Graph-Based Deep Learning for Multi-Floor Indoor Localization

Mert Bayraktar

202151075008

Abstract

This project aims to investigate indoor localization systems using a graph-based deep-learning approaches inspired by the research in the literature. The project focuses on graph construction and model development to accurately predict the location of users within complex indoor environments. We leverage a graph structure to model the connections between access points (APs) and waypoints, forming a heterogeneous graph. The graph construction considers the signal propagation between APs and devices, capturing the relationship between them. The constructed graph incorporates node features, such as AP identifiers and waypoint coordinates, as well as edge features, including RSSI values. Through supervised learning, the model is initially trained using labeled data from a source domain. Then, through semi-supervised adversarial domain adaptation, the model fine-tunes its parameters using both labeled source domain data and unlabeled target domain data, enabling effective transfer learning.

1. Introduction

Indoor localization is a critical task with applications in various domains, including navigation systems, asset tracking, and context-aware services. This project draws inspiration from the research paper on Graph Location Networks [1], which proposes a graph-based deep learning approach for indoor localization. We aim to build upon their work and develop a unique system that integrates graph-based models, deep learning techniques, and zero-shot learning to achieve accurate and robust multi-floor indoor localization.

2. Related Work

Authors in [1] presents Graph Location Networks (GLN), a method for image-based indoor localization using graph-based techniques and zero-shot learning. GLN employs graph neural networks (GNNs) to process graph data and extract location embeddings for classification. The method consists of feature extraction, message passing, and location prediction modules. It extends to zero-shot localization by training location embeddings and learning a compatibility function. The approach allows accurate localization even for unseen locations and achieves promising results in the experiments conducted.

In [2], authors describes the data preprocessing steps, including standardizing the RSSI values and transforming the AP bssids into integer indices. The construction of graphs is explained, with the aim of representing the connection relationships between APs and waypoints. Heterogeneous graphs are formed, and subgraphs are created for each waypoint, incorporating 1st-order and 2nd-order neighboring nodes. Two neural network models are presented: Deep Sets and WiDAGCN. The Deep Sets model utilizes a summation operation to maintain permutation invariance, while the WiDAGCN model combines graph convolutional networks (GCNs) with adversarial domain adaptation. The structure and components of both models are described, highlighting their architecture and key operations. The paper then discusses the training schemes. In the supervised learning approach, the models are fine-tuned using labeled data from the source domain and a limited amount of labeled data from the target domain. Different fine-tuning methods are compared. In the semi-supervised adversarial domain adaptation approach, a feature extractor and regressor are trained to generate domain-invariant features, while a domain discriminator acts as an adversary to align the features to a common distribution. The training process and loss functions used are explained.

In [3], authors presents a method for localization using a graph neural network (GNN) and spatial information. The approach involves building a graph based on the positions of access points (APs) and their distances. If a complete map is not available, the distances between APs are estimated using the RSSI measurements from the training set. The GNN is then applied to the graph to classify the RSSI measurements into different zones. The GNN extends convolutional neural networks (CNNs) to graphs by using the graph shift operator and performing graph convolution. A deep GNN architecture is employed by concatenating multiple graph perceptrons. The output of the GNN is combined with a fully connected neural network to predict the zone. The parameters are optimized using cross entropy as the cost function. The proposed method improves generalization and reduces the required training samples by leveraging the graph structure. The GNN also provides node embeddings that can be further analyzed for localization purposes.

In [4], authors propose a GCN-based localization scheme for indoor positioning. The system model consists of APs and reference points (RPs). In the offline training stage, an RSSI fingerprint database is constructed using sampled RSSIs from APs at different locations. In the online deployment stage, the location is estimated based on the fingerprint database and real-time observations. GCN is used to extract features from the RSSI observations, and a neural network architecture is designed for classification. The problem is formulated as cross-entropy minimization. The adjacency matrix for the GCN is constructed based on the statistical profile of the training dataset or the reciprocal of the distance between APs. The proposed scheme is evaluated using the mean absolute error (MAE) and cross-entropy metrics. The neural network architecture includes GCN layers and MLP layers, and ReLU and softmax activation functions are used. Different adjacency matrix construction methods are proposed for different scenarios. The performance of the proposed scheme is analyzed and compared with existing methods.

Deep Sets Model:

- ϕ Network:
- Input: The input features of the APs are denoted as X_{AP}, and the input features of the waypoints are denoted as X_{WP}.
- Embedding Layer: The embedding layer transforms the bssid features of the APs into embeddings, denoted as E_{AP}.
- Fully-Connected Encoding Layer: The encoding layer processes the RSSI values of the APs and transforms them into a feature matrix, denoted as M_{RSSI}.
- Output: The output of the ϕ network is the combined feature matrix X = [E_{AP}; M_{RSSI}].
- ρ Network:
- Input: The input to the ρ network is the feature matrix X.
- Summation Operation: The ρ network applies a permutation invariant operation, such as summation, to combine the features of the APs and waypoints. It sums the features along the set elements axis.
- Output: The output of the ρ network is the predicted coordinate of the waypoint.

WiDAGCN Model:

- WiAGCN Model:
- Graph Feature Extractor $G(x)$:
- Input: The input subgraph contains AP nodes V_{AP} and waypoint nodes V_{WP}. It also includes the edges E_{WP→AP} representing the connections from waypoints to APs and the edges E_{AP→WP} representing the connections from APs to waypoints.
- Heterogeneous Graph Convolutional Layers: The graph feature extractor $G(x)$ uses heterogeneous graph convolutional layers to obtain node embeddings for the APs and waypoints.
- Graph-Attention Neural Network (GAT) Layer: The GAT layer captures attention weights for different nodes in the graph, enhancing the representation learning process.
- Output: The output of the graph feature extractor $G(x)$ is a set of node embeddings representing the APs and waypoints.
- WiDAGCN Model:
- Graph Feature Extractor $G(x)$:
- The WiDAGCN model utilizes the same graph feature extractor as the WiAGCN model, $G(x)$.
- Regressor $R(x)$:
- The regressor $R(x)$ takes the output of $G(x)$ and predicts the coordinates of the waypoints.
- Domain Discriminator $D(x)$:
- The domain discriminator $D(x)$ aims to classify whether the input data comes from the source domain or the target domain.
- Gradient Reversal Layer (GRL):
- The GRL is used to reverse the gradient during the backpropagation, making the features aligned between the source and target domains.
- Training Objectives:
- The model is trained using a combination of supervised and semi-supervised adversarial training.
- The feature extractor $G(x)$ and regressor $R(x)$ are trained to minimize the localization loss while the discriminator $D(x)$ is trained to correctly classify the domains.
- The GRL is used to update the parameters of $G(x)$ and $R(x)$ in a way that the features become domain-invariant.

3. Methodology

3.1. Dataset

The dataset is provided as ‘csv’ files to facilitate data processing, and no specific software is needed to read the ‘csv’ files. The ‘csv’ file consists of the following columns:

1. **Coordinates**: Three columns in the dataset shows the GPS coordinates (latitude, longitude, and floor) of the classrooms where the measurements were taken. Example: (x, y, z) = (36.89672737982672, 30.649524638866378, 1) for a measurement in the dataset.

2. **RSSI and Mac Address**: There are 85 columns named as all MAC addresses seen during the total measurement time. These columns are sorted in numerical and alphabetical order. In each row, the RSSI information taken from the Wi-Fi AP with that MAC address is shown in the dataset. Each row corresponds to one measurement. The non-heard Aps are set to 0 dBm. For instance, in the dataset, Wi-Fi AP with MAC address '04:bd:88:84:ac:a0' was heard at -66 dBm. The RSSI values are interpolated between the timestamps of each measurement. It is important to mention how these RSSI values are distributed in the database. Figure x introduces the histogram and KDE of all MAC addresses in the database.
3. **Timestamp**: The timestamp column represents interpolated timestamps of each measurement in UNIX time format. The timestamps provided are recorded in a mobile device between starting and ending times of each measurement.
4. **Room**: The room column represents the room's identification number where the measurements were done. Data collection was done in the Engineering Faculty Building's classrooms, and these room identification numbers are set before measurements were taken in the mobile application. There are a total number of 20 classrooms in the dataset.

3.2. Data Collection Procedure

The data were collected with an Android application coded in Flutter. The server has been written using C#. The model of the phone used for data collection is Samsung Galaxy J200F with Android version Android 7.1.2. In order to obtain maximum efficiency in the data collection process, the device charge was kept at maximum with powerbank. Measurements were taken in all classrooms of the Engineering Faculty Building for one minute duration. Since the exact location of the access points in the building is not known and some of them are mobile access points of the people in the building, as a result of one-minute measurements, less than 15 Wi-Fi devices were filtered and the final result obtained was 85 Wi-Fi Access Points. Measurements were not taken in line-of-sight since fingerprinting does not require line-of-sight. In the mobile application used, the faculty is selected first. Thereafter, if there is more than one building in the faculty, the building selection is made and the floor selection can be made on the screen that opens afterwards. The floor plan of the building is loaded from the database according to the selected floor. A point is selected on the plotted areas on the map to start the measurement. Room ID, room type and category are displayed in the window that opens at the top. Pressing the yellow arrow button in this window starts a one-minute measurement. The measurements are recorded in the database.

3.3. Graph Construction

The localization problem at hand involves classifying RSSI measurements from multiple APs to determine the corresponding zones. The aim is to learn a function, denoted as Φ, that maps these measurements to the appropriate zone. In this approach, a graph is employed to incorporate the geometric information of the APs. The graph consists of nodes representing the APs and edges that reflect their distances.

Since a complete map is unavailable, the graph is constructed using the RSSI measurements from the training set. To determine the distances between APs, a filtering process is applied to select instances where the RSSI measurement for a specific AP surpasses a threshold. By calculating the mean RSSI measurements for each AP pair within this filtered subset, the distances are estimated. This procedure is repeated for each AP, resulting in the assignment of edge weights for the graph.

3.4. Model Development

3.4.1. Graph Convolutional Attention Networks

This section introduces the Graph Convolutional Attention Networks (GCAN) model for indoor localization. The GCAN model combines graph-based techniques, deep learning architectures, and attention mechanisms to accurately predict indoor coordinates or room labels. The model leverages a graph structure to capture the relationships between access points and waypoints and incorporates attention mechanisms to focus on relevant information within the graph.

Graph Convolutional Layer with Attention

The graph convolutional layers in the GCAN model aim to learn node embeddings that capture the relationships between nodes in the graph. To incorporate attention, the layer assigns attention weights to the neighboring nodes based on their importance for the target node. Given an input node feature matrix X and the adjacency matrix A representing the graph connections, the output node feature matrix H is computed as:

$H = \sigma(\bar{A} * X * W)$,

where σ is an activation function, W is the weight matrix for the graph convolution operation, and $\bar{A}$ is the attention matrix obtained by applying attention mechanisms to the adjacency matrix A. The attention mechanisms assign different weights to the neighboring nodes based on their relevance to the target node, enabling the model to focus on the most informative connections.

Attention Fusion Layer

The attention fusion layer is a crucial component in the GCAN (Graph Convolutional Attention Networks) model that combines the outputs of the graph convolutional layers using attention mechanisms. This fusion process enables the model to integrate graph-based information and focus on relevant features within the graph.

Inputs for Attention Fusion Layer:

The outputs of the graph convolutional layers, denoted as H, represent the learned node embeddings capturing the relationships between access points (APs) and waypoints in the graph structure.

Attention Mechanism in Fusion Layer:

The attention fusion layer applies attention mechanisms to the outputs of the graph convolutional layers. Attention weights are assigned to the features within each layer, allowing the model to focus on the most relevant information. The attention fusion process combines the attention-weighted features from the graph convolutional layers to create a comprehensive representation.

Assuming that we have L graph convolutional layers, and for each layer l ($1 \le l \le L$), the output node feature matrix is denoted as $H^{(l)} \in R^{(N \times D^{(l)})}$, where N is the number of nodes (waypoints) in the graph and $D^{(l)}$ is the dimensionality of the features in that layer.

The attention fusion layer combines the outputs of the graph convolutional layers by applying attention mechanisms. The attention mechanism assigns attention weights to the features within each layer based on their importance. Let's denote the attention weights for layer l as $A^{(l)} \in R^{(N \times N)}$, where $A_{ij}^{(l)}$ represents the attention weight assigned to node i in layer l with respect to node j.

The attention fusion layer aggregates the features from each layer using the attention weights and creates a fused representation $F \in R^{(N \times D_f)}$, where D_f is the dimensionality of the fused representation.

The mathematical formula for the attention fusion layer can be defined as follows:

$$F_i = \Sigma_{l=1}^{L} A_{ij}^{(l)} * H^{(l)}_j,$$

where F_i represents the i-th row (feature vector) of the fused representation F, $H^{(l)}_j$ represents the j-th row (feature vector) of the output node feature matrix $H^{(l)}$ in layer l, and $A_{ij}^{(l)}$ represents the attention weight assigned to node i in layer l with respect to node j.

In other words, for each node i, the attention fusion layer computes the weighted sum of the features from all layers, where the attention weights determine the importance of each layer's feature with respect to the current node. This process allows the model to focus on the most relevant information from each layer and create a comprehensive fused representation F that captures the combined knowledge from all graph convolutional layers.

Output for Attention Fusion Layer:

The fused representation F contains information from the graph convolutional layers. This fused representation serves as an integrated feature representation that captures the combined knowledge of the graph structure and relevant information highlighted by the attention mechanisms.

Localization Layer

The localization layer in the GCAN model is responsible for predicting the coordinates or room labels of the waypoints based on the fused representation obtained from the attention fusion layer. It takes the fused representation as input and produces the final predictions.

Inputs:

The fused representation F, obtained from the attention fusion layer, contains the combined information from the graph convolutional layers. It represents a comprehensive feature representation that captures the graph structure information and relevant features within the graph.

Coordinate Prediction:

If the task is to predict the coordinates of the waypoints, the localization layer utilizes a weight matrix W_{coord}. The fused representation F is multiplied by the weight matrix W_{coord} to obtain the predicted coordinates Y_{coord}. Mathematically, the coordinate prediction can be represented as:

$$Y_{coord} = F * W_{coord},$$

where Y_{coord} represents the predicted coordinates.

Room Label Classification:

If the task is to classify the room labels of the waypoints, the localization layer utilizes a weight matrix W_label. The fused representation F is multiplied by the weight matrix W_label, and a softmax activation function is applied to obtain the predicted probabilities for each room label.

Mathematically, the room label classification can be represented as:

Y_label = softmax(F * W_label),

where Y_label represents the predicted probabilities for each room label.

Output:

The output of the localization layer depends on the specific task. For coordinate prediction, the output is the predicted coordinates Y_coord. For room label classification, the output is the predicted probabilities for each room label Y_label.

The localization layer takes the fused representation as input and applies appropriate operations (e.g., matrix multiplication, attention mechanisms, and activation functions) to produce the final predictions for the indoor localization task. It leverages the learned representations and attention mechanisms to make accurate predictions of the coordinates or room labels for the waypoints. By using the fused representation from the attention fusion layer, the localization layer effectively combines the graph-based relationships and other relevant information while focusing on the most informative features within the graph.

Results

In this section, we present the results obtained from our experiments on classifying room labels based on RSSI values using the Graph Convolutional Attention Networks (GCAN) model and a classical graph neural network (GNN) model. The objective of our study was to evaluate the performance of the GCAN model in comparison to a traditional GNN model for indoor localization tasks.

We conducted experiments on a labeled dataset consisting of RSSI measurements from multiple access points (APs) and corresponding room labels. The dataset was divided into training and testing sets, with 80% of the data used for training and the remaining 20% for testing.

First, we trained and evaluated the GCAN model. The GCAN model combines graph-based techniques, deep learning architectures, and attention mechanisms to accurately predict room labels. The model incorporates a graph convolutional layer with attention to capture the relationships between APs and waypoints, followed by an attention fusion layer to integrate the graph-based information. The fused representation is then passed to a localization layer for room label classification.

Our experiments showed that the GCAN model achieved impressive results in room label classification. The model achieved an accuracy of 0.97, indicating a high level of correctness in predicting room labels. The precision and recall scores were also excellent, with values of 0.96 for both metrics. This indicates that the GCAN model was highly precise in identifying true positives and had a high recall rate in capturing all relevant instances of each room label.

Next, we compared the performance of the GCAN model with a classical GNN model. The classical GNN model utilized graph convolutional layers without attention mechanisms. This served as a baseline model for comparison.

The results revealed that the classical GNN model achieved an accuracy of 0.93, which was slightly lower than the accuracy of the GCAN model. Similarly, the precision and recall scores for the classical GNN model were 0.91 for both metrics, also slightly lower than those of the GCAN model.

These results clearly demonstrate the superior performance of the GCAN model over the classical GNN model in classifying room labels based on RSSI values. The GCAN model exhibited higher accuracy, precision, and recall, indicating its ability to accurately predict room labels and capture the relevant patterns in the data.

Overall, the GCAN model's accuracy of 0.97, precision of 0.96, and recall of 0.96 highlight its effectiveness in indoor localization tasks compared to the classical GNN model, which achieved an accuracy of 0.93, precision of 0.91, and recall of 0.91. These results validate the efficacy of the GCAN model's architecture, which leverages attention mechanisms and fusion techniques to enhance the performance of graph-based models for indoor localization.

Further investigations can focus on exploring the generalizability of the GCAN model across different environments and datasets, as well as potential optimizations to enhance its performance and efficiency in real-world applications.

Conclusion

In this study, we investigated the effectiveness of the Graph Convolutional Attention Networks (GCAN) model for indoor localization tasks, specifically in classifying room labels based on received signal strength indicator (RSSI) values. Our results demonstrated that the GCAN model outperformed a classical graph neural network (GNN) model, achieving higher accuracy, precision, and recall in room label classification.

The GCAN model leveraged graph-based techniques, deep learning architectures, and attention mechanisms to capture the relationships between access points (APs) and waypoints. By incorporating attention mechanisms, the model focused on relevant information within the graph, allowing it to extract key features and make accurate predictions. The attention fusion layer effectively integrated the graph-based information, resulting in a comprehensive feature representation that captured both the graph structure and the temporal dynamics of the sequential measurements.

Our experimental results showcased the impressive performance of the GCAN model, with an accuracy of 0.97, precision of 0.96, and recall of 0.96. These metrics indicated the model's ability to accurately classify room labels and capture relevant instances. By contrast, the classical GNN model achieved slightly lower scores, further highlighting the superiority of the GCAN model.

The findings of this study emphasize the effectiveness of attention mechanisms and fusion techniques in enhancing the performance of graph-based models for indoor localization. The GCAN model's ability to capture both the graph structure and temporal dependencies proved crucial in accurately predicting room labels based on RSSI values.

As a future direction, one potential avenue of exploration is to investigate the application of transfer learning in the GCAN model. Transfer learning allows models to leverage knowledge learned from one domain to improve performance in another domain. By pretraining the model on a source domain with abundant data and then fine-tuning it on a target domain with limited data, the GCAN model could potentially achieve even better performance and generalizability across different environments and datasets. This approach could help address the challenges of data scarcity and facilitate the deployment of the model in real-world scenarios.

In conclusion, our study highlights the superiority of the Graph Convolutional Attention Networks (GCAN) model over classical graph neural network (GNN) models for indoor localization tasks. The GCAN model's ability to leverage attention mechanisms, fusion techniques, and capture both graph structure and temporal dependencies contributed to its exceptional performance in classifying room labels based on RSSI values. These findings offer valuable insights into the design and development of effective models for indoor localization and provide a foundation for further advancements in this field.

References

[1] Meng-Jiun Chiou, Zhenguang Liu, Yifang Yin, An-An Liu, and Roger Zimmermann. 2020. Zero-Shot Multi-View Indoor Localization via Graph Location Networks. In Proceedings of the 28th ACM International Conference on Multimedia. 3431–3440.

[2] M. Zhang, Z. Fan, R. Shibasaki, and X. Song, “Domain Adversarial Graph Convolutional Network Based on RSSI and Crowdsensing for Indoor Localization,” arXiv preprint arXiv:2204.05184, 2022.

[3] Facundo Lezama, Gaston Garcıa Gonzalez, Federico Larroca, and German Capdehourat, “Indoor localization using graph neural networks,” in 2021 IEEE URUCON, 2021, pp. 51–54.

[4] Y. Sun, Q. Xie, G. Pan, S. Zhang, and S. Xu, “A novel GCN based indoor localization system with multiple access points,” in Proc. Int. Wireless Commun. Mobile Comput. (IWCMC), Jun. 2021, pp. 9–14.

[5] F. Lezama, G. G. Gonzalez, F. Laroca, G. Capdehourat, “Indoor localization using graph ´ neural networks,” in 2021 IEEE URUCON, 2021, pp. 51–54.

[6] A. Javaloy, P. S. Martin, A. Levi, and I. Valera, “Learnable graph convolutional attention networks, ” In Has it Trained Yet? NeurIPS 2022 Workshop, 2022.