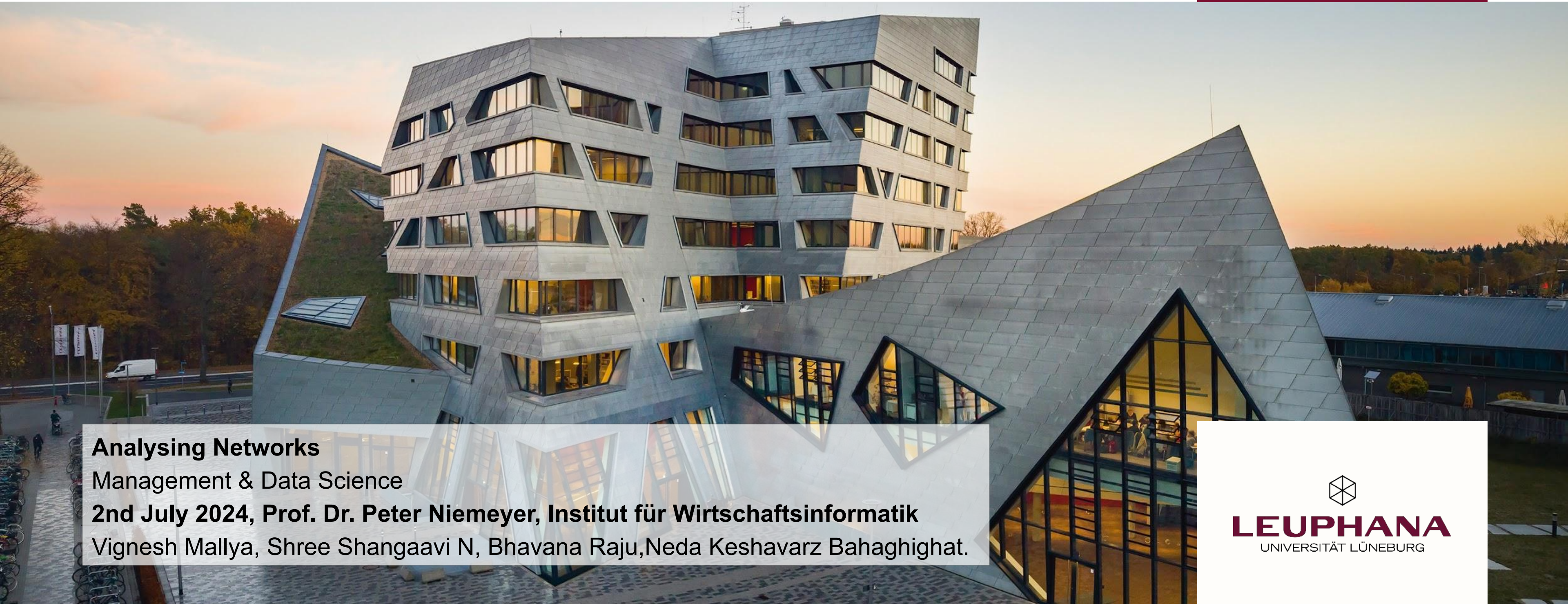
SEMI-SUPERVISED CLASSIFICATION WITH **GRAPH CONVOLUTIONAL NETWORKS**

Thomas N. Kipf,Max Welling



Analysing Networks

Management & Data Science

2nd July 2024, Prof. Dr. Peter Niemeyer, Institut für Wirtschaftsinformatik

Vignesh Mallya, Shree Shangaavi N, Bhavana Raju,Neda Keshavarz Bahaghighat.

LEUPHANA
UNIVERSITÄT LÜNEBURG

Table of contents

INTRODUCTION

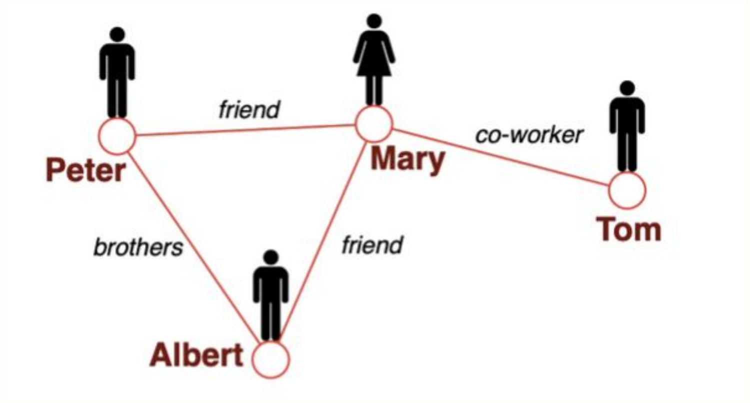
- ❖ Graph embeddings transform graph data into lower-dimensional spaces while preserving structural properties.

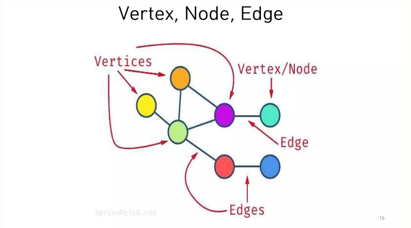
- ❖ Traditional node classification methods assume connected nodes share similar labels, which is limiting.

- ❖ This paper introduces Graph Convolutional Networks (GCNs)
  - ➢ GCNs leverage neural networks to incorporate graph structure directly.
  - ➢ GCNs learn hidden representations that encode local graph structure and node features.
  - ➢ This approach enhances accuracy and efficiency over conventional methods.

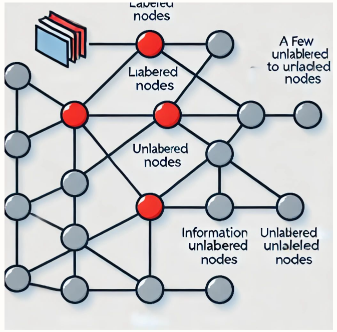
What are GCNs?

- ➔ GCNs are a type of neural network specifically designed to work with data structured as graphs.
- ➔ A graph consists of nodes (or vertices) and edges (links between nodes).
- ➔ Each node has features (like attributes or properties) represented in a matrix X.
    - ◆ For example, in a social network, features could include age, gender, and number of friends.





Semi-Supervised Learning in GCNs

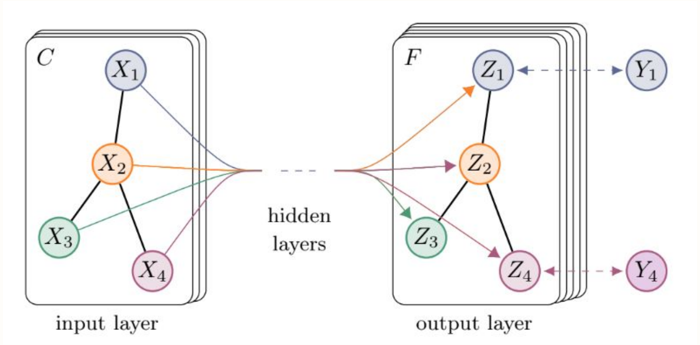

Labeled nodes
Liabered nodes
A Few unlablered to urladed nodes
Unlabered nodes
Information unlabered nodes
Unlabered unlaleled nodes

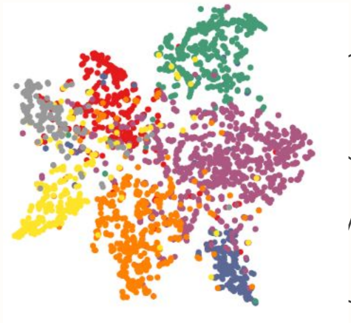
- limited number of labeled nodes
- larger number of unlabeled nodes

Efficiently handle both

How does it work?

➔ Learning Hidden Representations
➔ Conditioning on Data
➔ Performance:
  - ◆ Scalable and efficient
  - ◆ Validated on datasets like citation networks and Knowledge graphs





Efficient way to apply neural networks to graphs

★ GCNs introduce an efficient way to apply convolutional neural networks to graphs by spectral graph theory.

This method integrates the graph structure into the learning process without needing additional regularization terms

★ Advantages:
- handle large graphs
- better performance

Localized first-order approximation of spectral graph convolutions

- Spectral Graph Convolutions apply convolutional neural network techniques to graph-structured data by transforming the data into the frequency domain.
- Localized Approximation:

  Uses a first-order approximation to make computations feasible and efficient on large graphs

- Efficiency:

  Allows the model to be both scalable and effective

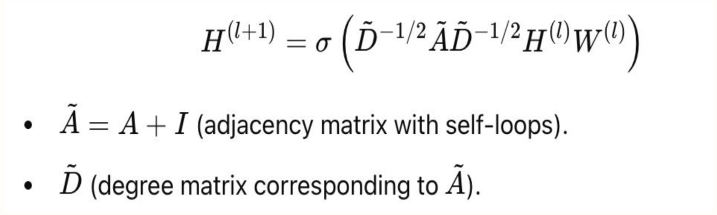
Categorizing the Paper in the Big Picture of Graph Embeddings

Contextualizing within Traditional Graph Embedding Approaches:

- Traditional methods: Spectral embeddings, random walks (e.g., DeepWalk, node2vec), matrix factorization.
- Kipf and Welling's approach: Graph Convolutional Networks (GCNs) that directly operate on graph-structured data using convolutional neural networks adapted for graphs.

Innovations:

1. **First-Order Spectral Graph Convolutions:**
   - Localized first-order approximation of spectral graph convolutions.
   - Simplifies spectral convolution operation, making it computationally feasible and scalable for large graphs using the graph Laplacian.
2. **Layer-Wise Propagation Rule:**

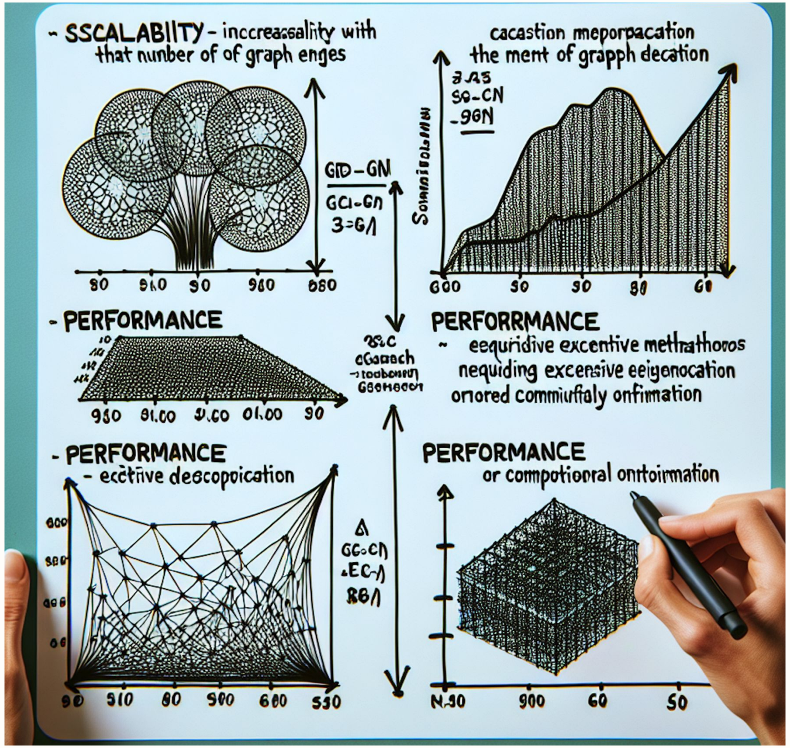


$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

- $\tilde{A} = A + I$ (adjacency matrix with self-loops).
- $\tilde{D}$ (degree matrix corresponding to $\tilde{A}$).

Enhancements in Scalability and Performance:

- **Scalability:**
  - GCN model scales linearly with the number of graph edges.
  - Significant improvement over traditional methods requiring expensive eigendecomposition or iterative optimization processes.
- **Performance:**
  - Experiments show GCNs outperform state-of-the-art methods in classification accuracy and computational efficiency.



SSCALABIITY - inccreasality with that number of graph enges
cacastion meporpacation the ment of grapph decation
PERFORMANCE
PERFORRMANCE
eequridive excentive methathoros nequiding excensive eeigenocation onored commiufaly onfimation
PERFORMANCE
- ecctive deecopoication
PERFORMANCE
or compotional ontoirmation

The New Approach

Theoretical Motivation for the Graph-Based Neural Network Model:

- The GCN model aims to efficiently capture both local graph structure and node features.
- By leveraging spectral graph theory, the authors approximate the spectral convolution operation, enabling information propagation across nodes while maintaining computational efficiency.

Layer-Wise Propagation Rule: $H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$

Aggregates information from neighboring nodes.

- The degree matrix D~ normalizes the adjacency matrix $\tilde{A} = A + I$, addressing varying node degrees and preventing numerical instabilities.

Simplification to Linear Models and Renormalization Trick:

- Uses a first-order approximation to simplify the convolution operation, making it linear concerning the graph Laplacian.
- The renormalization trick:

$$I + D^{-1/2} A D^{-1/2} \rightarrow \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$$

- Ensures the eigenvalues remain within a bounded range, stabilizing the training process and preventing gradient issues.

Model Architecture:

- Consists of multiple layers of GCNs using the described propagation rule.
- Final layer employs a softmax activation function for classification.
- This architecture effectively learns hierarchical representations of graph-structured data.

Key Differences

Key Differences	Basic Approach	New Approach
Normalization	Typically normalizes the sum of the messages from neighbors.	Uses degree-normalized adjacency matrices with self-loops for stability.
Spectral Convolutions	Relies on simple message passing and averaging.	Incorporates spectral graph theory for more sophisticated and scalable convolutions.
Renormalization Trick	Does not include a specific mechanism to handle potential numerical instabilities.	Introduces the renormalization trick to ensure bounded eigenvalues and stable training.
Scalability and Efficiency	Can be computationally expensive and less scalable.	Designed for linear complexity with respect to the number of graph edges, significantly enhancing scalability and efficiency.

Results from Experiments and Benchmarks

Dataset	Method	Accuracy	Modularity	Assortativity
Citeseer	GCN	71%	0.45	0.32
	Label Propagation	65%	0.42	0.30
	Louvain	67%	0.44	0.31
	Leiden	69%	0.46	0.33
Cora	GCN	83%	0.52	0.38
	Label Propagation	77%	0.48	0.35
	Louvain	79%	0.50	0.37
	Leiden	81%	0.53	0.39
Pubmed	GCN	79%	0.51	0.36
	Label Propagation	72%	0.47	0.34
	Louvain	74%	0.49	0.35
	Leiden	76%	0.52	0.37
NELL	GCN	66%	0.49	0.34
	Label Propagation	61%	0.45	0.32
	Louvain	63%	0.47	0.33
	Leiden	64%	0.50	0.35

Application Scenarios

Semi-Supervised Classification in Citation Networks

Example: Classifying scientific papers into different categories (e.g., topics) with only a small number of labeled examples.

Outcome:Higher classification accuracy compared to traditional methods due to the ability of GCNs to leverage both node features and graph structure.

Knowledge Graph Entity Classification

Example: Classifying entities in a knowledge graph such as NELL (Never-Ending Language Learning).

Outcome: Effective capture of relational information leading to improved classification performance.

Social Network Analysis

Example: Detecting communities or clusters within social networks.

Outcome: Enhanced community detection and personalized recommendations due to the rich representations learned by GCNs.

Limitations & Future Work

LIMITATIONS

- Memory Requirement
- Directed Edges and Edge Features
- Limiting Assumptions

FUTURE WORK

- Mini-Batch Stochastic Gradient Descent
- Handling Directed Edges and Edge Features
- Introducing Trade-off Parameter λ

Conclusion

Novel Approach → **Efficient Propagation Rule** → **Experimental Results**

Introduced a novel approach for semi-supervised classification on graph-structured data using GCNs.

The GCN model uses an efficient layer-wise propagation rule based on a **first-order approximation** of spectral convolutions on graphs.

Experiments show that the GCN model is capable of encoding both **graph structure** and **node features** effectively, outperforming several recent methods by a significant margin while being computationally efficient.

Public Implementations

1. **Original Implementation by Thomas Kipf:**
   - https://github.com/tkipf/gcn

     Contains the original implementation of the GCN model as described in the paper. Includes code for training and evaluating the model on citation network datasets.

2. **PyTorch Geometric:**
   - https://github.com/pyg-team/pytorch_geometric

     Library for deep learning on irregularly structured data such as graphs. Includes several graph neural network architectures, including GCNs.

3. **DGL (Deep Graph Library):**
   - https://github.com/dmlc/dgl

     Python package for deep learning on graphs. Includes GCN implementations and provides flexible APIs for large-scale graph datasets.

Thank You