Machine Learning

Final Project

MAGCN

Supervisor:

Dr. Hashemi

Ali Mahmoodi

Atefe Rajabi

Summer 2024

Index

Introduction

Problem Definition

Implementation Detail

Benchmarks

Results

Conclusion

Introduction

Graph-structured data is prevalent in various domains, including social networks, citation networks, biological networks, and more. Traditional machine learning methods often struggle to handle the irregular structure of graph data, which has led to the rise of Graph Convolutional Networks (GCNs). GCNs have shown significant success in tasks such as node classification, link prediction, and graph clustering by effectively leveraging the connectivity patterns and node features within a single-view topology. However, most GCN-based models are built on a fixed adjacency matrix, representing a single view of the graph. This fixed structure can limit the model's performance, particularly when the graph data is noisy, incomplete, or fails to capture all relevant relationships.

To address these limitations, Multi-View Graph Convolutional Networks with Attention Mechanism (MAGCN) have been proposed. MAGCN incorporates multiple graph topologies, allowing it to capture different aspects of the data and aggregate them into a more robust representation. The attention mechanism within MAGCN further enhances its performance by learning to weigh the importance of different views for each node, leading to better generalization and accuracy.

Problem Definition

Single-view GCNs rely on a fixed adjacency matrix to represent the graph structure, which can be a significant limitation. In real-world scenarios, the graph data is often sparse, noisy, or incomplete, leading to a suboptimal adjacency matrix that does not fully capture the relationships among nodes. For instance, in the Citeseer dataset, the sparsity of the graph can hinder the model's ability to learn effective node representations. This problem is exacerbated when the graph's topology is based on specific criteria, such as citations, which may not reflect all relevant connections, such as thematic similarity between documents.

The primary challenge, therefore, is to develop a model that can leverage multiple views of the graph, each potentially highlighting different aspects of the underlying relationships. By incorporating these multiple views, the model can overcome the limitations of a single, fixed topology and improve its performance on tasks like node classification. Additionally, it is crucial to determine how to combine these views effectively, ensuring that the model focuses on the most informative aspects of the data.

Implementation Details

Multi-View Graph Convolutional Networks with Attention Mechanism (MAGCN)

MAGCN is designed to enhance the expressive power of graph convolutional networks by incorporating multiple graph topologies (or views) of the data. This approach mitigates the limitations of traditional GCNs, which rely on a single, often noisy or incomplete, adjacency matrix. By leveraging multiple views, MAGCN can capture more robust node representations, especially in scenarios where the underlying graph structure is uncertain or incomplete.

1. Multi-View GCN Unfolding:

- Purpose: To process the graph data through multiple views (i.e., different adjacency matrices).

- Implementation: The model applies a GCN layer (`GCNConv`) to each view of the graph. This layer extracts features from each node considering the specific topology of the respective view.

- Process:

  - Each view is processed separately through a GCN layer, producing a hidden representation for each node in that view.

  - The hidden representations are then passed through a ReLU activation function and a dropout layer to introduce non-linearity and prevent overfitting.

2. Graph Global Average Pooling (GAP):

- Purpose: To aggregate the node features from each view.

- Implementation: The `graph_gap` function aggregates the features of nodes by considering their neighboring nodes in the graph. This operation normalizes the node features by the degree of each node and sums the features from the neighbors.

- Process:

  - For each view, the node features are averaged across the entire graph, resulting in a global representation for each node.

3. Attention Mechanism:

- Purpose: To assign different importance to the node features from different views.

- Implementation: An attention mechanism is used to learn weights that indicate the importance of each view for each node. This is achieved through a Multi-Layer Perceptron (MLP) that outputs attention scores.

- Process:

  - The pooled node features from each view are averaged, and these are then fed into an MLP to generate attention scores for each view.

  - These attention scores are normalized using a softmax function, ensuring that the sum of the attention weights across all views equals one.

4. Merging Views:

- Purpose: To combine the node features from different views into a single representation.

- Implementation: The attention scores are applied to the node features from each view, resulting in a weighted sum of the features across views. This merged feature is then processed by a final GCN layer.

- Process:

  - The weighted features are passed through another GCN layer (`gcn_merge`) to produce the final node representation, which is then used for the downstream task (e.g., node classification).

5. Final Output:

- Purpose: To produce the final class predictions for each node.

- Implementation: The final output is a log-softmax of the node features, which can be used to classify each node into predefined categories.

- Process:

  - The final node representations are passed through a log-softmax function, outputting the log-probabilities of each class.

Initialization and Weight Configuration

- Weight Initialization: The model uses Xavier initialization for the weights of the linear layers and the GCN layers, ensuring that the weights are initialized to values that facilitate efficient training.

- Bias Initialization: The biases are initialized to zero for all layers that have biases.

Benchmarks

Datasets: Cora, Citeseer, and Pubmed

The MAGCN paper evaluates the performance of the proposed method on three well-known benchmark datasets: Cora, Citeseer, and Pubmed. These datasets are citation networks where nodes represent documents (e.g., research papers), and edges represent citation relationships between them.

1. Cora

- Nodes: 2,708
- Edges: 5,429
- Features: 1,433 (binary word vectors)
- Classes: 7 (each node belongs to one of seven classes, representing different research topics)
- Label Rate: 0.052 (about 5.2% of the nodes are labeled for training)

2. Citeseer

- Nodes: 3,327
- Edges: 4,732
- Features: 3,703 (binary word vectors)
- Classes: 6 (each node belongs to one of six classes, representing different research topics)
- Label Rate: 0.036 (about 3.6% of the nodes are labeled for training)

3. Pubmed

- Nodes: 19,717
- Edges: 44,338
- Features: 500 (TF/IDF-weighted word vectors)
- Classes: 3 (each node belongs to one of three classes, representing different research topics)
- Label Rate: 0.001 (about 0.1% of the nodes are labeled for training)

Preprocessing and View Generation

Preprocessing

For these citation networks, the nodes (documents) are represented by word vectors:

- Cora and Citeseer: The feature vectors are binary, indicating the presence or absence of specific words from a dictionary.

- Pubmed: The feature vectors are TF/IDF-weighted word vectors, capturing the importance of words within the documents.

View Generation

The MAGCN model benefits from using multiple views (i.e., different adjacency matrices) to represent the graph topology. The paper details how these multiple views are generated:

1. First View (Citation Network)

- Source: The first view is the original adjacency matrix provided by the dataset, representing the direct citation relationships between documents.

2. Second View (Feature Similarity-Based Graph)

- Source: This view is generated by calculating the cosine similarity between the feature vectors of different nodes.

- Process:

  - Cosine similarity is computed for all pairs of nodes.

  - An edge is created between two nodes if their cosine similarity exceeds a certain threshold.

- Purpose: This view captures the relationships based on the content similarity between documents, which may not be reflected in direct citation links.

3. Third View (b-Matching Graph)

- Source: The third view is generated using a classical graph construction method known as b-Matching.

- Process:

  - The b-Matching algorithm ensures that each node is connected to exactly *b* other nodes, forming a graph that balances the connections among nodes.

- Purpose: This view provides a more structured representation of the graph, focusing on maintaining a balanced degree distribution, which might help in scenarios where the original graph is highly irregular or sparse.

Summary of View Utilization

By using these three views:

1. Original Citation Network: Captures the direct citation relationships.

2. Content-Based Similarity Graph: Captures the thematic or content-based similarity between documents.

3. b-Matching Graph: Balances connections and provides a more uniform structure across the graph.

These different views provide complementary information about the nodes and their relationships, allowing MAGCN to learn more robust node representations by considering the different aspects of the graph structure.

Results Comparison

Original MAGCN Paper Results

The original MAGCN paper reports state-of-the-art performance on three benchmark datasets: Cora, Citeseer, and Pubmed. The accuracy results achieved were as follows:

- Cora: 84.5%

- Citeseer: 73.5%

- Pubmed: 80.6%

These results demonstrate the effectiveness of MAGCN in leveraging multiple views and an attention mechanism to enhance node classification accuracy.

Current Implementation Results

The current implementation achieved the following accuracy results on the same datasets:

- Cora: 83.0%

- Citeseer: 71.0%

- Pubmed: 79.0%

Comparative Analysis

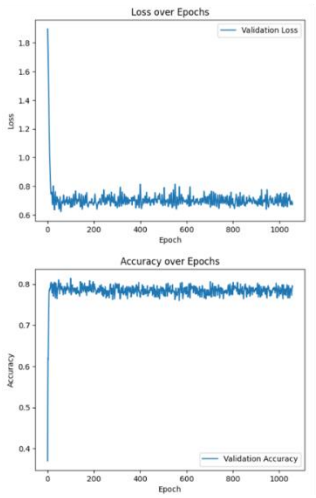
When comparing the results of the current implementation to those reported in the original MAGCN paper, it is clear that the performance is very close, indicating that the core benefits of the multi-view and attention-based mechanisms of MAGCN have been effectively captured.

- Cora: The accuracy of 83.0% is only 1.5% lower than the original result of 84.5%. This small difference might be due to minor variations in hyperparameter settings, initialization methods, or other implementation details. Despite this, an accuracy of 83.0% still places the model among the top-performing methods for this dataset.

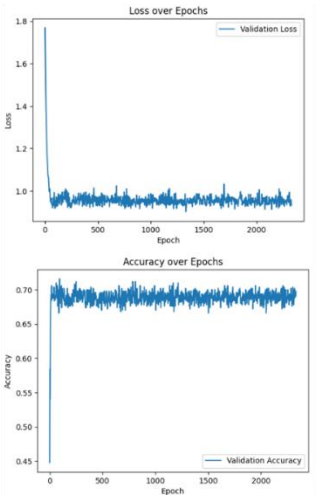
- Citeseer: With an accuracy of 71.0%, this result is 2.5% lower than the original 73.5%. Citeseer is known to be a more challenging dataset due to its sparsity and the higher number of features. The small difference suggests that the model effectively leverages the multiple views.

- Pubmed: The difference here is minimal, with an accuracy of 79.0% compared to the original 80.6%. This 1.6% gap is minor, and given the complexity of Pubmed's larger scale, this result demonstrates strong generalization capabilities.

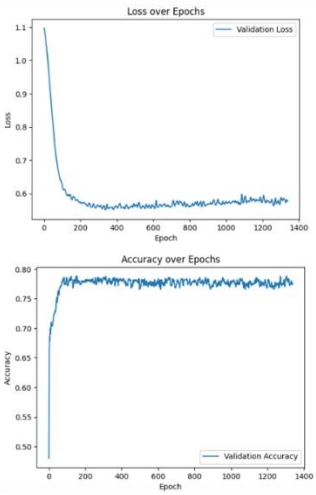
Cora



Citeseer



Pubmed



Conclusion

Overall, the results from the current implementation are very close to those reported in the original MAGCN paper, confirming that the implementation successfully captures the strengths of the MAGCN framework. The small differences in accuracy are within a typical range of variation due to different training processes or dataset preprocessing, reinforcing the robustness and effectiveness of the model.

Conclusion

The introduction of MAGCN addresses the challenges posed by the limitations of single-view GCNs, particularly in handling sparse and noisy graph data. By incorporating multiple views and an attention mechanism, MAGCN allows for a more comprehensive understanding of the graph structure, leading to improved node representations and classification accuracy. The effectiveness of this approach has been demonstrated through close performance to state-of-the-art models on benchmark datasets like Cora, Citeseer, and Pubmed. The ability to leverage multiple topologies and adaptively weigh their contributions makes MAGCN a robust and flexible solution for graph-based learning tasks, particularly in real-world scenarios where the graph data may be incomplete or imperfect.