Graph-Based Deep Learning for Multi-Floor Indoor Localization

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.

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.

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.

Methodology

Graph Construction: In this project, the graph construction step is crucial for capturing the relationships between access points (APs) and waypoints. The graph is represented as a heterogeneous graph consisting of two types of nodes: AP nodes (V_AP) and waypoint nodes (V_WP). The edges (E) in the graph include AP-to-waypoint edges (E_WP→AP) and waypoint-to-AP edges (E_AP→WP).

To construct the graph, the features of the nodes and edges are defined. The node features (F_V) contain the RSSI (Received Signal Strength Indicator) values and bssid (basic service set identifier) identifiers. These features represent the signal strengths of different APs and enable the identification of important APs.

The edge features (F_E) capture the RSSI values between connected APs and waypoints. These values reflect the signal propagation between the APs and devices, forming the connection relationship in the graph. The graph construction process involves creating subgraphs for each waypoint. The subgraphs are constructed by selecting the neighboring nodes up to a certain order, ensuring that the complexity of the graph is reduced.

Node Types:

- **AP Nodes (V_AP):** These nodes represent the access points in the indoor environment. Each AP node is associated with a specific bssid (basic service set identifier), which serves as a unique identifier for the AP. The bssid is transformed into an embedding or encoding to capture the characteristics of the AP.

- **Waypoint Nodes (V_WP):** These nodes represent the waypoints in the indoor environment. Waypoints are specific locations that need to be localized accurately. The waypoint nodes can be labeled with known coordinates (in the source domain) or have unknown coordinates (in the target domain).

Edge Types:

- AP-to-Waypoint Edges (E_WP→AP): These edges represent the connections from APs to waypoints. They capture the signal propagation and strength between the APs and the waypoints. The edge features for these connections include the RSSI (Received Signal Strength Indicator) values, which indicate the signal strength received from the AP.
- Waypoint-to-AP Edges (E_AP→WP): These edges represent the connections from waypoints to APs. They reflect the relationship between waypoints and the APs they are associated with. The edge features for these connections can also include RSSI values, representing the signal strength from the waypoint to the AP.

Graph Structure:

The constructed graphs are heterogeneous graphs with multiple types of nodes and edges. They are represented as G = (V, E, F_V, F_E), where V represents the set of nodes (V = {V_WP, V_AP}), E represents the set of edges (E = {E_WP→AP, E_AP→WP}), F_V represents the node features, and F_E represents the edge features.

It's important to note that the graph construction process involves creating subgraphs for each waypoint, considering neighboring nodes up to a certain order (such as 1st-order or 2nd-order neighbors). The subgraph construction focuses on selecting relevant nodes and maintaining the connectivity between them.

Model Development:

The second component focuses on developing a graph-based deep learning model for indoor localization. Building upon the Deep Sets model and WiDAGCN model proposed in [2], a novel model architecture is designed. The model comprises feature extraction modules, graph convolutional layers, and prediction modules. The feature extraction modules leverage embedding layers to represent bssid features, while the graph convolutional layers capture the deep graph features using Graph Convolutional Networks (GCNs). The prediction modules use regression techniques to predict the coordinates of the waypoints.

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.

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.