Structural Dependencies in Graph-Based Node Classification with Classical Machine Learning Models

Ananya Uday Naik

December 21, 2025

Abstract

Graph-structured data appears in many real-world systems such as citation networks, communication platforms, and email networks. In this work, we study how graph structure alone affects the performance of classical machine learning models for node classification. Given a graph $G = (V, E)$, we construct interpretable structural feature mappings $\phi : V \to \mathbb{R}^d$ using node-level properties and neighborhood-level aggregates, while intentionally excluding node attributes such as textual content.

Using the Cora citation network, we evaluate models with different assumptions, including Naive Bayes, Softmax Regression, and Random Forests. Our experiments show that node-level structural features provide limited predictive power, whereas neighborhood-based features contribute most of the classification performance. We further observe that linear models capture only part of this structural signal, while non-linear models benefit significantly from feature interactions introduced by the graph topology.

In addition to empirical results, this study highlights practical advantages of graph-structural learning. By analyzing interaction patterns at the network level rather than processing individual messages or documents, such approaches can reduce computational cost and scale more efficiently to large systems. Overall, our findings suggest that label information in graph-based data is largely determined by neighborhood structure and non-linear relationships.

1 Introduction

Graphs are a natural representation for many complex systems, including citation networks, social networks, and biological systems. In such settings, data points are not independent but are connected through rich relational structure. A fundamental task in graph learning is node classification, where the goal is to predict a label for each node based on available information.

An important question in this setting is how much information about node labels is already present in the structure of the graph itself. Classical machine learning models offer a simple and interpretable framework for studying this question, allowing us to isolate the role of graph topology without introducing complex architectures.

In this work, we investigate how different aspects of graph structure influence the performance of classical machine learning classifiers. Rather than relying on node attributes such as text content, we focus exclusively on structural features derived from the graph topology. By comparing models with different assumptions—including feature independence, linear decision boundaries, and non-linear interactions—we aim to understand where classical methods succeed and where they fail in graph-based learning problems.

2 Problem Formulation

We consider a graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Each node $v \in V$ is associated with a categorical label $y_v \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the set of possible classes. The edges represent interactions between nodes, such as citation relationships in a citation network or message exchanges in a communication network.

The objective is to perform node classification: given the graph structure, predict the label y_v for each node $v \in V$. In this study, we focus exclusively on information derived from the graph topology. Specifically, each node is mapped to a feature vector $\phi(v) \in \mathbb{R}^d$ constructed from structural properties of the graph, without using any node attributes such as textual content.

We evaluate classical machine learning models that learn a mapping

$$f_\theta : \phi(v) \to y_v,$$

where θ denotes the model parameters. By comparing models with different assumptions about feature independence, linearity, and non-linear interactions, we aim to understand how graph structure influences classification performance.

3 Dataset

We conduct our experiments on the Cora citation network, a widely used benchmark dataset for node classification tasks. In this dataset, each node represents a research paper, and a directed edge from node u to node v indicates that paper u cites paper v. Each node is associated with a topic label corresponding to the research area of the paper.

The dataset contains several thousand nodes distributed across multiple classes, with a sparse citation structure typical of real-world networks. In this study, we intentionally exclude textual features and use only the graph structure to construct node representations. This allows us to isolate the influence of graph topology on classification performance.

The Cora dataset is well suited for this analysis because citation relationships exhibit strong structural patterns, such as communities and hub nodes, which are informative for understanding how neighborhood structure relates to node labels.

4 Graph Structural Features

To study the influence of graph topology on node classification, we represent each node using a set of interpretable structural features derived from the graph. These features are computed solely from the network structure and do not rely on any node attributes. We group the features into node-level and neighborhood-level categories.

4.1 Node-Level Features

Node-level features capture local properties of individual nodes within the graph. For each node $v \in V$, we compute:

- **In-degree**: the number of incoming edges to v, representing how many nodes cite or connect to v.
- **Out-degree**: the number of outgoing edges from v, representing how many nodes v cites or connects to.
- **Clustering coefficient**: a measure of how densely connected the neighbors of v are, indicating local cohesion.

- **Triangle count**: the number of triangles involving v, capturing small-scale connectivity patterns.

These features describe the immediate structural role of a node but do not incorporate information about the broader neighborhood beyond one-hop connections.

4.2 Centrality Measures

We additionally compute global and semi-global centrality measures to capture the importance of nodes within the overall graph structure:

- **Betweenness centrality**: measures how often a node lies on shortest paths between other nodes, reflecting its role as a connector or bridge.
- **Closeness centrality**: measures how close a node is to all other nodes in the graph based on shortest path distances.
- **Eigenvector centrality**: assigns higher scores to nodes that are connected to other highly connected nodes, capturing recursive notions of importance.

While these measures capture global structural roles, they still characterize nodes individually rather than their surrounding context.

4.3 Neighborhood-Level Features

To incorporate local context, we construct neighborhood-level features by aggregating structural properties of neighboring nodes. For each node v, we compute statistics such as the mean and maximum values of selected node-level and centrality features over its immediate neighbors. These aggregated features capture patterns in the local neighborhood structure, such as whether a node is surrounded by highly connected or influential neighbors.

Neighborhood-level features enable the model to exploit relational information encoded in the graph, providing a richer representation than node-level features alone. As shown in our experiments, these features play a critical role in improving classification performance.

5 Models and Assumptions

To study how graph structure interacts with different learning assumptions, we evaluate three classical machine learning models: Naive Bayes, Softmax Regression, and Random Forests. These models were chosen because they make progressively weaker assumptions about feature independence and linearity, allowing us to analyze how structural dependencies affect performance.

5.1 Naive Bayes

Naive Bayes is a probabilistic classifier that assumes conditional independence between features given the class label. For a node feature vector $\phi(v) = (x_1, x_2, \ldots, x_d)$, the model assumes

$$P(y \mid \phi(v)) \propto P(y) \prod_{i=1}^{d} P(x_i \mid y).$$

This assumption is often violated in graph-structured data, where structural features such as degree, clustering, and neighborhood statistics are inherently correlated. As a result, Naive Bayes serves in this study not as a competitive classifier, but as a baseline to illustrate the impact of feature dependence induced by graph topology.

5.2 Softmax Regression

Softmax Regression is a multi-class linear classifier that models the conditional probability of each class using a softmax function. For a node feature vector $\phi(v) \in \mathbb{R}^d$ and a set of class-specific weight vectors $\{\mathbf{w}_k\}_{k=1}^{|\mathcal{Y}|}$, the predicted probability for class k is given by

$$P(y = k \mid \phi(v)) = \frac{\exp(\mathbf{w}_k^\top \phi(v))}{\sum_{j=1}^{|\mathcal{Y}|} \exp(\mathbf{w}_j^\top \phi(v))}.$$

The model predicts the class with the highest probability and learns the weight vectors by minimizing the cross-entropy loss over the training data. Unlike Naive Bayes, Softmax Regression does not assume feature independence. However, it is limited to linear decision boundaries in the feature space.

In the context of graph-structural features, this model allows us to examine how much of the structural signal can be captured using linear combinations of node-level and neighborhood-level features, without explicitly modeling higher-order feature interactions.

5.3 Random Forest

Random Forests are ensemble models composed of multiple decision trees trained on random subsets of the data and feature space. Each decision tree partitions the feature space through a sequence of axis-aligned splits, producing a prediction $f_t(\phi(v))$ for tree t.

The final prediction of a Random Forest with T trees is obtained by aggregating the predictions of individual trees:

$$\hat{y} = \arg\max_k \sum_{t=1}^{T} \mathbb{I}\big(f_t(\phi(v)) = k\big),$$

where $\mathbb{I}(\cdot)$ is the indicator function.

By averaging across many trees, Random Forests can model complex, non-linear interactions between features while remaining robust to noise and feature correlations. In graph-structural learning, this enables the model to capture interactions between node-level and neighborhood-level features that are not representable using linear decision boundaries.

6 Experimental Setup

Each node is represented using the structural feature vectors described in Section 4. The dataset is split into training and test sets using a stratified 80/20 split to preserve class proportions. All continuous features are standardized using statistics computed from the training set.

We evaluate Naive Bayes, Softmax Regression, and Random Forest models using identical data splits and feature representations to ensure fair comparison. Hyperparameters are kept fixed at reasonable default values, as the goal of this study is to analyze the effect of model assumptions rather than to maximize performance.

Model performance is evaluated using classification accuracy and macro-averaged F1 score. Macro F1 is reported to account for class imbalance and to highlight differences in per-class behavior. Confusion matrices are used for qualitative analysis of prediction patterns.

7 Results and Analysis

7.1 Overall Model Performance

Table 1 summarizes the performance of the evaluated models using structural features derived from the graph. Among the three classifiers, Random Forest achieves the highest performance, with an accuracy of approximately 0.62 and a macro-averaged F1 score of approximately 0.61. Softmax Regression attains moderate performance, while Naive Bayes performs substantially worse across both metrics.

The gap between accuracy and macro F1 is particularly notable for Softmax Regression and Naive Bayes, indicating uneven class-wise performance and difficulty in correctly predicting minority classes. In contrast, Random Forest exhibits more balanced predictions across classes, reflected in its higher macro F1 score.

Table 1: Classification performance using graph-structural features

Model	Accuracy	Macro F1
Naive Bayes	0.24	0.21
Softmax Regression	0.43	0.29
Random Forest	0.62	0.61

7.2 Effect of Neighborhood-Level Features

A key finding of this study is the significant performance improvement observed when neighborhood-level structural features are included. Models trained using only node-level features, such as degree and clustering coefficient, achieve relatively low accuracy and macro F1 scores. However, incorporating neighborhood-aggregated features—such as mean and maximum neighbor degree and centrality—leads to a substantial increase in performance across all models.

Notably, Random Forest and Softmax Regression both exhibit clear jumps in accuracy and macro F1 when neighborhood features are added, highlighting the importance of local structural context in determining node labels. This result supports the hypothesis that graph labels are strongly influenced by the structure of a node's immediate neighborhood rather than by isolated node properties.

7.3 Node-Level versus Neighborhood-Level Contributions

An additional observation is that removing node-level features while retaining neighborhood-level features results in only a minor drop in performance. In several cases, the performance remains comparable to that achieved using the full feature set. This suggests that neighborhood-level features capture most of the label-relevant information present in the graph, while node-level statistics contribute relatively little additional signal.

This finding reinforces the view that node classification in graph-structured data is inherently relational: labels are better explained by how nodes are embedded within their local neighborhoods than by their individual structural characteristics.

7.4 Model Behavior and Structural Assumptions

The comparative behavior of the three models further illustrates how learning assumptions interact with graph-structural features. Naive Bayes performs poorly due to its conditional independence assumption, which is strongly violated by correlated node-level and neighborhood-level features. Softmax Regression relaxes this assumption but remains constrained by linear decision boundaries, leading to confusion between structurally similar classes.

Random Forest, by contrast, effectively captures non-linear interactions between structural features and demonstrates significantly improved class-wise performance. This suggests that the predictive signal encoded in graph structure is inherently non-linear and benefits from models capable of exploiting feature interactions.

8 Discussion and Limitations

This study highlights the role of graph structure in node classification using classical machine learning models. A central observation is that neighborhood-level structural features contribute more strongly to classification performance than isolated node-level statistics. This indicates that node labels are largely shaped by local interaction patterns rather than individual structural properties.

The comparative behavior of the evaluated models reflects how their assumptions interact with graph-structural features. Naive Bayes performs poorly due to its conditional independence assumption, which is strongly violated by correlated structural and neighborhood features. Softmax Regression relaxes this assumption but remains limited by linear decision boundaries, leading to uneven class-wise performance and the collapse of structurally similar classes. Random Forests, by contrast, capture non-linear feature interactions and show improved performance, suggesting that meaningful information in graph structure is encoded through such interactions.

Despite these insights, this study has several limitations. First, classification performance using structure alone remains moderate compared to approaches that incorporate node attributes or more expressive graph-based models. This indicates that while graph structure provides useful signal, it is not sufficient for high-accuracy classification in isolation. Second, the analysis is limited to a single dataset, and the observed trends may vary across different types of networks. Finally, hyperparameter tuning and model optimization were intentionally minimal, as the focus of this work was interpretability rather than performance maximization.

Overall, these results suggest that graph-structural features are best viewed as complementary signals that provide interpretable and computationally efficient insights into relational data, rather than as standalone replacements for more complex models.

9 Conclusion

In this work, we studied how graph structure influences the performance of classical machine learning models for node classification. By restricting the analysis to structural features derived solely from graph topology, we isolated the role of node-level and neighborhood-level information in determining classification outcomes.

Our experiments show that neighborhood-level structural features capture most of the predictive signal present in the graph, while isolated node-level statistics contribute relatively little. We further observed that model assumptions play a critical role: linear models are limited in their ability to exploit structural dependencies, whereas non-linear models benefit from interactions among structural features.

Overall, this study demonstrates that meaningful label information in graph-based data is largely encoded in local neighborhood structure and non-linear relationships. These findings provide an interpretable perspective on graph-based learning and highlight the value of classical models as analytical tools for understanding how graph structure shapes predictive performance.

References

[1] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[2] A. Ng. CS229: Machine Learning. Stanford University, course lecture notes, 2023. `https://cs229.stanford.edu/`

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.