Indoor Wireless Localization with Neural Net

Manoj Basnet
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN USA
mb9g@mtmail.mtsu.edu

Steven Hearne
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN USA
smh9w@mtmail.mtsu.edu

Lian Sang
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN USA
lss3z@mtmail.mtsu.edu

Gregory Wagner
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN USA
gw2t@mtmail.mtsu.edu

***Abstract*—The next generation of GPS-like navigation lies within the concept of modern buildings. With the use of mobile devices and the popular rise of user-friendly applications, indoor wireless localization is a possibility in today's society. By taking advantage of Received Signal Strength (RSS) of WiFi signals and fingerprint database, the existence of GPS-like navigation indoors is in the near future. However, the trouble of indoor localization is that the RSS values on the reference device and target device are not quite the same, or exact; therefore, it has caused errors in localization. To capture and minimize the errors, a method of multilayer network will be used. In order to proceed with this method, fingerprinting several locations that one desires within the indoor space is required. As mentioned before, a reference device is needed; therefore, a WiFi equipped laptop is set up for each location to be fingerprinted. Then a Multi-Layered Perceptron was used after some data manipulation and PCA for better understanding, which help makes decision boundaries. The result shows a perfect accracy of 100% and minor loss, which shows success.**

***Index Terms*—feed-forward, classification, indoor localization, wireless networks**

I. Introduction

Since the GPS has been a very functional tool that provides many beneficial factors for society from advancing technology of auto-pilot vehicles to calculating movements of the Earth's surface, the potential may also be numerous for indoor localization, especially for civilians. Knowing that many buildings in today's populated areas, such as residential houses and commercial zones, are equipped with wireless technology, by taking a step further towards advancing navigation, it could possibly lead to another breakthrough. Imagine being a single tap away from navigating indoors without the help of anyone else except for a turn-by-turn visualization mapping that feeds out. The benefits would be quite substantial in several fields, whether it is for public safety or private investment. With the capability that indoor wireless localization holds, the technology extends the nature of safety by monitoring senior citizens and children from endangerment. As for luxurious use of the localization technology, many commercially-based companies may lure in consumers based on their current location by the user's shopping interests. It is a win-win situation for consumers and merchandisers with the help of indoor localization technology.

II. Background

Although it is unfortunate that today's navigation systems are not meant to be used for indoor environment, it is time that the tech community introduce a new method of wireless localization for inside buildings. Starting with the selection of signal, which is a crucial drawback of GPS for the indoors, there are several signal types that can accommodate indoor localization: common signal types such as FM radio that got used in [1] and the use of visible light in [2], and a potential signal type that could extend GPS signal with pseudolite in [3]. Given that these various types of signal could possibly extend the navigation dream, WiFi signal that is used in [4] could potentially be the best candidate for this project. Since WiFi signals are being used nearly everywhere in our modern society, it is ideally a smart choice considering the fact that a user's device, such as a smartphone or tablet, is already equipped with built-in WiFi-capable hardware. After deciding the systematic signals for wireless localization, having good techniques and methods is the next step. When it comes to the localization of indoor wireless, there is also a selection for the techniques in which that can be done. Therefore, just like selecting a signal type, it can be a bold move to choose any type of wireless method of navigation to create an indoor version of GPS. This is because there are several techniques for wireless indoor localization, which can be found in [5], each with its own set of drawbacks. As for the convenience of measurement issue, mapping methods is required due to the conversion of physical measurements to environmental locations. Selecting an appropriate method of techniques is very pivotal. Depending on the different types of mapping, the techniques vary as well. According to [4], the most representable type of mapping would be geometric mapping due to the complexity of the environment since the specific mapping will require fingerprinting methods.

III. Methods

A. Data

An appropriate step to approach this project would be to obtain some form of data set. Since the study is dealing with WiFi localization, some very important key words (numerical data) such as number of scans, training vectors, location, and wireless access points (APs), etc. are necessary to be trained using neural network. Depending on the Neural Network Model one chooses to exercise, the values previously listed need to be adjusted accordingly. The training data for the Neural Net will need to be set up in a data set that consists of the label with feature vectors. This data set, for example, should look something like [loc 0 - number of choice] and its features should be RSSI values from the APs. The difficult part of this data set is planning the number of locations according to the environment and figuring out the count of APs necessary to satisfy the project. Here is an example vector:

As for raw data, using WifiInfoView.exe program should do the trick. If one is on a Linux machine, then it is necessary to 'wine' to run a Windows executable. Keep in mind that if it involves the use of Jupyter notebook and being on a remote system, then WiFi information being collected is not from ones machine. For the db_wifiinfo.bat and db_wifi_bash.ipynb files, these are for Windows and Linux respectively and will run the .exe program several times to produce scans of any location one chooses. For this, also keep in mind that a robust data set is data that is taken over different periods of time due to the nature of signal interference in APs. Since the tendency for the scans to produce NULL data where a signal has been momentarily lost, it is a good idea to get at least 500 scans per desired location for the Net. In provided examples, the data is taken continuously (1 - max_loc_scans). In this experiment, the data was initially taken and spread into vector files in order to find an average of 10 scans, to deal with some of the signal loss that will inevitably occur. We found that this averaging had a severe effect on the variance the data should have represented to train the network properly. In this case, one should take 500 or more scans at each location moving from location to location, space them out over time, then allow small variance in time for the raw data. By following this method, one should now be able to take all the raw data that is necessary for this study.

As for processing the data that is obtained, ensure that the raw data is set up in location folders. In this case (loc_1_MAXLOC), vector sub folder (vec_1-MAXVEC) exists but is not required, and approximately five to ten scans (dbXX.csv which XX is 1-MAXSCANS) are inside the vector folders. The scans can be allocated (1-MAXSCANS) in each location folder. If data is sorted as in this experiment, reusing the data_to_vector.ipynb to sort the vector files should be an appropriate choice. One should note that clean_loc_data is a bash file for removing vec_X folders from each loc_X folder quickly. One could alter these simple scripting tools to his/her desired setup.

Once the files are correctly set up, one should modify the number of key words inside of builddata.ipynb, then run the script to produce a training set for the Neural Net.

A set of testing data should be taken from each location in about 20 scans, or however many it takes to have at least a few vectors that do not have nan values. These can be sorted in the same way as the training locations.

It should be noted that 3000 scans were taken for the three locations total and only around 1500 provided valid training vectors for the network. Therefore, one can conclude the raw data mechanisms are about 50% efficient.

B. PCA

In our PCA, we are able to get a better understanding of how the Neural Network will view our data and make decision boundaries. The data is read in along with the planned testing locations and we used Singular Value Decomposition to find the variance attributed to each feature. Then, we used the first two features, where approximately 54% of our variance lies and plotted our data along these as basis vectors. In our PCA, we found relatively clear decision boundaries that could be handled by a Multi-layered Neural Network with non-linear activation functions. Our testing data was also included to see if our network was able to group the testing and training data similarly. We found that our network would be able to do so.

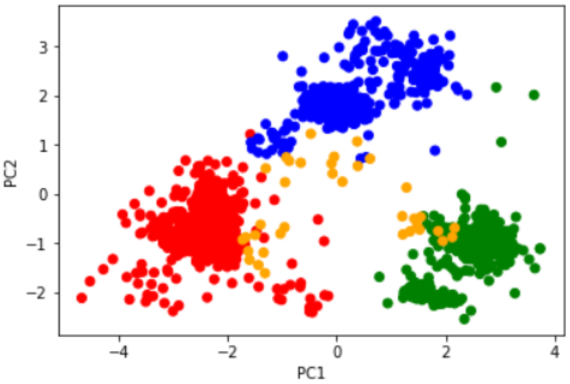


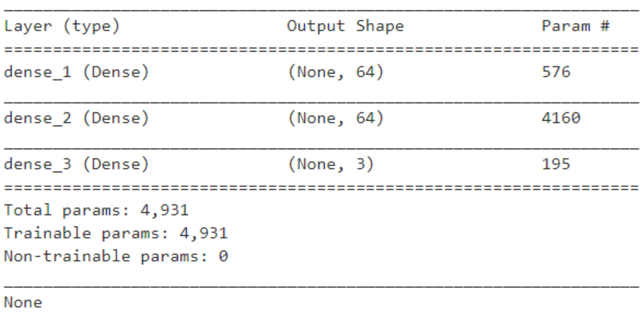
Fig. 1. PCA

C. Network

The neural network needed for the wireless localization is fairly small and simple Multi-Layered Perceptron(MLP) with two hidden layers each with 64 hidden units. The non-linear discriminant functions used for the first and second hidden layer were ReLU and Sigmoid respectively with bias weight initilized to 0.01 for avoiding zero bias. The reason behind using those functions is it seemed to be the most reliable for our data set. Since, we are intersted in classifying 3 locations, the loss funtion would be categorical-cross entropy. Hence, the natural activation function for the output layer would be softmax. A deeper net is always preferred as compared to wider net. That's why MLP with two hidden layers are used in our net. Ideally, any net with a single hidden layer would be able to classify the datasets.

Optimzer Adam is widely used in the state of art applications that's why it became natural choice for us too.

Our input data has eight feature vectors each feature extracted from a shared AP among all three locations.

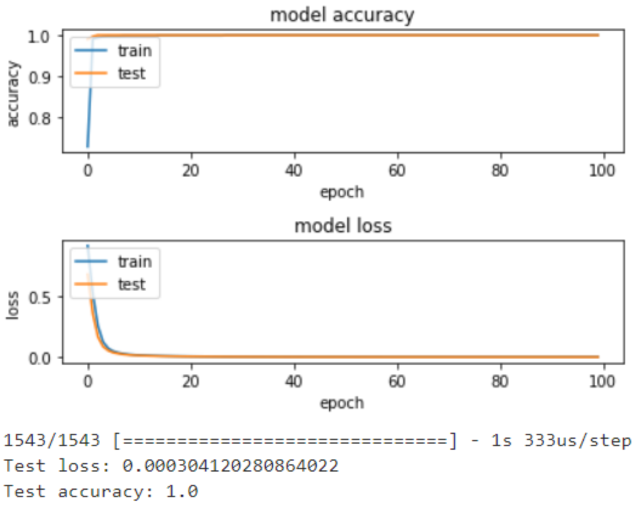


```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 64)                576       
_________________________________________________________________
dense_2 (Dense)              (None, 64)                4160      
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 195       
=================================================================
Total params: 4,931
Trainable params: 4,931
Non-trainable params: 0
_________________________________________________________________
None
```

Fig. 2. Multilayer-Layered Perceptron

IV. Results

100% accuracy and loss of 0.0003041 for both train and validation data were observed within just 5 epochs (27.55 seconds) with an MLP composed of two hidden layers each with 64 hidden units and with the batch size of 56. Ideally, an MLP with a single hidden layer would be able to classify the training data. However, a deeper net is always preferred over the wider one. This net is very efficient and simpler compared to other current state-of-the-art application for localization as depicted in accuracy and loss plot.



```
1543/1543 [==============================] - 1s 333us/step
Test loss: 0.000304120280864022
Test accuracy: 1.0
```

Fig. 3. Accuracy and Loss

V. Discussion

More than 53% of variance accounted for the first two principal components. The dimensionality reduction technique might cost the useful information to be thrown away. As depicted from PCA plot, almost 96% variance accounted for the first six principal components. Therefore, we cannot throw away the first six PCAs. Apart from some outliers, most of the data are clustered around three means (referring to data from three different locations) with different variances (spreading of clusters). If we discard some outliers from each class, we might be able to use MLP with linear activation. However, not a single data should be discarded in order to classify oddly distributed data during the testing. Therefore, for the perfect classification, an MLP with a hidden layer with some non-linear discriminant function such as ReLU or Sigmoid is recommended from PCA analysis.

100% accuracy and loss of 0.0003041 for both train and validation data were observed within just 5 epochs (27.55 seconds) with an MLP composed of two hidden layers each with 64 hidden units and with the batch size of 56. Ideally, an MLP with a single hidden layer would be able to classify the training data. However, a deeper net is always preferred over the wider one. This net is very efficient and simpler compared to other current state-of-the-art application for localization as depicted in accuracy and loss plot.

The real challenge was how the net was going to perform for the new set of test data that was not trained. A total of 36 test data, 14 from location 1, and 11 each from location 2 and location 3 were captured on different days at different times knowing that the temporal variance of RSS would be there. Our net was 100% accurate to distinguish between those locations, provided that the test data was gone for the same pre-processing and curve fitting as the train data.

There are various, considerable factors that might affect our data. One factor is the indoor environment such as room topology because the crowd could cause Wi-Fi signals to be attenuated, reflected, and suffer from multipath fading. That's why from the same AP at different scans, the same RSSI values might differ for the same location. The larger the number of scans, the more variance of RSSI that can be captured. Another factor is Wifi-Module because there is no uniform standard followed by different vendors and also 802.11 protocols do not precisely define how the RSS values should be interpreted. Therefore, there might be discrepancies using different devices for fingerprinting and testing. The solution for this is to train the model with a variety of devices.

During each scan, the number of scanned APs might be different due to various causes as explained above. The first eight strongest RSS signals from eight shared APs among three locations were useful for our data. If the scan missed any one of the shared AP, then the RSS information of that scan would be automatically discarded by our code. In order to resolve this issue, we might use only APs inside the room which are very strong compared to faraway APs. Since the RSS obtained from faraway AP is relatively weak and easily perturbed, it does not have enough contribution towards localization. In this way, the proposed NN can be made more robust and efficient.

References

[1] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "Indoor localization using fm signals," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 1502–1517, 2013.

[2] S. Jung, S. Hann, and C. Park, “Tdoa-based optical wireless indoor localization using led ceiling lamps,” *IEEE Transactions on Consumer Electronics*, vol. 57, pp. 1592–1597, 2011.

[3] C. Kee, D. Yun, H. Jun, B. Parkinson, S. Pullen, and T. Lagenstein, “Centimeter-accuracy indoor navigation using gps-like pseudolites,” *GPS World*, 11 2001.

[4] L. Miao, “Calibration-free wireless indoor localization (cafloc).”

[5] Z. Yang, Z. Zhou, and Y. Liu, “From rssi to csi: Indoor localization via channel response,” *ACM Computing Surveys (CSUR)*, vol. 46, 2013.