Jordan Salsman

Unit 2 Case Study
Jordan Salsman
January 17th, 2021

I. Introduction

Real-time locations systems (RTLS) are a way to track the location of either people or items. It is often the case that RTLS are used to track items in confined spaces. The reason behind this is that RTLS allows for more exact tracking of things and therefore is best used in smaller spaces. For instance, other types of global tracking that is less exact would be by satellite or cell phone pings off of towers. These methods would not be as useful as RTLS in tracking down where an asset is inside of a building.

As businesses are constantly looking for ways to become more efficient and productive, RTLS becomes a more viable and important part of many companies. In the textbook "Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving" by Nolan and Lang, explores an RTLS experiment developed by the University of Mannheim in Germany. The experiment involved looking at the addresses of different macs and trying to develop a knn model that could help with the tracking. The report will further the analysis down in Nolan and Lang by revisiting their approach and offering new potential solutions.

II. Background

The data used in both the research was collected at by the University of Mannheim by placing seven macs at different positions within a controlled room. It is important to note that two of these macs were placed at the same location and therefore had the same (x,y) coordinates. The room itself was 15 by 36 meters. Each mac had its own access point associated with it. The orientation (angle) of the position was also included within the data. The data is broken up into two different datasets. The offline data set which serves as the training data and the online set which is used as the test set.

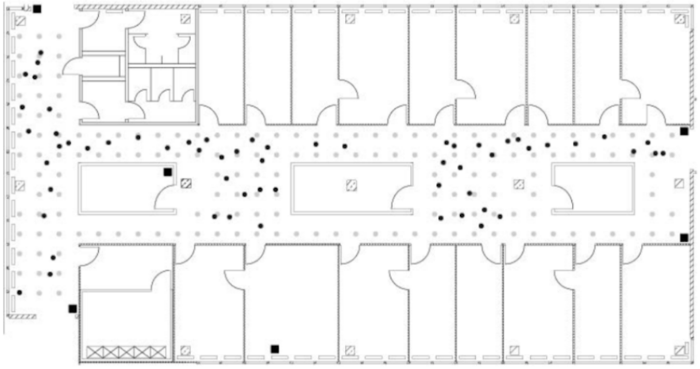


Figure 1 Mannheim University Floor Layout

In the above figure the six black squares are the six different access points. There are seven macs but two share the same access point. The offline dataset is denoted by the gray dots and was taken by a handheld measurement. The black dots are for the online dataset and were taken randomly throughout.

The data needed a fair amount of cleaning before it was readable into a dataset. This was accomplished by following the code provided by Nolan and Lang.

From there this research aims to extend the research done previously. Nolan and Lang used six of the seven macs to create their knn model. They arbitrarily chose to use the mac with the id ending in “c0” and exclude the mac at the same address with the id ending in “cd”.

III. Methodology

Like Nolan and Lang, k-nearest neighbors modeling is the modeling type chosen. This modeling uses the position of the macs as the response feature.KNN works by assigning an unknown observation to a certain value based on its proximity to other known observations. In KNN, how proximity, or distanced, is measured can vary the results of the model. Here, the standard euclidean distance was chosen by Nolan and Lang and is also continued in this research. The other vital component in the KNN model is to choose the number of neighbors in which to use. The process is known as tuning the model. In order to select the optimal amount of nearest neighbors, cross-validation and and mean-squared error will be used. Five fold cross-validation splits the data into 5 different folds and trains the knn model on four of the folds and tests on the fifth. This is repeated until every fold is the test fold and the average mean-squared error is our result. Mean-squared error is a metric used for measuring the accuracy of the knn prediction by

taking the difference of the predicted values with the actual values and then squaring them.

An important part to note is that most of the time knn works as a classifier and the model works by assigning a classification of a certain value based on a majority rule of its neighbors. This knn uses regression instead so instead of a simple vote of the majority, an average of the nearest neighbors is taken.

This research goes beyond the attempts of Nolan and Lang by also looking at the other mac observation at the same address. Furthermore, modeling is also done on a training set that includes all seven macs and does not exclude either of the macs at the same position. The final extension of Nolan and Lang is to use a weighted k-nearest neighbor model.

The weighted k-nearest neighbor model will use the formula:

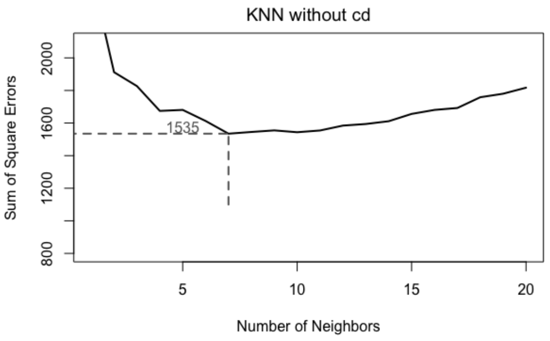
$$\sqrt{(}\ \sigma_x^2 * \sum_{1}^{i} x^2\ )$$

This is almost identical to the Euclidean distance formula. The only difference is this has a weighted element to it. That weight is the sigma squared of x, or variance. The idea behind using variance is to proportion the distances of each x and give more weight to those points closer in the nearest neighbors model.

IV. Results

The results are from the four KNN models mentioned above, three new ones, as well as reproducing the KNN model from Nolan and Lang. All four models are tuned to produce the lowest mean-squared error as that is the metric chosen as accuracy is the goal.

Jordan Salsman



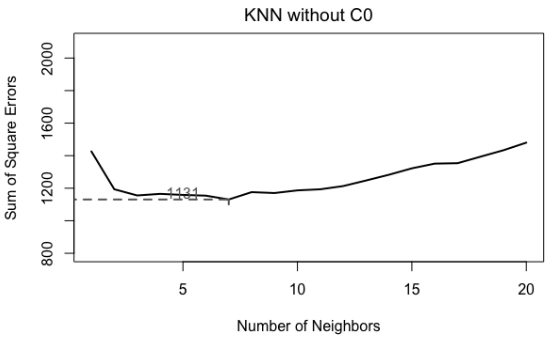
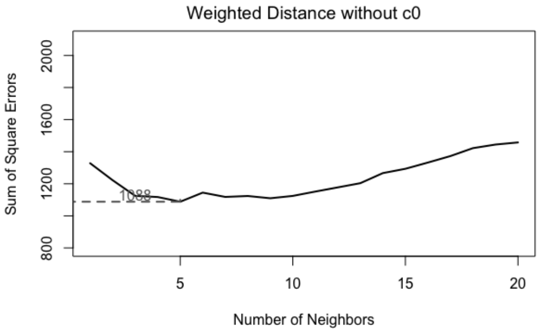


Figure 2: KNN without cd mac Reproduced from Nolan and Lang

Figure 3: KNN without c0 mac

These plots show that the KNN model performed better when the mac ID ending in "C0" was excluded instead of the KNN model where "cd" was excluded. This shows that the better mac ID for KNN models to include is the one "cd" in it. The sum of squared errors for the model without "cd" 1535 where the SSE for the other is 1131. This is a fairly large difference. It is also notable that the optimal KNN model for each has the value of k = 7.



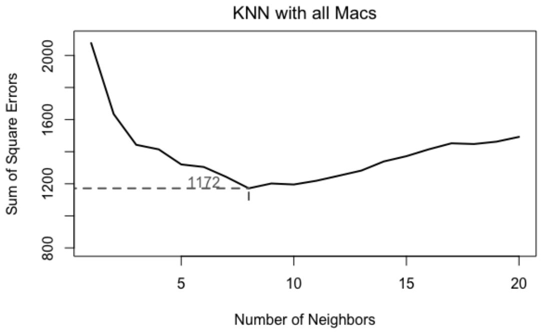


Figure 4: KNN with weighted distance excluding "c0"

Figure 5: KNN with all 7 macs

The weighted KNN model uses 6 of the 7 macs and it excludes "c0" since the model with it performed poorer previously. The KNN with all macs included was also built however it performed worse than the KNN with 6 macs that excluded "c0". This could be an instance of overfitting since the "c0" mac and the "cd" mac have the same position so the KNN with all 7 may have overfit the data because of this redundancy. The best overall model was the weighted KNN model. Intuitively this makes sense. After finding that excluding "c0" was more beneficial than excluding "cd" and after finding that

including both of them was harmful to performance, naturally the weighted KNN was built excluding "c0" from the data.

Model	**Sum of Squared Errors**	**Number of K's**
Without "cd"	**1534.86**	**7**
Without "c0"	**1130.63**	**7**
With both	**1171.53**	**8**
Weighted without "c0"	**1088.04**	**5**

V. Conclusion

The best model for performance was the weighted model that excluded "c0". This is the model that should be implemented if a KNN model was to be implemented from this research. Another important note is that it is extremely likely that any future modeling done on this data should exclude at least one of the redundant position macs. It is clear that by including both overfitting occurs. The strength of this KNN model is that it is computationally effective. The downside is that this KNN model is not transferable to other RTLS's. This research should be used as a jumping off point for either other modeling on this exact Mannheim University data or for modeling techniques that could be applicable in other RTLS data.

VI. References

1. **Nolan, D., & Lang, D. T. (2015). *Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving*. Boca Raton, FL: Chapman & Hall.**

VII. Appendix

1. **All R code included in the R codebook attached entitled "Case_Study#1_QTW.RMD"**