Chapter 1

Graph Matching

In this chapter we review basic definitions and notations from the graph theory used in this thesis and introduce different forms and formulations of the graph matching problem together with some algorithms for solving them. The classification we use is based on [4], although we concentrated our attention mainly on one specific group of algorithms: those, which consider the graph matching problem as a quadratic optimization problem. Not all algorithms, we present thereby, were initially mentioned in [4], but we also do not cover all of the resent ones due to their quantity. We focus our selfs specially on those, that are important for further reading of the thesis.

1.1 Basic definitions and notation

A *undirected graph* $G = (V, E)$ is defined as a pair of disjoint sets V, E, where $E \subseteq \{\{u, v\}|u, v \in V\}$ [5]. The elements of the set V are called *vertices* or *nodes*[1] and the elements of E are called *edges*. Where it is necessary, we will write $V(G), E(G)$ to refer node and edge sets to the graph G.

The number of nodes in V defines the *size* of a graph G. Two nodes $v_i, v_{i'} \in V$ are called *adjacent*, if there is an edge $e = \{v_i, v_{i'}\} \in E$. A graph, whose each pair of nodes is connected by an edge is called *complete*.

Each graph can be represented by its *adjacency matrix* $A{=}(a_{ij})_{n\times n}$, where

$$a_{ij} = \begin{cases} 1, & \text{if } \{v_i, v_{i'}\} \in E, \\ 0, & \text{otherwise.} \end{cases}$$

The adjacency matrix of undirected graphs is symmetric.

A graph $G' = (V', E')$ is called *subgraph* of the graph G, if $V' \subseteq V$ and $E' \subseteq E$. We use the notation $G' \subseteq G$. We also define graph cut $G \cap G' = (V \cap V', E \cap E')$ and union $G \cup G' = (V \cap V', E \cap E')$.

In case, when each node $v \in V$ of the graph G has an associated attribute d_i, one speaks about attributed graph $G = (V, E, D)$.

Consider two undirected attributed graphs $G^I = (V^I, E^I, D^I)$ and $G^J = (V^J, E^J, D^J)$. We assume the situation, where $|V^I| = n_1$, $|V^J| = n_2$ and $n_1 \leq n_2$. A matching function between G^I and G^J is a total injective map $m : V^I \to V^J$.

[1] We use terms vertex and node further in the text as synonyms

Graph matching problem between G^I and G^J is a problem of finding a map $m : V^I \to V^J$, that maximizes the alignment score $F(G^I, G^J, m)$ and satisfies some additional constraints:

$$m = \operatorname*{argmax}_{\hat{m}} F(G^I, G^J, \hat{m}) \tag{1.1}$$

Based on the constraints the graph matching problems can be divided into two large groups [4]: *exact* and *inexact* graph matching problems.

1.2 Exact graph matching

The first group represents more strict problems, where the mapping between nodes of two graphs should be *edge preserving*. With other words: if $\{v_i, v_{i'}\} \in E^I$, then $\{m(v), m(v_{i'})\} \in E^J$ for all $v_i, v_{i'} \in V^I$. There are several forms of the exact graph matching. The most known one is *graph isomorphism*: two graphs are called *isomorph* ($G^I \simeq G^J$), if the edge preserving map is bijective. This implies, that two graphs have the equal number of nodes. If it is not the case, and the isomorphism holds between the one graph and a node-induced subgraph of the other graphs, the problem is called *subgraph isomorphism*. Its further extension is a isomorphism between subgraphs of the graphs. The last problem can obviously have several solutions, but one is normally interested in finding a common subgraph with maximum number of nodes or edges (*maximum common subgraphs*).

The further simplification of the graph isomorphism is to require an injective edge-preserving mapping, instead of bijective. The correspondences here are still one-to-one, but the second graph may contain additional nodes and edges, comparing to the first graph. Even weaker form is to allow many-to-one mapping, which means that the only one restriction on the mapping m is to be total. On the Fig. Fig one can see examples of different exact graph matching problems.

All problems except graph isomorphism are proofed to be $NP-$complete cite. This can be shown through the reduction of the matching problem to .... The graph isomorphism is currently shown to be $NP-$hard, although there exist polynomial time algorithms for some types of graphs (e.g. for trees [3]).

The most widespread approach to solve exact graph matching problem is branch and bound cite?. It starts from some empty solution and tries in each step to expend it's current known solution based on some rules. If it happens, that a current solution does not fulfill problem's constraints, the solution is cut and algorithm backtracks to the last feasible solution and tries to expand it in other way. The most famous algorithm, that uses this principle is the one by Ullman [9]. Later it was extended and improved (see for example [6] for comparison of algorithms based on the same technique).

The other techniques, that were successfully applied for the (sub)graph isomorphism are decision trees [8, 1, 2] and group theory[7] for maximal subgraph matching.

1.3 Inexact graph matching

Sometimes it is impossible to find isomorphism between two graphs. On the one hand, a reason can be a different structure of the matched graphs because of natural deformations or noise influence, that can occur in some real word applications. On the other hand, solving the graph matching problem exactly can be time or memory consuming. As a consequence one can be interested in solving graph matching problem inexactly. In this case, no edge preserving mapping between the nodes of two graphs is required. At the same time one defines a similarity function between two graph with a given matching to measure quality of this mapping. The aim is to find a mapping, that maximizes similarity value between two graphs:

$$m = \operatorname*{argmax}_{\hat{m}} F(G^I, G^J, \hat{m}) \tag{1.2}$$

In some literature instead of defining the similarity function one speaks about dissimilarity between two graphs cite or matching score cite. The goal in this case is to minimize those values.

An algorithm, that finds the global solution of the problem (1.2), is called *optimal*, otherwise *suboptimal*. The choice, which algorithm to select depends on a specific problem. It should be noted, that an optimal inexact algorithms is not necessary faster than the exact one. On the other hand, suboptimal inexact algorithms often do not have any performance guarantee.

In following we describe a common approaches for solving inexact graph matching problem.

1.4 Graph matching algorithms studied in this thesis

Appendix A

Bibliography

[1] Efficient graph matching for video indexing. In *Graph based representations in pattern recognition*, Computing supplementum, pages 53–62. Springer-Verlag, 1998.

[2] Video sequence matching via decision tree path following. *Pattern recognition letters*, pages 479–492, 2001.

[3] A. V. Aho and J. E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman Publishing Co., Inc., 1st edition, 1974.

[4] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty Years of Graph Matching in Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[5] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 2000.

[6] J. Lee, W.-S. Han, R. Kasperovics, and J.-H. Lee. An in-depth comparison of subgraph isomorphism algorithms in graph databases. In *Proceedings of the 39th international conference on Very Large Data Bases*, PVLDB'13, pages 133–144. VLDB Endowment, 2013.

[7] B. D. McKay and A. Piperno. Practical graph isomorphism, II. *CoRR*, 2013.

[8] B. Messmer and H. Bunke. A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, 32(12):1979–1998, 1999.

[9] J. R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42, 1976.

Declaration of Authorship

I confirm that this Master's thesis is my own work and I have documented all sources and material used. This thesis was not previously presented to another examination board and has not been published.

Heidelberg, den (Datum) ....