted KNN Error at K=8: 248.353546"
```

Comparing the weighted KNN error (shown above) for KD and the simple KNN error (shown below) for CD we can see the errors are roughly similar.

```
## [1] "CD KNN Error at K=4: 243.430300"
```

The distance calculation for the weighted KNN can also be visualized, as shown below. This visualization allows for determining how the proximity of points is pulled together.



Conclusion

In conclusion the analysis was able to leverage a simple KNN to evaluate whether to maintain the C0, CD or both MAC addresses. Through the analysis it was determined that CD was the MAC address that should be maintained due to having the least calculated errors. The dataset containing CD was then passed through a weighted KNN model to determine if leveraging the signal strength for location had any impact on the evaluation. The weighted KNN showed that eight neighbors was the optimal number of neighbors for the analysis. The simple KNN version had k=4. Given the smaller number of neighbors in the simple KNN algorithm, the weighted KNN with a higher number of neighbors would be more appropriate to use for further analysis. This larger k and the weights applied based on signal strength provide a more reliable and consistent approach in part due to the larger pool of neighbors which covers a more defined general pool of clusters.

Using the KNN methodology (simple or weighted) does have some drawbacks when applied to RTLS. Speed is one predominant factor that comes into play. KNN is not speedy and in this particular instance it was required to aggregate the data on angle of orientation to obtain a speed boost for processing. The speed boost was necessary to make this analysis work however for true real time analysis this may not be enough for real time tracking.

The angles provided in the data were not overly useful for the KNN method. Given that the devices were always held at multiple angles based on user preference. Having this information did not add any value as the main goal was to determine location. The angle may be more applicable if the analysis was over multiple floors and elevation came into play.

In order to overcome some of these perceived drawbacks a few options could be implemented. Caching of the training data at the angle sweeps to prevent re-creation of data would dramatically help improve processing

speed. Increasing the size of the testing test by including the cached angle information would allow for the creation of many kNN models and then select the one that best met the needs. This would help improve both speed an accuracy.

Varying the number of angles included in the training aggregation would also be an interesting analysis to perform. In the current analysis angles were managed in 45 degree increments. An argument could be made that including all available angles would be ideal to prevent "data loss", however there could be a performance impact leveraging this approach.