Graph Based Pattern Recognition

Exercise 1

Basis

- Chapters 1 and 2

Submission

- The submission takes place online on ILIAS.
- Solutions to the theory tasks must be submitted as `*.pdf` file. Other formats will not be accepted.
- Source code for the implementation tasks must be submitted as `*.py` files. Source code that cannot be executed will not be accepted.
- Individual submissions or submissions in teams of two are allowed (hand in only one copy per group). In the source code file, include the *names and matriculation numbers* of both group members in the first two lines as comments.

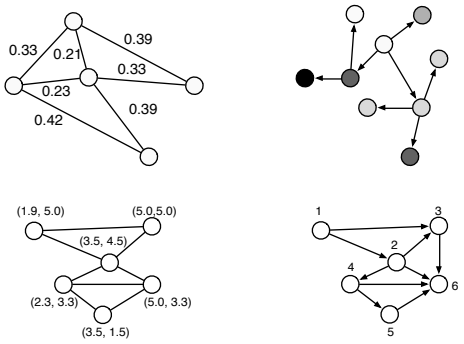
Dates

- Briefing: 01.03.2023
- Submission: 15.03.2023
- Debriefing: 15.03.2023

Theoretical Tasks

1. Summarize the advantages and drawbacks of graph based pattern representation when compared with vectorial approaches.

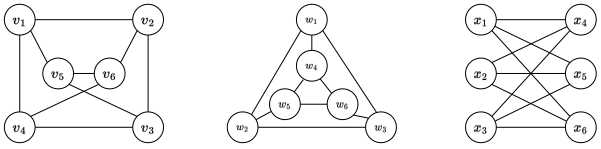
2. Regard the following fours graphs and decide for each of them which special graph type it represents the best.



3. (a) Suppose $N(n, k)$ is the number of non-isomorphic graphs with n nodes and k edges. Find $N(4, 3)$.

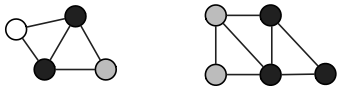
   (b) Define and sketch the set of distinct (i.e., non-isomorphic) unlabeled graphs of size 4. Additionally, find the total number of non-isomorphic graphs of size 4.

4. Determine whether the three unlabeled graphs g_1, g_2 and g_3 given in the figure above are isomorphic. Formally, for any isomorphic graph pair g_i, g_j explicitly determine the isomorphism $f : V_i \to V_j$.



g_1 g_2 g_3

5. Compute the association graph on the following two labeled graphs and find the maximum clique in it (different shades of gray represent different node labels).



6. You will find the following four scientific papers on ILIAS.

   (a) *Shortest-path kernels on graphs*

   (b) *Graph Similarity Features for HMM-Based Handwriting Recognition in Historical Documents*

   (c) *Malware Classification based on Call Graph Clustering*

   (d) *A Graph Matching Based Approach to Fingerprint Classification Using Directional Variance*

   For each paper briefly describe how the underlying patterns/objects are actually represented as a graph $g = (V, E, \mu, \nu)$. In particular, elaborate on the semantic of both nodes and edges as well as the corresponding labeling functions.

Implementation Tasks

In this implementation task, the goal is to implement Ullman's subgraph isomorphism test presented in the lecture notes (Alg. 2).
First, go to the ILIAS's webpage of the course and download/unzip `Exercise_1.zip` in your `PR_Lecture` folder. Then, navigate to `PR_lecture/Exercise_1/ex1.py` and complete the missing part of the source code.

1. Implement the unlabeled/undirected version of Ullman's subgraph isomorphism in `Exercise_1/ex1.py`. In order to adapt Ullman's algorithm to the unlabelled/undirected version, you have to change the initialization of the future match table $F(u, v)$ as follows:

$$F(u,v) = \begin{cases} 1, & \text{if } \deg(u) \leq \deg(v) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

   Lines 5 and 6 of Alg. 2 in the lecture notes set entry $F(w, w')$ to zero, if the edge structure constraint is violated. Formally, this can be accomplished by means of the following test:

```
if A[u, w] != A[v, w']
    F(w, w') = 0
```

2. Perform your implementation of Ullman's test between all pair of graphs in `Exercise_1/graphs`. To this end, create an $N \times N$ binary matrix $\mathbf{R} = (r_{ij})$, where $N = 6$ is the number of graphs in the `PR_lecture/Exercise_1/graphs` directory. Matrix $\mathbf{R}$ contains the results of the subgraph isomorphism test for all pairs of graphs. More formally, entry r_{ij} at position i, j is 1, if graph g_i is a subgraph of graph g_j according to the Ullmans' test. Vice versa, entry r_{ij} equals 0, if the corresponding graphs are not subgraph isomorphic.

   Save your matrix as `ullman_subgraph_isomorphism.csv` in the result folder.

Submission: For the coding part, you must submit a `.zip` file containing the following files. Additionally, include your solution for the theoretical tasks as a `*.pdf` in the same `.zip` file. The file you upload on ILIAS as a `.zip` should be formatted in the following way: `exercise_1_firstname_lastname.zip`.