Indoor Localization using Graph Neural Networks

Facundo Lezama
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
facundo.lezama@fing.edu.uy

Gastón García González
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
gastong@fing.edu.uy

Federico Larroca
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
flarroca@fing.edu.uy

Germán Capdehourat
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
gcapde@fing.edu.uy

***Abstract*—The topic of indoor localization is very relevant today as it provides solutions in different applications (e.g. shopping malls or museums). We consider here the so-called Wi-Fi fingerprinting approach, where RSSI measurements from the access points are used to locate the device into certain pre-defined areas. Typically, this mapping from measurements to area is obtained by training a machine learning algorithm. However, traditional techniques do not take into account the underlying geometry of the problem.**

We thus investigate here a novel approach: using machine learning techniques in graphs, in particular Graph Neural Networks. We propose a way to construct the graph using only the RSSI measurements (and not the floor plan) and evaluate the resulting algorithm on two real datasets. The results are very encouraging, showing a better performance than existing methods, in some cases even using a much smaller amount of training data.

***Index Terms*—Localization, Graphs, GNN.**

I. Introduction

The great access to smartphones and the growing need of people to stay connected has allowed the development of a wide range of applications that benefit from the indoor localization service.

The devices periodically interact with the access points (APs) of different Wi-Fi networks, so that the user has at her disposal an updated list of the networks available in the premises. In large buildings such as shopping malls, museums, universities, among others, several APs are distributed so that the Wi-Fi service reaches all its corners. In this context, if the information of the power with which the phones receive the APs' signal in their environment is accessed, and since these are fixed, the position of the person could be estimated in real time.

The most common solution for this type of problem, constituting the so-called Wi-Fi fingerprinting approach, is to use classical machine learning (ML) techniques for multiclass classification [1]. The particular building is divided into non-overlapping zones (e.g. classrooms), and the objective is to estimate the corresponding zone based on the power measurements. The input is thus a vector containing the powers measured at a given moment and the target is the corresponding area of the premises. However, these techniques do not take into account the underlying geometry of the problem. To illustrate the importance of considering the structure of the data, suffice to say that it is one of the main reasons behind the success of Convolutional Neural Networks (CNNs) for image and audio processing.

In this work, a novel approach is proposed, using deep learning models for graphs. The idea is that the classifier leverages geometric information through a graph where the nodes are the APs and the edge weights are related to a notion of distance between them. The model is based on the Graph Neural Network (GNN) [2], which basically extend the concept of CNNs for data represented in graphs.

The evaluation of the proposed approach was carried out on two different datasets: MNAV [3] and UjiIndoorLoc [4], surpassing in different cases the performance of previous works, with an Accuracy of 97.2%. In some cases a very good performance was obtained even using a small subset of the data.

The document is organized as follows. Section II briefly presents the literature on indoor localization. In Sec. III we explain the proposed method in detail. Finally, section IV describes and analyzes the experiments carried out before concluding the article in Sec. V.

II. Related Work

The problem of indoor localization is a widely studied topic and different methods and technologies have been used to address it. In [5] technologies such as the use of infrared sensors, ultrasound sensors, cameras and their subsequent processing, radio frequency technologies such as Radio Frequency Identification (RFID) and methods based on Wireless Local Area Network (WLAN) are presented.

Among the techniques based on WLAN, fingerprinting stands out, where the space is divided into zones and with the readings of signal strength between devices and APs, ML models can be trained to classify new measurements in any of the zones [1], [5], [6]. The most used ML method is the K-Nearest Neighbors (KNN) method due to its simplicity and good results. In [7] KNN is used to find the floor a device is on using the Euclidean distance. In [8] a system based on triangulation and KNN is used to obtain the position. Other machine learning techniques used for this problem include Deep Learning [9], Random Forests, Support Vector Machine and Multi Layer Perceptron (MLP) [3].

This work seeks to use GNNs to perform the classification, taking advantage of the spatial structure of the deployment of APs.

III. Proposed Method

A. Spatial Information

The localization problem is basically a classification one: given the RSSI measurements of the n_{AP} APs as received by the device, the objective is to learn how to map these values to the corresponding zone. Let us denote by $\mathbf{X} \in \mathbb{R}^{n_{AP} \times F_{in}}$ one of these measurements, where for instance $F_{in} = 2$

when measurements for both the 2.4 and 5 GHz bands are available. We will use $\mathbf{x}_i \in \mathbf{R}^{F_{in}}$ to indicate the i-th row of $\mathbf{X}$, corresponding to the RSSI measurement from AP i (with a default value of, for instance, -100 dBm in case this particular AP's RSSI was below the sensitivity of the device). Given n_z possible zones, we want to estimate the parameters of a function $\mathbf{\Phi} : \mathbb{R}^{n_{AP} \times F_{in}} \to \{1, \dots, n_z\}$ that minimizes a certain loss over the available training set.

As we mentioned before, the family of functions $\mathbf{\Phi}$ typically chosen (e.g. a Neural Network) does not consider at all the underlying structure of the problem, which is expected to be learned from the training set instead. Here we consider an alternative approach, where the geometric information of the APs is provided *a priori* by means of a graph. This increases the generalization power of the method as well as decreasing the number of samples needed in the training phase to obtain the same performance.

In particular, we define a graph where nodes are the APs and edges are related to the distance between them. If a map with the position of the APs and the characteristics of the scene (floors, walls, etc.) is available, then the graph its easy to build. Given that a complete map was unavailable for both datasets used in this work, we used the RSSI measurements in the training set to define a distance between APs as follows. Given a certain AP_i, we take all instances in the training set which have measurements for AP_i. We then filter out the instances in which the RSSI measurement for AP_i don't surpass a certain (restrictive) threshold. The idea behind this filter is to only consider those instances that were obtained in an area near to the AP_i, this way we can use them to estimate the RSSI measurement that the AP_i would have had when trying to reach the other APs. Taking into account this filtered subset, we estimate the distance between the AP_i and any other AP_j as the mean over this subset of the RSSI measurements corresponding to AP_j. We then repeat this procedure for every AP and define the graph with the edge weights calculated.

Note that in the case when there are measurements for both bands (i.e. $F_{in} = 2$) we will obtain two weights per edge. Although this is easily accommodated for the methods we discuss below, results do not change significantly when using either of them or both, so we will focus on the results of using a single weight per edge (the one corresponding to the 2.4 GHz band). Furthermore, in order to work with positive weights, we have subtracted the minimum RSSI value to all measurements as a pre-processing step. Note that this way AP pairs may be disconnected on the graph (with a weight equal to zero), effectively reflecting they are far apart. Finally, note that the resulting graph is not necessarily symmetric.

B. Graph Neural Network

Equipped with the graph G we built as described in the previous section, we may now consider the localization problem in the context of Graph Signal Processing [10]. Indeed, we may view each RSSI measurement $\mathbf{X}$ as a signal over G: each node i has an associated vector $\mathbf{x}_i$. The objective is thus to classify this graph signal into one of the possible n_z categories. To this end, we will use the framework of Graph Neural Networks (GNNs) [2] which we now briefly present.

GNNs may be regarded as an extension of CNNs to graphs. In this sense, we first have to define convolution on graphs, for which the so-called Graph Shift Operator (GSO) $\mathbf{S} \in \mathbb{R}^{n_{AP} \times n_{AP}}$ is introduced. This is a matrix representation of the graph, which should respect its sparsity (i.e. $\mathbf{S}_{i,j} \neq 0$ whenever there is an edge between nodes i and j). In our work we have used the normalized adjacency matrix, but further examples such as the Laplacian may be used instead.

Computing the matrix product $\mathbf{SX} = \mathbf{Y}$ (with $\mathbf{Y} \in \mathbb{R}^{n_{AP} \times F_{in}}$) we end up with another graph signal that aggregates at each node the information of its neighbors. By writing $\mathbf{S}^K\mathbf{X} = \mathbf{S}(\mathbf{S}^{K-1}\mathbf{X})$ we may see that this way we aggregate the information K hops away. Graph convolution is defined simply as a weighted sum of these K signals (i.e. $\sum_k \mathbf{S}^k\mathbf{X}h_k$, where scalars h_k are the taps of the filter). Notice that we may even change the output dimension by considering an $F_{in} \times F_{out}$ matrix $\mathbf{H}_k$ instead of the scalar taps. A single-layer GNN (or graph perceptron) results of applying a pointwise non-linear function $\sigma(\cdot)$ to this convolution:

$$\mathbf{Y} = \sigma\left(\sum_{k=0}^{K-1} \mathbf{S}^k\mathbf{X}\mathbf{H}_k\right) \tag{1}$$

and a deep GNN is constructed by concatenating several perceptrons.

In order to classify the original signal $\mathbf{X}$ into the possible n_z areas, we have further concatenated the GNN with a fully connected neural network whose output size is precisely n_z. Finally, the softmax function is applied and the maximum value of the result is the predicted zone. Cross entropy was used as the cost function to optimize the parameters. Note that the output of the GNN may be regarded as a node embedding [11] specifically for localization, an interesting by-product which may be further studied.

IV. Experimental Results

In order to be able to compare the obtained results with those from other studies, two datasets were used: MNAV [3] and UJIIndoorLoc [4]. We now briefly describe them and discuss the obtained results.

A. UJIIndoorLoc

This dataset can be found at *Kaggle*[1] and was used as the official dataset of the IPIN2015 competition [4]. This dataset was designed to test WLAN fingerprinting techniques. The data was acquired in three buildings of the University of Jaume I, each with 4 floors or more and covering an area of at least 110,000m2. In total, it has 19,937 training samples and 1,111 validation/test samples acquired 4 months after the training data. More than 20 users and 25 different models of devices were used. To test the proposed method, the 520 features corresponding to the APs were used from a total of 529 features offered by the dataset. The columns *Floor* and *BuildingID* were used as labels, and the remaining user information, longitude and latitude were not taken into account.

[1]https://www.kaggle.com/giantuji/UjiIndoorLoc

TABLE I
CLASSIFIER PERFORMANCE ON EACH BUILDING SEPARATELY USING THE UJIINDOORLOC DATASET.

BuildingID	Floors	Test Accuracy	Instances	Number of APs
0	4	95.5 %	5249	69
1	4	82.4 %	5196	166
2	5	92.1 %	9492	53

Different classifications were explored: by *BuildingID*, by *Floor* filtering the instances by *BuildingID*, and finally by *Floor* using all the dataset. Intuitively, the first classification should be the simplest due to the distance between the different buildings. On the other hand, the most interesting classification is to find the floor where the device is located by looking at the entire dataset.

Note that in this case the positions of the APs are unavailable, highlighting the practical importance of the method to construct the graph we discussed in Sec. III. Furthermore, it was observed that the columns associated with some APs had very few significant values, so they were not taken into account for the construction of the graph. This significantly simplifies the complexity of the graph.

The final structure of the model is very simple. It has two GNN layers with a matrix $\mathbf{H}_k$ of 20 units ($F_{out} = 20$) and a value of $K = 2$. As the output layer a MLP was used with the same size as the number of zones.

Beginning by evaluating the classification by buildings, a very good result was obtained: 99% accuracy. This performance is also obtained with simpler classifiers such as KNN and MLP. It should be noted that only the most relevant 194 APs were used. This allowed us to validate that the GNN was indeed doing the job of classifying in the different zones. The next test was to estimate the floor for each instance within each building. This allowed us to see in more detail the distribution of the instances on the floors of each building without including too much interference from the rest of the buildings. The table I shows the performance of the classifier in each of the buildings, the number of instances and the number of APs used.

Good performance can be seen in two of the buildings with 92.1% and 95.5%. On the other hand, the second building has a performance of 82.4%. Reviewing in detail the accuracy that was obtained in the different floors of the second building, it was observed that the highest number of errors occurred in the floor 0, where only 50% of the instances were correctly classified. We can expect this problem to happen again when classifying the floors using the entire dataset.

Finally, let us discuss the complete classification, including both building and floor. For this, new classes were created combining the values of the *BuildingID* and *Floor* columns, resulting in 13 classes. In this case, for the GNN model an accuracy value of 92.3% was obtained in test, surpassing the performance of the KNN model for which 85.5% was obtained.

The corresponding confusion matrix is shown in Fig. 1. In general terms, it can be seen that the classification is carried out correctly in all the classes except for the one corresponding to one of the floors of the second building.

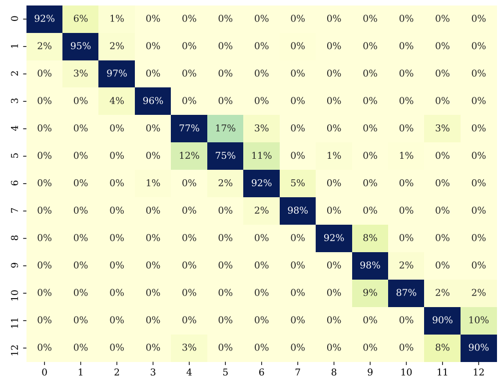

Fig. 1. Confusion matrix obtained when classifying the instances on the different floors of the buildings.

This is consistent with the analysis previously conducted for that particular building.

The results show that the use of GNNs to tackle indoor localization problems is encouraging. A very good accuracy was obtained using a simple model for a dataset that represents a large surface, this means that there is space to improve by exploring more complex GNN models.

B. MNAV

This second dataset was created within the framework of the work [3], which sought to provide an indoor localization system to the *Museo Nacional de Artes Visuales* (MNAV, National Museum of Visual Arts) in Uruguay, using fingerprinting techniques with Wi-Fi. The dataset is available at *Github*[2]. Furthermore, the article includes a map of the museum, including the position of the deployed APs and the 16 areas that were defined.

The dataset has 10,469 measurements from 188 AP addresses. Inside the museum there are 15 APs, each one using both the 2.4GHz and 5GHz bands, thus defining 30 of the 188 features available in the dataset. The rest are APs outside the museum that the devices found while searching for Wi-Fi networks. In this work, only the features corresponding to the APs found within the museum were used, thus the rest of the features were discarded.

The dataset was divided into two sets: train and test, with a ratio of 80-20. The hyperparameters of the model were chosen by cross-validation. We used the same model architecture as in the previous problem to tackle the classification in the MNAV dataset, with a slight difference using $K = 3$. The best results were obtained for a batch size of 8 instances, a learning rate of $1\mathrm{e}^{-3}$ and a weight decay of $1\mathrm{e}^{-4}$.

Taking into account the accuracy as the first performance measure, a value of 97.7% was obtained on the training set and 97.2% on the test set showing not only that the overall performance is good but also that the model generalizes well.

Figure 2 shows how performance varies when using different amounts of fingerprints from the dataset. This analysis is important because as stated by the dataset authors in [3] the

[2]https://github.com/ffedee7/posifi_mnav/tree/master/data_analysis

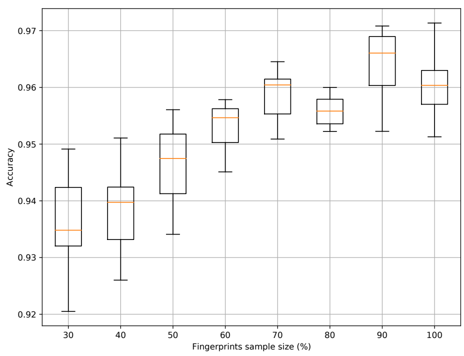

Fig. 2. Accuracy using different amount of fingerprints.

fingerprint gathering stage is time-consuming (roughly 12 hours not considering the deployment and setup of the APs) and represents a non-negligible part of the total cost of the system. It can be seen that a good performance is obtained even using a small proportion of the dataset. Comparing with the reported results in [3] an improvement in the performance is observed. The maximum value of accuracy reported in [3] of 96% was achieved with the proposed method using only 70% of the dataset.

What is most interesting is that the result of this work is obtained with the use of a single method (GNN) while in [3] an assembly of 6 different ML methods is used including an MLP model. This represents an advantage in the training stage when looking for the best hyperparameters.

V. Conclusions

This work sought to explore the usefulness of machine learning techniques in graphs applied to indoor localization, a novel approach for this type of problem. Encouraging results were obtained when compared with the the state of the art, using two different datasets (code is available at *Github*[3]). For the MNAV dataset, an accuracy value of 97.2% was obtained, exceeding the result obtained in [3]. Also a good performance was achieved using a small amount of fingreprints. In the case of UjiIndoorLoc [4], an accuracy of 92% was obtained when using the entire dataset to classify instances in different floors. For the construction of the graphs, a technique was used that takes advantage of the information of the measured RSSI to define the edges. This allows to take into account the geometry of the problem despite not having the complete floor plan and the positions of the APs.

As future work, the plan is to explore other network architectures, analyze the trade-off between performance and number of APs, include temporal information to the model and include the zones information to the graph.

References

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[2] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2019.

[3] A. Bracco, F. Grunwald, A. Navcevich, G. Capdehourat, and F. Larroca, "Museum accessibility through wi-fi indoor positioning," *arXiv preprint arXiv:2008.11340*, 2020.

[4] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems," in *2014 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE, 2014, pp. 261–270.

[5] C. Basri and A. El Khadimi, "Survey on indoor localization system and recent advances of wifi fingerprinting technique," in *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE, 2016, pp. 253–259.

[6] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to wlan user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.

[7] A. Varshavsky, A. LaMarca, J. Hightower, and E. De Lara, "The skyloc floor localization system," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*. IEEE, 2007, pp. 125–134.

[8] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, vol. 2. Ieee, 2000, pp. 775–784.

[9] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with wifi fingerprints using deep learning," in *International Conference Automation*. Springer, 2017, pp. 575–584.

[10] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[11] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy, "Machine learning on graphs: A model and comprehensive taxonomy," *arXiv:2005.03675 [cs.LG]*, 2020.

[3]https://github.com/facundolezama19/indoor-localization-gnn