Finding Money Launderers Using Heterogeneous Graph Neural Networks

What is a Heterogeneous GNN?

A graph neural network (GNN) is a type of machine learning model that works on data represented as graphs (networks of nodes and edges).

A heterogeneous graph has different types of nodes (e.g., people, companies) and edges (e.g., transactions, relationships), unlike a homogeneous graph, which has only one type of node and edge. Therefore, unlike homogeneous graphs, which consist of a single type of node and a single type of edge, heterogeneous graphs (also known as heterogeneous information networks) contain multiple types of nodes and edges. This complexity reflects real-world networks more accurately, such as social networks, biological networks, and financial transaction networks.

Heterogeneous GNNs are designed to handle these complex graphs, making them ideal for real-world applications like detecting money laundering in financial networks.

Key Components of Heterogeneous GNNs

Nodes and Edges:

Nodes: Different types of entities (e.g., people, companies, transactions) are represented as nodes in the graph.

There are three types of nodes in the graph.

1) The first one is called **individual** and represents a human individual's customer relationship in the bank. It includes all of the individual's accounts in the bank.
2) The second type of node is called **organization** and represents an organizations or company's customer relationship in the bank in the same manner as a node representing an individual.
3) The third type of node is called **external** and represents a sender or recipient of a transaction that is outside of the bank.

Edges: Relationships or interactions between these entities (e.g., financial transactions, social connections) are represented as edges.

Heterogeneous Features:

Nodes and edges can have different features or attributes depending on their type. For example, a "person" node might have attributes like age and income, while a "transaction" edge might have attributes like amount and date.

Message Passing:

In GNNs, information is propagated through the network using a process called message passing. Nodes aggregate information from their neighbors to update their own representations.

In heterogeneous GNNs, this process needs to account for the different types of nodes and edges, requiring more complex aggregation and update mechanisms.

Proposed Model: HMPNN (Heterogeneous Message Passing Neural Network)

The paper introduces the Heterogeneous Message Passing Neural Network (HMPNN), an extension of MPNN for heterogeneous graphs. HMPNN uses edge features and combines embeddings from different node-edge types. The HMPNN model significantly improves the detection of money laundering activities, highlighting the

importance of using appropriate GNN architectures for heterogeneous graphs. This work represents the first application of GNNs on a large real-world heterogeneous network for AML purposes.

Heterogeneous GNN Models

Several models have been developed to handle the complexities of heterogeneous graphs. Some of these include:

1. **RGCN (Relational Graph Convolutional Networks)**:
    - Extends the traditional GCN (Graph Convolutional Network) to handle multiple types of edges by learning a separate weight matrix for each edge type.
2. **HAN (Heterogeneous Graph Attention Networks)**:
    - Uses attention mechanisms to learn the importance of different neighbors and meta-paths (sequences of edges connecting nodes) in heterogeneous graphs.
3. **MAGNN (MetaPath Aggregated Graph Neural Network)**:
    - Aggregates information along meta-paths to capture complex relationships in heterogeneous graphs.
4. **HGT (Heterogeneous Graph Transformer)**:
    - Applies the transformer model to heterogeneous graphs, using attention mechanisms to handle different types of nodes and edges.
5. **HetGNN (Heterogeneous Graph Neural Network)**:
    - Combines embeddings from different node and edge types using various aggregation methods.

Proposed HMPNN Model

The paper introduces a new model called the **Heterogeneous Message Passing Neural Network (HMPNN)**. Key features include:

1. **Message Aggregation**:
    - The HMPNN incorporates edge features and uses a novel method for aggregating messages, which helps in capturing the complex relationships in heterogeneous graphs.
2. **Real-World Application**:
    - The model is applied to a large dataset of bank transactions to detect money laundering activities. The use of heterogeneous GNNs in this context allows for more accurate identification of suspicious patterns compared to traditional rule-based systems.

What is HMPNN?

Heterogeneous Message Passing Neural Network (HMPNN) is an advanced type of graph neural network designed to handle heterogeneous graphs. Heterogeneous graphs have different types of nodes and edges, making them more complex than homogeneous graphs. HMPNN is specifically tailored to leverage this complexity to improve the detection of patterns such as money laundering in financial transaction networks.

How HMPNN Works

1. Graph Representation

- **Nodes**: Represent different entities, such as individuals, companies, and transactions.

- **Edges**: Represent relationships or interactions between these entities, such as financial transactions or business connections.
- **Node and Edge Features**: Each node and edge can have features (attributes) that provide additional information. For example, a transaction edge might have features like the transaction amount and date.

2. Message Passing Mechanism

- **Message Passing**: In GNNs, information is propagated through the graph using a process called message passing. Nodes send and receive messages (information) from their neighbours.
- **Heterogeneous Message Passing**: In HMPNN, this process is extended to handle different types of nodes and edges. This means the model can aggregate and update node representations based on the type of node and edge involved in the interactions.

3. Aggregation and Update Functions

- **Aggregation**: For each node, the HMPNN aggregates messages from its neighbours. This involves combining information from various types of edges and nodes. The aggregation function can vary based on the type of node and edge, allowing the model to capture complex interactions.
- **Update**: After aggregation, the node's representation is updated. This new representation incorporates the aggregated information and reflects the node's new state after considering its neighbor's information.

4. Edge Features Integration

- HMPNN incorporates edge features directly into the message-passing process. This is crucial because the characteristics of the transactions (edges) play a significant role in detecting suspicious activities. For example, unusually large transaction amounts or transactions occurring at odd times might indicate money laundering.

5. Output Layer

- After several layers of message passing, aggregation, and updates, the final node representations are used for the task at hand. In this case, the task is to detect suspicious nodes (entities involved in money laundering).
- The model outputs a score or classification for each node, indicating the likelihood of being involved in money laundering.

Advantages of HMPNN

1. **Enhanced Detection Capability**:
    - By leveraging the rich structure of heterogeneous graphs, HMPNN can capture more intricate patterns and relationships that simpler models might miss.
2. **Use of Edge Features**:
    - Incorporating edge features into the message passing mechanism allows the model to utilize critical information about transactions, improving its accuracy in identifying suspicious activities.
3. **Flexibility**:
    - The model can be adapted to various types of heterogeneous networks beyond financial transactions, making it a versatile tool for different applications involving complex relational data.

Practical Implementation

1. **Data Preparation**:
    - Collect and preprocess data to form a heterogeneous graph. This includes defining the types of nodes and edges, as well as their features.
2. **Model Training**:
    - Train the HMPNN on historical transaction data, where some nodes are labelled as suspicious or non-suspicious. The model learns to distinguish patterns associated with money laundering.
3. **Prediction**:
    - Apply the trained HMPNN to new, unseen data to predict suspicious activities. The model provides scores or classifications for nodes, helping to identify potential money laundering cases.

By explaining these details, you can give a clear and comprehensive understanding of what HMPNN is and how it works, emphasizing its advantages and practical implementation for detecting money laundering.

How can we implement it for the unsupervised data:

Yes, the Heterogeneous Message Passing Neural Network (HMPNN) can be adapted for unsupervised learning, which is particularly useful when labeled data is scarce or unavailable. In an unsupervised approach, the model can learn patterns and structures within the data without explicit labels indicating which transactions are laundering and which are not. Here's how it can work:

Unsupervised HMPNN for Money Laundering Detection

1. Graph Representation

- **Nodes and Edges**: As before, nodes represent entities (individuals, companies, transactions) and edges represent relationships (financial transactions). Node and edge features provide additional information.
- **Graph Construction**: Create a heterogeneous graph using the available transaction data, ensuring that all relevant attributes are included.

2. Unsupervised Learning Objectives

- **Node Embedding Learning**: The goal is to learn **embeddings (representations)** for each node that capture the underlying patterns and structures in the graph.
- **Graph Autoencoders**: One common approach is to use graph autoencoders, where the model tries to reconstruct the graph from the learned embeddings. This helps the model learn meaningful representations.

3. Graph Autoencoder Framework

- **Encoder**: The HMPNN acts as the encoder, which processes the heterogeneous graph and generates **low-dimensional embeddings** for each node.
- **Decoder**: The decoder reconstructs the graph from these embeddings. The **reconstruction loss** (difference between the original graph and the reconstructed graph) guides the training process.

4. Anomaly Detection

- **Anomaly Score Calculation**: After training, nodes, and edges with high reconstruction errors (i.e., those that are poorly reconstructed) are considered anomalies. These anomalies may indicate unusual patterns potentially associated with money laundering.
- **Clustering**: Another approach is to cluster the learned embeddings. Nodes that do not fit well into any cluster or belong to small, isolated clusters can be flagged as suspicious.

5. Implementation Steps

- **Data Preparation**: Construct the heterogeneous graph from the transaction data, including all relevant features.
- **Model Training**: Train the HMPNN-based autoencoder on the graph, minimizing the reconstruction loss.
- **Anomaly Detection**: After training, analyze the reconstruction errors or cluster the embeddings to identify anomalies.

Example Workflow

1. **Construct the Graph**:
    - Nodes: Individuals, companies, transactions.
    - Edges: Transactions between entities, with attributes like amount, date, and frequency.
2. **Initialize the HMPNN Autoencoder**:
    - **Encoder**: Use HMPNN to generate embeddings for nodes.
    - **Decoder**: Reconstruct the graph from embeddings and calculate reconstruction loss.
3. **Train the Model**:
    - Use an unsupervised learning algorithm to minimize reconstruction loss.
4. **Identify Anomalies**:
    - Calculate reconstruction errors for nodes and edges.
    - Flag those with high errors as potential money laundering activities.
    - Alternatively, cluster node embeddings and flag outliers.

Benefits of Unsupervised HMPNN

1. **No Need for Labelled Data**: The model can learn from the structure and features of the data without requiring explicit labels.
2. **Discovery of Hidden Patterns**: Unsupervised learning can uncover patterns and anomalies that are not apparent through traditional rule-based methods or supervised learning.
3. **Adaptability**: The model can continuously learn and adapt as new data is added, making it suitable for dynamic environments.

Challenges

1. **Evaluation**: Without labeled data, evaluating the model's performance can be challenging. One approach is to validate the model on a subset of the data with known labels or through expert review.
2. **Interpretability**: Understanding why certain nodes are flagged as anomalies may require additional analysis and domain expertise.

By using an unsupervised HMPNN approach, financial institutions can enhance their ability to detect suspicious activities even when explicit labels are not available, leveraging the model's ability to learn from the inherent structure and features of transaction data.

Contributors:

Harshit Bhushan