COL 761 HW1 Report Question 3 Analysis

February, 2025

Team members	Entry Number
Hasit Nanda	2024VST9015
Ghosal Subhojit	2022MT61976
Arnab Goyal	2022MT61963

1 Introduction

The goal in this question was to classify molecular graphs by identifying key subgraphs that serve as discriminative features. We developed a systematic pipeline for selecting frequent and unique subgraphs by incorporating a variety of statistical tests, and then transforming graphs into feature vectors, and utilizing them in classification.

2 Methodology

2.1 Step 1: Splitting Graphs Based on Labels

To ensure a structured approach to identifying discriminative subgraphs, we first split the dataset into two subsets based on labels (0 and 1). This step was crucial to mitigate potential label imbalance in the dataset, as one label type may be more prevalent than the other. By handling each category separately, we ensured that the frequent subgraph selection process remained unbiased.

2.2 Step 2: Frequent Subgraph Mining

We applied the Gaston algorithm to extract frequent subgraphs within each category. The threshold for Gaston was set at 40%, ensuring that the selected subgraphs are frequent enough to be representative while avoiding noise from overly rare patterns.

2.3 Step 3: Filtering Common Subgraphs Across Categories

When combining the frequent subgraphs in the two categories, we ensured that each common subgraph only appeared once to prevent redundancy while still retaining potentially useful structural information.

To do this task we generate canonical labels for each subgraph, and keep only the unique canonical labels.

For generating canonical labels we define a partial ordering of the vertexes of the graph based on vertex invariants which we choose the *label* and *degree* of the vertex. And we create all possible adjacent matrices corresponding to all the valid permutations of of the vertexes following the partial ordering.

2.4 Step 4: Selecting a Set of Frequent Subgraphs

After filtering, we retained the most frequent subgraphs from both categories. This resulted in a pool of around 300-500 unique subgraphs, providing a sufficiently large yet manageable feature set.

2.5 Step 5: Ranking Subgraphs

To rank subgraphs, we made a feature selection pipeline that combines multiple statistical methods, ensuring both relevance and non-redundancy.
First, we filter redundant features using the Jaccard similarity coefficient.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are sets of samples containing a given feature. If similarity exceeds a threshold, the feature is removed.
For ranking, we use four statistical methods:
1. **Mutual Information (MI)** Which is used to determine the dependency between a feature and the target using:

$$I(X, Y) = \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

2. **Chi-Square Test** determines the independence between features and the target using:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

where O and E are the observed and expected counts.
3. **ANOVA F-Test** assesses variance differences between class distributions:

$$F = \frac{(\bar{X}_0 - \bar{X}_1)^2}{\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}}$$

where $\bar{X}_i$ and s_i^2 are means and variances for class i.
4. **Gini Index** evaluates feature purity:

$$G = 1 - \sum p_c^2$$

where p_c is the class probability within each feature split.
Finally, we find a weighted combination of these scores. Sort them and select the top 100 subgraphs, ensuring robust and discriminative feature selection.

Mathematical justification

To justify the use of a weighted average for ranking subgraphs, we analyze how it reduces variance using the properties of variance in linear combinations.

Suppose we have n different statistical scores $S_1, S_2, \ldots, S_n$, each representing a different ranking method (e.g., MI, Chi-Square, ANOVA, and Gini Index). Let $w_1, w_2, \ldots, w_n$ be the corresponding weights assigned to each score, where:

$$\sum_{i=1}^{n} w_i = 1.$$

The combined ranking score S for a subgraph is given by the weighted sum:

$$S = \sum_{i=1}^{n} w_i S_i.$$

Using the variance property of linear combinations:

$$\text{Var}(S) = \sum_{i=1}^{n} w_i^2 \text{Var}(S_i) + 2 \sum_{i<j} w_i w_j \text{Cov}(S_i, S_j).$$

Case 1: Uncorrelated Scores If the scores $S_1, S_2, \ldots, S_n$ are approximately uncorrelated (i.e., $\text{Cov}(S_i, S_j) \approx 0$ for $i \neq j$), then:

$$\text{Var}(S) = \sum_{i=1}^{n} w_i^2 \text{Var}(S_i).$$

Since weights satisfy $\sum w_i = 1$, choosing smaller w_i for high-variance methods and larger w_i for stable methods reduces the overall variance of S.

Case 2: Correlated Scores If the scores have positive correlations, the covariance terms contribute positively. However, assigning lower weights to highly correlated methods helps mitigate redundancy and ensures diverse statistical perspectives contribute to ranking.

Conclusion By using a weighted combination, we: - Reduce the influence of noisy and high-variance ranking methods. - Benefit from multiple perspectives, leveraging the strengths of different ranking methods. - Ensure that no single method dominates the ranking, leading to more robust and discriminative feature selection.

Thus, the weighted approach provides variance reduction, leading to more stable and generalizable subgraph selection.

2.6 Step 6: Transforming Graphs into Feature Vectors

Each molecular graph was converted into a binary feature vector, where each dimension represented the presence or absence of one of the selected 100 discriminative subgraphs. Subgraph isomorphism was performed using the `retworkx` library.

2.7 Step 7: Classification

The resulting feature vectors were fed into the provided classification algorithm to evaluate performance.

3 Results

We evaluated our algorithm on two datasets:

Dataset	**Train Accuracy**	**Test Accuracy**
NCI	0.946	0.845
Mutagenicity	0.819	0.787

Table 1: Performance of the classification algorithm on different datasets.

These results demonstrate that our feature selection approach was effective, achieving a reasonably high test accuracy while avoiding overfitting.

4 Justification of Our Approach

The key justifications for our approach are:

- Mitigating Label Imbalance: By splitting the graphs into two groups before mining frequent subgraphs, we prevent bias due to an uneven distribution of labels.
- Using Gaston for Efficient Mining: Gaston is well-suited for mining frequent subgraphs efficiently in large datasets.
- Support Threshold Justification: Using a 40% support threshold ensures that the selected subgraphs are frequent enough to be representative while avoiding noise from overly rare patterns.
- Canonical Labeling for Unique Subgraphs: This ensures that subgraphs are not counted multiple times, reducing redundancy while preserving useful structural information.
- Robust Feature Ranking: Our ranking method combines multiple statistical approaches to ensure strong, non-redundant feature selection, leveraging mutual information, chi-square independence, variance-based ANOVA testing, and Gini impurity measures to create a comprehensive evaluation.

5 Conclusion

We successfully implemented a graph classification pipeline that identifies and utilizes discriminative subgraphs as features. Our method achieved strong performance on the provided datasets, demonstrating its viability for molecular classification tasks. Future improvements could explore optimizing subgraph selection criteria further to enhance generalization performance.