Multi-Task Machine Learning
Understanding Social Networks
Leveraging Local Node Features for Structural Profiling

Motivation

With the ever-increasing prevalence of complex network data spanning various domains such as social networks, biological networks, transportation networks, and more, there arises an urgent need to comprehensively analyse and understand the fundamental structure of these intricate networks. This analysis plays a pivotal role in acquiring invaluable insights and facilitating astute decision-making processes. Moreover, it holds immense potential in enabling personalization, targeted interventions, bolstering security measures, optimising system performance, and driving advancements in research and knowledge.

Problem Statement

This project aims to harness the power of machine learning techniques to delve into the intricate world of graph analysis, specifically focusing on

1. Link Prediction: Predict connections within the graph.
2. Influence Prediction: Forecast the influence of nodes within the network.
3. Community Detection: Determine if two nodes belong to the same network community.
4. Clustering: Cluster nodes based on their local properties.

While you may be new to these topics, this project is designed to encourage your curiosity and innovative spirit. Embrace this opportunity to dive deep into graph theory, utilise local node features, and innovate in these exciting areas of network analysis. Dive into in-depth literature reviews, navigate new topics, and refine your ideas through regular guidance.

This endeavour holds tremendous potential across various domains, be it for targeted marketing, community detection initiatives, or the deeper comprehension of complex social dynamics.

Introduction

A plethora of features can be derived from any given graph. Some of the features are given here. Local and global features are both valuable for extracting insights from graph data. The extraction of features from graphs can encompass various characteristics such as node degree, centrality measures, neighbourhood information, and more. However, local features are particularly advantageous due to their simplicity in extraction. By focusing on individual nodes and their immediate surroundings, local features involve less computational complexity compared to analysing the entire graph structure. This ease of extraction makes local features a preferred choice in many graph analysis tasks.

In this project, we will utilize local features only. Leveraging local features for graph machine learning tasks offers other notable advantages. Local features also provide interpretability, allowing for an easier understanding of node behaviour and characteristics. Additionally, local

features exhibit robustness, as they remain relatively stable even when the graph structure changes. They also enable scalability by enabling parallelized feature extraction for efficient processing of large-scale graphs. Lastly, the generalizability of local features ensures their applicability across different graph datasets and facilitates the transferability of trained models.

Our focus solely lies on understanding the interplay between nodes within their local context. We will use engineered features derived from the neighbourhood of the nodes within a range of 2-3 hops. The objective is to uncover relationships between nodes, identifying patterns, similarities, and potential connections.

Procedure

- ***Data Collection***: In this project, we will utilise one of the social networks from the Stanford SNAP repository, which provides a collection of real-world network datasets. We will use the Facebook dataset for this project. Use the file ***'facebook_combined.txt'*** for generating the graph.

- ***Graph Representation***: Convert the collected graph data into a suitable format for analysis using relevant Python libraries. Choose an appropriate representation, such as an adjacency matrix or an edge list, based on the characteristics of the graph and the specific analysis requirements. Python libraries like NetworkX and igraph can be employed for efficient graph manipulation and representation.

- ***Feature Engineering***: Derive informative features by extracting pertinent information ***from the local properties of the graph nodes.*** This process involves considering various features such as node degree, centrality measures, neighbourhood characteristics, and other relevant local structural properties that can effectively differentiate nodes. Creating an exhaustive and well-informed feature list for this task is crucial, leveraging online articles and research papers as valuable resources for gathering insights on deriving local node features from network data.

- ***Data Preparation***: Prepare the feature-engineered data. This may involve normalisation or scaling of the features, handling missing values, or any other necessary preprocessing steps to ensure the data is suitable for applying ML algorithms.

- ***Evaluation***: Assess the quality of individual tasks by employing appropriate evaluation metrics.

- ***Interpretation and Analysis:*** Analyse the obtained results to gain insights into the structural properties shared by the nodes. Interpret the results in the context of the problem domain.

- ***Documentation and Reporting***: Document the methodology, results, and findings of the analysis process. Prepare a comprehensive report summarising the problem, approach,

experiments conducted, and conclusions drawn. Communicate the insights and potential applications of the analysis.

Cautions and Recommendations

- ***Utilise local features only***.
- Begin by executing your code on a small dataset as an initial step. A sample dataset is provided for reference.
- It is important to note that visualising a network becomes challenging and computationally intensive when the number of nodes exceeds 500. Avoid investing excessive time in generating visualisations under such circumstances.
- Consider using the PyCaret package for running ML algorithms.

Additional Pointers for Similar Projects

➔ *Domain Knowledge*: Cultivate a profound understanding of the domain(s) from which the graph data originates. This knowledge will empower you to make informed decisions about feature selection, result interpretation, and validation.

➔ *Data Preprocessing*: Master essential data preprocessing techniques, including data cleaning, outlier handling, and normalisation. This will enhance the quality and reliability of your analysis results.

➔ *Visualisation Techniques*: Explore a variety of visualisation methods tailored for graph data, such as network visualisation and feature distribution plots. Leverage visualisations to unveil patterns, comprehend the graph's structure, and effectively communicate your findings.

➔ *Handling Large-Scale Graphs*: Acquire strategies for addressing the challenges associated with large-scale graphs. Familiarise yourself with techniques like graph sampling, partitioning, and parallel processing to analyse and scale computations efficiently.

➔ *Ethical Considerations*: Develop a keen awareness of the ethical aspects involved in working with graph data, particularly regarding privacy and sensitive information. Apply appropriate data anonymization techniques and adhere to ethical guidelines.

Note:

☐ In the realm of graph-related tasks, there has been a notable shift towards Graph Neural Networks (GNNs). GNN-based methodologies commonly necessitate comprehensive knowledge of the entire network. However, our goal is to harness local network structure information. You have the option to investigate pre-trained alternatives for Graph Neural Networks that are capable of fulfilling this objective.