Chapter 7

EXACT AND INEXACT GRAPH MATCHING: METHODOLOGY AND APPLICATIONS

Kaspar Riesen

Institute of Computer Science and Applied Mathematics, University of Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland
riesen@iam.unibe.ch

Xiaoyi Jiang

Department of Mathematics and Computer Science, University of Münster
Einsteinstrasse 62, D-48149 Münster, Germany
xjiang@math.uni-muenster.de

Horst Bunke

Institute of Computer Science and Applied Mathematics, University of Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland
bunke@iam.unibe.ch

Abstract

Graphs provide us with a powerful and flexible representation formalism which can be employed in various fields of intelligent information processing. The process of evaluating the similarity of graphs is referred to as graph matching. Two approaches to this task exist, viz. exact and inexact graph matching. The former approach aims at finding a strict correspondence between two graphs to be matched, while the latter is able to cope with errors and measures the difference of two graphs in a broader sense. The present chapter reviews some fundamental concepts of both paradigms and shows two recent applications of graph matching in the fields of information retrieval and pattern recognition.

Keywords: Exact and Inexact Graph Matching, Graph Edit Distance, Information Retrieval by means of Graph Matching, Graph Embedding via Graph Matching

© Springer Science+Business Media, LLC 2010
1. Introduction

After many years of research, the fields of pattern recognition, machine learning and data mining have reached a high level of maturity [4]. Powerful methods for classification, clustering, information retrieval, and other tasks have become available. However, the vast majority of these approaches rely on object representations given in terms of feature vectors. Such object representations have a number of useful properties. For instance, the dissimilarity, or distance, of two objects can be easily computed by means of the Euclidean distance. Moreover, a large number of well-established methods for data mining, information retrieval, and related tasks in intelligent information processing are available. Recently, however, a growing interest in graph-based object representation can be observed [16]. Graphs are powerful and universal data structures able to explicitly model networks of relationships between substructures of a given object. Thereby, the size as well as the complexity of a graph can be adopted to the size and complexity of a particular object (in contrast to vectorial approaches where the number of features has to be fixed beforehand).

Yet, after the initial enthusiasm induced by the “smartness” and flexibility of graph representations in the late seventies, a number of problems became evident. First, working with graphs is unequally more challenging than working with feature vectors, as even basic mathematic operations cannot be defined in a standard way, but must be provided depending on the specific application. Hence, almost none of the common methods for data mining, machine learning, or pattern recognition can be applied to graphs without significant modifications.

Second, graphs suffer from of their own flexibility. For instance, computing the distances of a pair of objects, which is an important task in many areas, is linear in the number of data items in the case where vectors are employed. The same task for graphs, however, is much more complex, since one cannot simply compare the sets of nodes and edges, which are generally unordered and of different size. More formally, when computing graph dissimilarity or similarity one has to identify common parts of the graphs by considering all of their subgraphs. Regarding that there are $O(2^n)$ subgraphs of a graph with $n$ nodes, the inherent difficulty of graph comparisons becomes obvious.

Despite adverse mathematical and computational conditions in the graph domain, various procedures for evaluating proximity, i.e. similarity or dissimilarity, of graphs have been proposed in the literature [15]. The process of evaluating the similarity of two graphs is commonly referred to as graph matching. The overall aim of graph matching is to find a correspondence between the nodes and edges of two graphs that satisfies some, more or less, stringent constraints. That is, by means of the graph matching process similar substructures in one graph are mapped to similar substructures in the other graph. Based on