Dhirubhai Ambani Institute of Information and Communication Technology

LEARNING GRAPH-NEURAL NETWORKS FOR SEMANTIC MAPPING

Supervisor : Prof. Tapas Maiti
Prepared By : Jigar Shekhat - 202211004

PROBLEM STATEMENT

- Indoor environments offer numerous significant problems for mobile robots, such as appropriate and **precise navigation**, **Dynamic Environment Adaptation**, and **Localization Accuracy**, among others.

- These issues can be solved by adding semantic mapping, resulting in efficient navigation, contextual understanding, task execution, environment adaption, autonomy, adaptability, and accurate localisation.

OBJECTIVE

- Our objective is to utilize semantic mapping to enhance node classification performance by **Graph Convolutional Networks (GCNs)**.
- By integrating semantic information into the GCN model, the objective is to improve effectiveness of node classification tasks by considering the context and meaning associated with the nodes in the graph.
- This can enable more informed and accurate classification decisions.

WORK

- **Dataset - COLD**
- **Graph Sum-Product Network** :
  - GraphSPN is a probabilistic graphical model that extends Sum-Product Networks (SPNs) to handle graph-structured data. GraphSPN extends this capability to graphs, allowing for modeling and reasoning over complex relational data.
- **Graph Convolutional Network**
  - GCN is a deep learning model designed for processing graph-structured data. It extends convolutional neural networks (CNNs) to operate on graph data by leveraging the neighborhood relationships between nodes.

COLD - DATASET

- **Cognitive rObot Localization Database (COLD)**, a large database of localization, sensory information and local environment representations (place scans), as well as topological maps, completely annotated with semantic place categories.
- There are 100 sequences in total, collected by a mobile robot in three buildings at three different locations:
  - Stockholm (40 seqs)
  - Freiburg (26 seqs)
  - Saarbrucken (34 seqs)

COLD

COLD dataset divided into multiple datasets as listed below:

- **COLD-PlaceScans** - a dataset of place scans.
- **COLD-TopoMaps** - a dataset of topological maps.
- **COLD-Meta** - a dataset of metadata, that is, robot maneuver paths, streams of localizations and annotations, SLAM maps.
- **COLD-Images** - a dataset of image frames from multiple on-board cameras.

TopoMaps Dataset

https://www.coldb.org/site/topo_map_dataset/

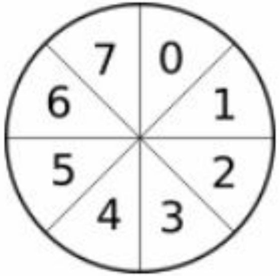
Characteristics of TopoMaps

- **nodes.dat :**

Node_ID	Placeholder	X(m)	Y(m)	label	Views
int	boolean	float	float	string	8 views

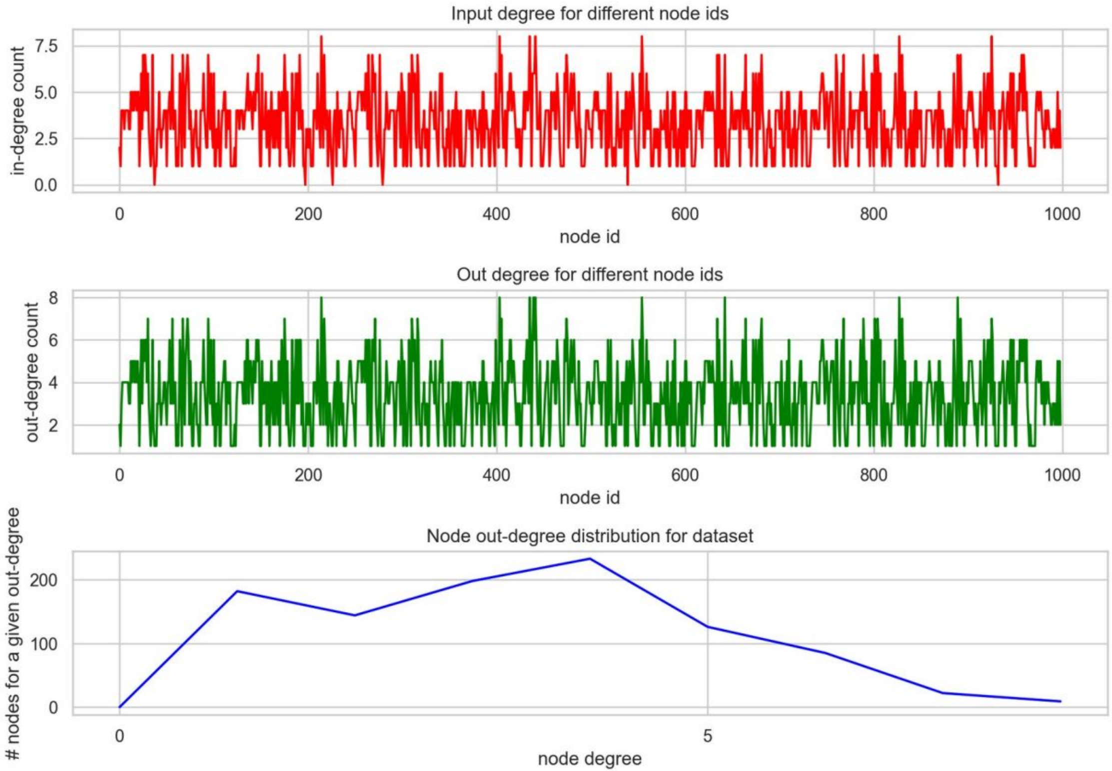
- **views :**

neighbor node ID	affordance	view number
int	float	int



8 views

EDA

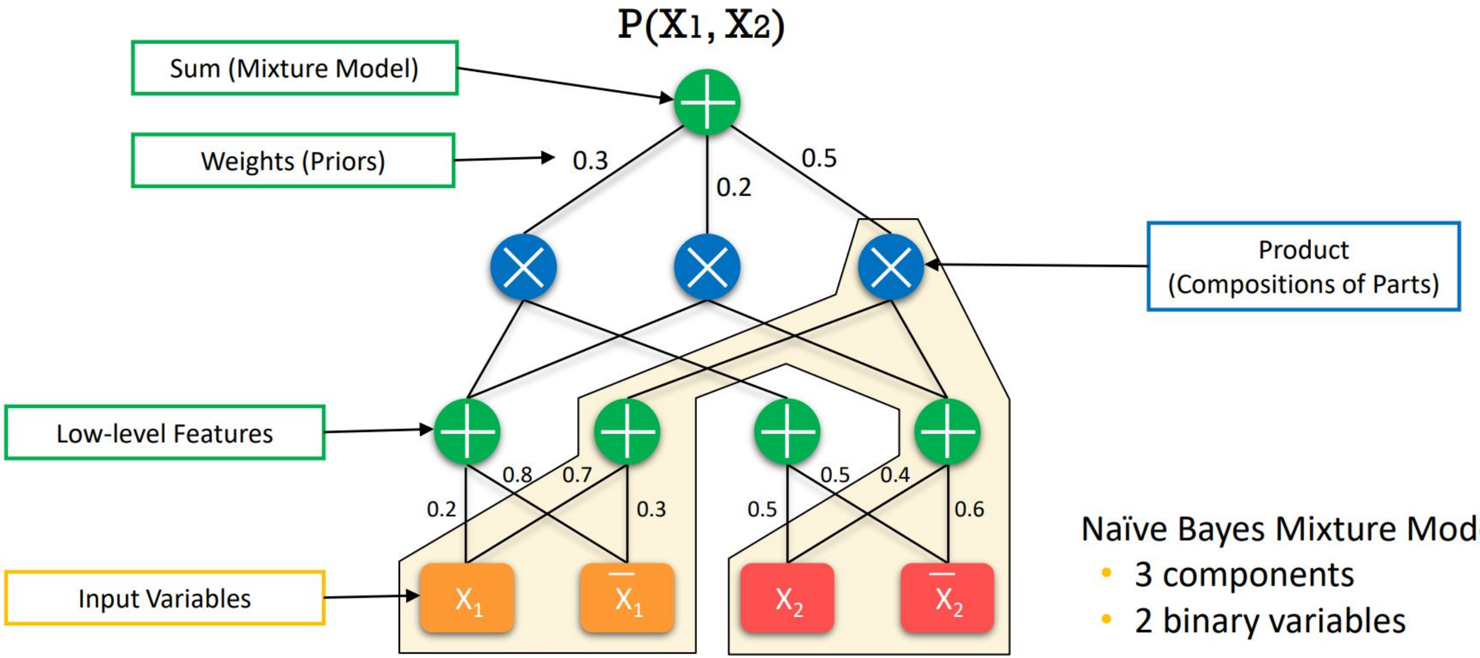


Input degree for different node ids
in-degree count
7.5
5.0
2.5
0.0
0
200
400
600
800
1000
node id
Out degree for different node ids
out-degree count
8
6
4
2
0
200
400
600
800
1000
node id
Node out-degree distribution for dataset
nodes for a given out-degree
200
100
0
0
5
node degree

SPN

- **Sum-Product Network** is a probabilistic graphical model that provides an efficient and tractable representation for probability distributions over structured data.
- SPNs are based on a hierarchical structure of sum and product nodes, which enables efficient inference and learning.
- The sum nodes compute weighted sums of their children, while the product nodes compute element-wise products.
- This hierarchical structure allows for efficient computation of marginal probabilities and inference tasks.

Example of SPN



Naïve Bayes Mixture Model

- 3 components
- 2 binary variables

Zheng, Kaiyu, Andrzej Pronobis, and Rajesh Rao. "Learning graph-structured sum-product networks for probabilistic semantic maps." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

GraphSPN

- **Graph Sum-Product Network** is a type of probabilistic graphical model that extends the Sum-Product Network (SPN) framework to handle graph-structured data.
- GraphSPN represents a probability distribution over a set of graphs using a hierarchical structure of sum and product nodes, where the nodes operate over local subgraphs.
- One of the main advantages of GraphSPN is its ability to handle **variable-sized graphs** and to capture the dependencies and patterns within them.
- This makes it suitable for a wide range of applications.

GNN

- Graph Neural Network is a type of deep learning model specifically designed to process and analyze graph-structured data. It extends traditional neural networks to handle the complexities and dependencies present in graphs.
- each node in the graph is associated with a feature vector, representing its attributes or properties in GNN.
- three popular GNNs :
  - Graph Convolutional Network (GCN)
  - GraphSAGE
  - GAT (Graph Attention Network)

GCN

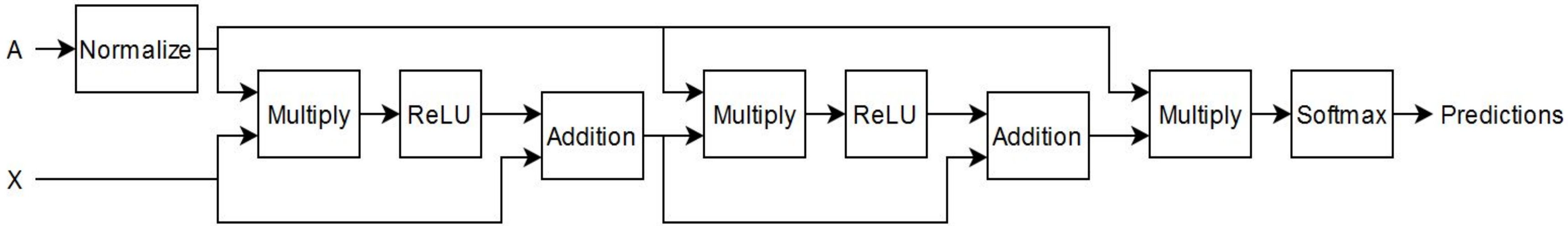
- convolutional neural networks (CNNs) to operate on graphs instead of grid-structured data like images.

$$Z_{l+1} = \sigma_l\left(\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2}Z_l W_l\right) + Z_l$$

  - σ_l is the activation function.
  - $Z_1 = X$.
  - W_l is the weight matrix for the multiplication.
  - $A' = A + I_N$ is the adjacency matrix of graph G with added self-connections. I_N is the identity matrix.
  - D' is the degree matrix of A'

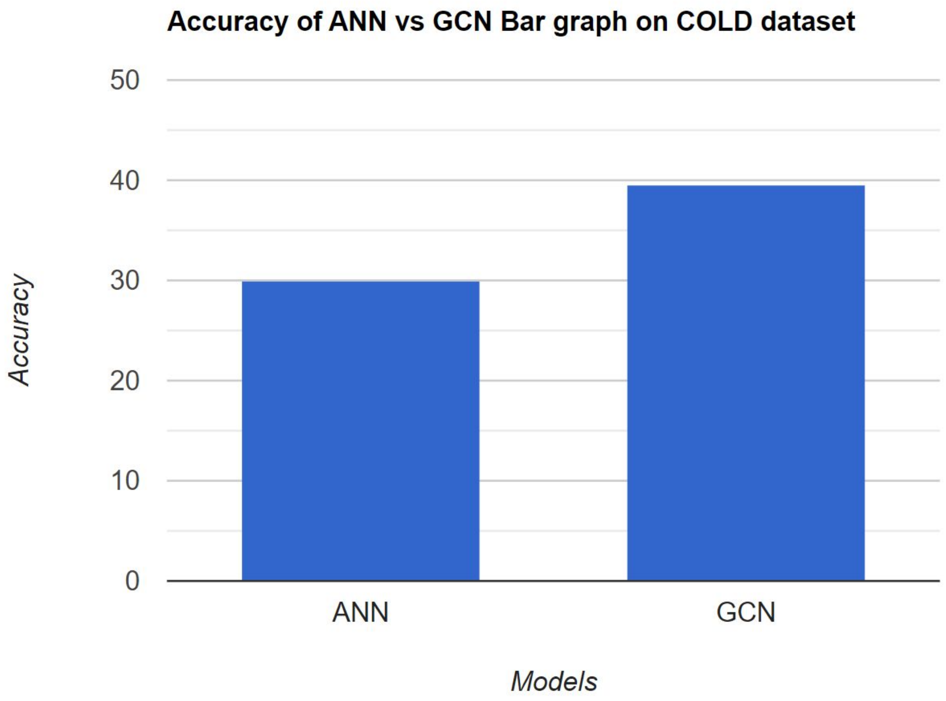
- $\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2}$ normalized adjacency matrix of the graph.

GCN



https://in.mathworks.com/help/deeplearning/ug/node-classification-using-graph-convolutional-network.htm
l

RESULT


Accuracy of ANN vs GCN Bar graph on COLD dataset
50
40
30
20
10
0
Accuracy
ANN
GCN
Models

CONCLUSION

- We examined the effectiveness of graph convolutional networks (GCNs) on the COLD dataset, which has 22 distinct labels. The GCN model classified the nodes into their respective labels with an accuracy of 39.5%. This suggests that GCNs are a potential method for node classification tasks in challenging datasets like COLD, where typical machine learning models have difficulty capturing connections between nodes.

FUTURE WORK

- We only classify the nodes(labels) but now We have to predict floor and building. For that, I already made changes in dataset like adding labels about floors and buildings.
- The COLD dataset we are using right now has only undirected graphs. As a result, we will begin working on a **real-time dataset** that will provide us with a wide range.

REFERENCES

1. Zheng, Kaiyu, Andrzej Pronobis, and Rajesh Rao. "*Learning graph-structured sum-product networks for probabilistic semantic maps*." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.
2. Zheng, Kaiyu, and Andrzej Pronobis. "*From pixels to buildings: End-to-end probabilistic deep networks for large-scale semantic mapping*." 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019.
3. Zhang, Si, et al. "*Graph convolutional networks: a comprehensive review.*" Computational Social Networks 6.1 (2019): 1-23.
4. Chen, Ming, et al. "*Simple and deep graph convolutional networks.*" International conference on machine learning. PMLR, 2020.
5. Zheng, Kaiyu. "*Learning Large-Scale Topological Maps Using Sum-Product Networks.*" arXiv preprint arXiv:1706.03416 (2017).

THANKS!

CNN Accuracy

```
Epoch 95/100
121/121 [==============================] - 1s 6ms/step - loss: 0.3095 - accuracy: 0.9005 - val_loss: 0.3133
996
Epoch 96/100
121/121 [==============================] - 1s 6ms/step - loss: 0.1779 - accuracy: 0.9421 - val_loss: 0.3451
788
Epoch 97/100
121/121 [==============================] - 1s 6ms/step - loss: 0.1963 - accuracy: 0.9335 - val_loss: 0.2988
001
Epoch 98/100
121/121 [==============================] - 1s 6ms/step - loss: 0.1799 - accuracy: 0.9413 - val_loss: 0.2962
136
Epoch 99/100
121/121 [==============================] - 1s 6ms/step - loss: 0.1657 - accuracy: 0.9476 - val_loss: 0.2732
204
Epoch 100/100
121/121 [==============================] - 1s 6ms/step - loss: 0.1688 - accuracy: 0.9464 - val_loss: 0.3460
835
```

In [93]:

```
# Evaluate the model on the test set
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print('Test accuracy:', accuracy)
```

```
Test accuracy: 0.8975853323936462
```

RESULT

- The GCN model achieved an accuracy of **39.5%**, while the ANN achieved an accuracy of **30%**. This suggests that the GCN model was better suited to this particular task of node classification, as it was able to capture higher-order interactions between nodes in the graph more effectively than the ANN. One possible reason for the superior performance of the GCN is that it was able to leverage the structure of the graph to improve its predictions. The ANN treats each node as an independent data point and does not explicitly model the relationships between nodes.