COL761: Data Mining - Assignment 1

Task 3: Graph Indexing Using Discriminative Subgraphs

Spandan	Viraat	Gaurav
2022CS51138	2022CS51651	2022CS51144

February 3, 2026

1 Problem Statement

Let $D = \{g_1, g_2, \ldots, g_n\}$ be a database of labeled graphs and let q be a query graph. The objective of the graph indexing problem is to return the set

$$R_q = \{g_i \mid g_i \in D \text{ and } q \subseteq g_i\},$$

where $q \subseteq g_i$ denotes that q is subgraph isomorphic to g_i.

Subgraph isomorphism is NP-complete, and performing it exhaustively between a query graph and every database graph is computationally infeasible for large graph collections. Therefore, instead of exact matching, we design an index-based filtering approach that produces a candidate set C_q such that

$$C_q \supseteq R_q,$$

while keeping $|C_q|$ as small as possible.

The quality of an indexing scheme is evaluated using the metric

$$s_q = \frac{|R_q|}{|C_q|},$$

which measures the fraction of true matches within the candidate set. A higher value of s_q indicates a more effective index that aggressively filters out non-matching graphs without introducing false negatives.

2 Motivation and Algorithmic Choice

2.1 Challenges in Graph Indexing

Frequent subgraph mining (FSM) is a natural approach for graph indexing, as frequent substructures can serve as features. However, classical FSM algorithms such as gSpan or FSG attempt to enumerate all frequent subgraphs, leading to exponential growth in both time and memory. For molecular graph datasets, which may contain tens of thousands of graphs with non-trivial structure, such exhaustive enumeration is impractical.

Moreover, frequency alone does not guarantee discriminative power. Subgraphs that are too frequent appear in almost all graphs and thus provide little pruning ability, while extremely rare subgraphs may never appear in queries. Effective indexing therefore requires selecting subgraphs that strike a balance between frequency and discrimination.

2.2 Why a GASTON-Inspired Approach

We adopt a GASTON-inspired approach that exploits the natural hierarchy of graph patterns:

$$\text{Paths} \rightarrow \text{Trees} \rightarrow \text{Cyclic Graphs}.$$

This hierarchy allows incremental pattern growth with early pruning, avoiding the combinatorial explosion associated with general subgraph enumeration.

Molecular graphs are particularly well-suited to this strategy. Most molecules consist of short chains of atoms and limited branching structures. As a result, a large fraction of meaningful chemical structure can be captured using paths and simple trees, without explicitly mining complex cyclic subgraphs.

Key advantages of this approach include:

- Reduced computational and memory overhead
- Strong alignment with molecular graph structure
- Natural parallelization of path extraction
- Interpretable and structurally meaningful features

2.3 Excluded Alternatives

Approach	Reason for Exclusion
gSpan	Canonical labeling overhead and complexity
Graph fingerprints	Insufficient structural discrimination
Neural methods	Explicitly prohibited by the assignment
Random-walk features	Poor interpretability and weak guarantees

3 Overall System Architecture

3.1 Indexing Pipeline

The system is divided into an offline indexing phase and an online query phase.

```
OFFLINE PHASE
-------------
identify.sh        : Mine k = 50 discriminative subgraph patterns
convert.sh (DB)    : Convert database graphs into binary feature vectors

ONLINE PHASE
------------
convert.sh (Query) : Convert query graphs into feature vectors
generate_candidates.sh : Filter database graphs to produce C_q
```

All computationally expensive operations are performed offline. Online query processing consists only of fast vector comparisons, enabling scalability.

3.2 Code Organization

```
q3_submission/
 identify.sh
 identify_subgraphs.py
 fsm.py
 convert.sh
 convert_to_features.py
 feature_extractor.py
 generate_candidates.sh
 generate_candidates.py
 graph_utils.py
 env.sh
```

4 Pattern Mining Strategy

4.1 Two-Phase Mining

Phase 1: Path Mining. For each graph, depth-first search is initiated from every node to extract all simple paths up to a fixed maximum length. Paths are canonicalized to ensure uniqueness. These patterns capture linear molecular structures such as atom chains and sequential functional groups.

Phase 2: Tree Mining. For nodes with degree at least two, star-shaped tree patterns are extracted. These patterns represent branching structures such as functional group attachment points and multi-bond centers.

4.2 Why Both Paths and Trees

Pattern Type	Coverage	Discrimination
Paths only	High (>90%)	Medium
Trees only	Medium (∼60%)	High
Paths + Trees	Very High	High

Paths ensure broad coverage across graphs, while trees provide stronger discrimination. Combining both yields a balanced and effective feature set.

5 Discriminative Pattern Selection

Mining yields thousands of candidate patterns, but only $k = 50$ can be retained as index features. These patterns must maximize discrimination, minimize redundancy, and provide broad coverage.

5.1 Information Gain Scoring

Each pattern f is scored using a modified Information Gain metric:

$$IG(f) = H(p) \times w(p) \times p,$$

where p is the fraction of graphs containing f,

$$H(p) = -p \log_2 p - (1-p) \log_2 (1-p), \quad w(p) = 2 \min(p, 1-p).$$

5.1.1 Interpretation of the Score

Frequency	p	$H(p)$	$w(p)$	IG	Interpretation
Low	0.05	0.29	0.10	Low	Rare, unreliable
Moderate	0.30	0.88	0.60	High	Strong discriminator
High	0.90	0.47	0.20	Low	Too common to filter

This formulation naturally favors patterns appearing in the 20–50% frequency range.

5.2 Diversity via Overlap Filtering

High-IG patterns often overlap significantly. To enforce diversity, we use a greedy overlap-based selection strategy. Let $G(f)$ denote the set of graphs containing pattern f, and let S be the set of already selected patterns. A candidate pattern is accepted only if

$$\frac{|G(f) \cap G(S)|}{|G(f) \cup G(S)|} < \tau.$$

The overlap threshold τ is chosen adaptively:

- Mutagenicity: $\tau = 0.8$ (stricter diversity for smaller dataset)
- NCI-H23: $\tau = 0.7$ (mild redundancy for better coverage)

This adaptive strategy balances diversity and robustness across datasets of different scales.

6 Feature Extraction

Each graph is mapped to a 50-dimensional binary feature vector:

$$v_i = \begin{cases} 1 & \text{if pattern}_i \subseteq g, \\ 0 & \text{otherwise.} \end{cases}$$

A critical implementation detail is the use of **subgraph monomorphism** rather than induced subgraph isomorphism. This allows patterns to match non-induced subgraphs and prevents false negatives caused by extra edges in the target graph. The implementation uses `rustworkx` for efficiency, with NetworkX as a fallback.

7 Candidate Generation

A database graph g is included in the candidate set C_q for query q if

$$v_q \leq v_g \quad \text{(component-wise)}.$$

This necessary condition guarantees correctness while enabling aggressive pruning.

8 Experimental Results and Analysis

8.1 Dataset Statistics

Dataset	Total Graphs	Unique Graphs	Duplicates Removed
Mutagenicity	4,337	4,337	0
NCI-H23	40,353	40,039	314

8.2 Best Configuration Performance

Dataset	Avg $\lvert C_q \rvert$	Avg $\lvert R_q \rvert$	Avg s_q	Precision
Mutagenicity	1,240	582	0.4917	49.17%
NCI-H23	10,861	3,916	0.4032	40.32%

8.3 Comprehensive Parameter Analysis

Configuration	min_support	max_size	overlap	Muta s_q	NCI s_q	Combined
output_19_30	10%	5	0.5	0.1909	0.3028	0.2469
output_21_32	10%	5	0.5	0.2166	0.3210	0.2688
output_22_38	15%	5	0.5	0.2244	0.3380	0.2812
output_25_33	15%	6	0.5	0.2596	0.3327	0.2962
output_30_30	20%	5	0.5	0.3073	0.3010	0.3042
output_32_32	20%	5	0.5	0.3278	0.3248	0.3263
output_35_33	25%	6	0.5	0.3515	0.3396	0.3456
output_36_37_0.55	25%	7	0.55	0.3622	0.3778	0.3700
output_38_33	30%	7	0.5	0.3853	0.3396	0.3625
output_38_40_adaptive	30%	8	adaptive	0.3853	0.4002	0.3928
output_25_40	15%	8	0.6	0.2945	0.4002	0.3474
output_46_37	20%	9	0.7	0.4609	0.3793	0.4201
output_46_40_0.7	20%	9	0.7	0.4657	0.4014	0.4336
output_48_38_0.8	20%	9	0.8	0.4817	0.3890	0.4354
output_49_40	**20%**	**9**	**adaptive**	**0.4917**	**0.4032**	**0.4475**

9 Conclusion

This work demonstrates that effective graph indexing can be achieved without exhaustive subgraph enumeration. By carefully exploiting the structural regularities of molecular graphs, we show that a relatively small set of well-chosen subgraph patterns can serve as strong necessary conditions for subgraph isomorphism.

The core of our approach lies in three ideas. First, we restrict pattern mining to paths and simple trees, which capture most meaningful molecular structures while remaining computationally tractable. Second, we explicitly optimize for discrimination rather than raw frequency, using Information Gain to identify patterns that best separate graphs. Third, we enforce diversity through overlap-based filtering with adaptive thresholds, ensuring both coverage and robustness across datasets of different scales.

The resulting index achieves strong filtering performance, with nearly 50% precision on Mutagenicity and over 40% precision on NCI-H23, while remaining scalable to large graph databases. Overall, the approach provides a practical, interpretable, and efficient solution to large-scale graph indexing.