Pattern Recognition Letters 33 (2012) 811–825

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Towards the unification of structural and statistical pattern recognition

Horst Bunke, Kaspar Riesen *

Institute of Computer Science and Applied Mathematics, University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland

ARTICLE INFO

Article history:
Available online 4 May 2011

Keywords:
Statistical pattern recognition
Structural pattern recognition
Graph classification
Graph clustering
Graph kernel
Graph embedding

ABSTRACT

The field of pattern recognition is usually subdivided into the statistical and the structural approach. Structural pattern recognition allows one to use powerful and flexible representation formalisms but offers only a limited repertoire of algorithmic tools needed to solve classification and clustering problems. By contrast, the statistical approach is mathematically well founded and offers many tools, but provides a representation formalism that is limited in its power and flexibility. Hence, both subfields are complementary to each other. During the last three decades several efforts have been made towards bridging the gap between structural and statistical pattern recognition in order to profit from the benefits of each approach and eliminate the drawbacks. The present paper reviews some of these attempts made towards the unification of structural and statistical pattern recognition and analyzes the progress that has been achieved.

1. Introduction

The discipline of pattern recognition is usually divided into the *statistical* and the *structural approach*.[1] In statistical pattern recognition, objects or patterns are given by feature vectors. Hence, a pattern is formally represented as a vector consisting of n measurements, or feature values, and can be understood as a point in the n-dimensional real space, i.e. $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Representing patterns by feature vectors $\boldsymbol{x} \in \mathbb{R}^n$ offers a number of useful properties, in particular, the mathematical wealth of operations available in a vector space. For example, quantities such as the sum, the product, the mean, or the distance of two entities are well defined in a vector space and, moreover, can be efficiently computed. The convenience and low computational complexity of algorithms that use feature vectors as their input have eventually resulted in a rich repository of algorithmic tools for statistical pattern recognition (Duda et al., 2000; Bishop, 2008). However, the use of feature vectors implicates two limitations. First, as vectors always represent a predefined set of features, all vectors in a given application have to preserve the same length regardless of the size or complexity of the corresponding objects. Second, there is no direct possibility to describe binary or higher-order relationships that might exist among different parts of a pattern. These two drawbacks are severe, particularly when the patterns under consideration are characterized by complex structural relationships rather than the statistical distribution of a fixed set of features.

Structural pattern recognition, by contrast, is based on symbolic data structures, such as *strings*, *trees*, or *graphs* for pattern representation. Graphs, which consist of a finite set of nodes connected by edges, is the most general representation formalism, and the other data types commonly used in structural pattern recognition are special cases of graphs. In particular, strings and trees are simple instances of graphs. In the remainder of the present paper we will focus on graphs. But the reader should keep in mind that strings and trees are always included as special cases.

The above mentioned drawbacks of feature vectors, namely the size constraint and the lacking ability to represent structural relationships, can be overcome by graph based representations (Conte et al., 2004). In fact, graphs are not only able to describe properties of an object, but also binary relationships among different parts of the underlying object, by means of edges. Note that these relationships can be of various nature, viz. spatial, temporal, or conceptual. Moreover, graphs are not constrained to a fixed size, i.e. the number of nodes and edges is not limited a priori and can be adapted to the size or the complexity of each individual object under consideration.

One drawback of graphs arises from the fact that there is little mathematical structure in the domain of graphs. For example, computing the (weighted) sum or the product of a pair of entities, which are elementary operations required in many classification and clustering algorithms, is not possible in the domain of graphs, or is at least not defined in a standardized way. Due to these problems, we observe a lack of algorithmic tools for graph based pattern recognition.

* Corresponding author.
E-mail address: riesen@iam.unibe.ch (K. Riesen).

[1] Note that there exists also the *syntactical approach* where the patterns are encoded as elements of a formal language (Fu, 1982). Methods from formal language theory can then be used in order to discriminate between different classes of objects. The present paper, however, considers the statistical and structural approach only.

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.04.017

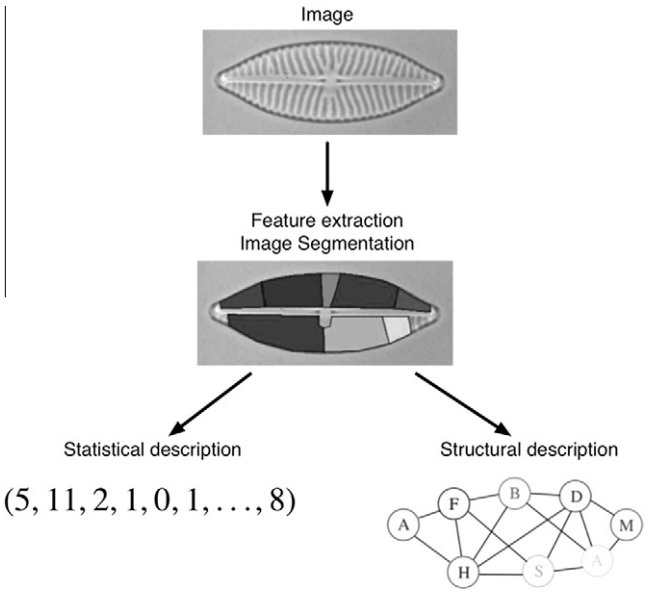

Fig. 1. The statistical and structural description of a diatom image.

In Fig. 1 the statistical and structural approach are illustrated with a vector and a graph representing the same diatom image.[2] In the statistical approach the image of the diatom is formally represented by a fixed set of numerical features describing, for example, properties of the object's shape or its internal texture, while – in this example – in the structural approach a region adjacency graph captures information about local regions together with their spatial relationships. Moreover, adding labels to the nodes and edges of the graph, one can not only describe the shape or texture of local regions, but also include properties, such as the length or the shape of the common boundaries between two adjacent regions.

Summarizing, in terms of their respective advantages and drawbacks, feature vectors and graphs have complementary properties (cf. Table 1). The question of how to combine these complementary properties in order to get the best out of both worlds has been a focus of attention in the pattern recognition community for a long time. The objective of the present article is to review some of the efforts undertaken to bridge the gap between vector based and graph based pattern recognition. We distinguish two different periods in our review, called the *classical* and the *modern period*.

In the classical period (1980's and 1990's in our review) the focus of graph based pattern recognition was on transferring some important concepts from the vector space to the graph domain. In this era various tools, originally developed for vectors, have been made applicable for graphs. However, all these attempts were on a case-by-case basis. There were no general methods known that would allow one to do such a transformation in a systematic, or principled, way. Section 2 of the present paper reviews some attempts that are characteristic for this period.

In the last decade (which we refer to as the modern period), a seminal paradigm shift in graph based pattern recognition could be observed. The basic idea of the new paradigm is that rather than working directly on graphs and defining handcrafted operations and quantities in the graph domain, the graphs are mapped into a vector or dot product space. Subsequently, all the operations needed in the pattern recognition process are carried out in the target space of this mapping rather than in the original graph domain. The benefit of such a graph embedding is that it instantly makes available all algorithmic tools, originally developed for vectors or dot product spaces, to graphs. In Section 3 several prominent examples of this approach are reviewed. In an experimental evaluation (Section 4) we show how the methods from the modern period can be used to improve the performance of various classification tasks.

[2] Diatoms are unicellular algae found in humid places where light provides the basis for photosynthesis. The identification of diatoms is useful for various applications such as environmental monitoring and forensic medicine (du Buf and Bayer, 2002).

Table 1
Feature vectors and graphs have complementary properties.

	Vectors	Graphs
Representational power	–	+
Available tools	+	–

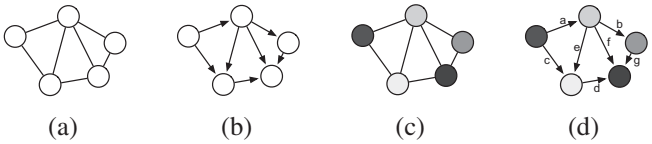

Fig. 2. Different kinds of graphs: (a) undirected and unlabeled, (b) directed and unlabeled, (c) undirected with labeled nodes (different shades of gray represent different labels), (d) directed with labeled nodes and edges.

2. The classical period – Working in the graph domain

A first step towards the unification of statistical and structural pattern recognition has been proposed in (Fu, 1986) by augmenting nodes and edges of a graph by attributes, i.e. feature vectors. This led to the standard definition of graphs used in pattern recognition today.

Definition 1 (*Graph*). Let L_V and L_E be a finite or infinite label set for nodes and edges, respectively. A graph g is a four-tuple $g = (V, E, \alpha, \beta)$, where V is the finite set of nodes, $E \subseteq V \times V$ is the set of edges, $\alpha : V \to L_V$ is the node labeling function, and $\beta : E \to L_E$ is the edge labeling function.

In the remainder of this paper, $\mathcal{G}$ represents the set of all graphs over the label alphabets L_V and L_E. Definition 1 allows us to handle arbitrarily structured graphs with unconstrained labeling functions. For example, the labels for both nodes and edges can be given by the set of integers $L = \{1, 2, 3, \ldots\}$, the vector space $L = \mathbb{R}^n$, or a finite set of symbolic labels $L = \{x, y, z, \ldots\}$. Given that the nodes and/or the edges are labeled, the graphs are referred to as *labeled* graphs. *Unlabeled* graphs are obtained as a special case by assigning the same label ε to all nodes and edges, i.e. $L_V = L_E = \{\varepsilon\}$. Edges are given by pairs of nodes (u, v), where $u \in V$ denotes the source node and $v \in V$ the target node of a directed edge. *Directed* graphs directly correspond to the definition above. In addition, the class of *undirected* graphs can be modeled by inserting a reverse edge $(v, u) \in E$ for each edge $(u, v) \in E$ with identical labels, i.e. $\beta(u, v) = \beta(v, u)$. In Fig. 2 some graphs (directed/undirected, labeled/unlabeled) are shown.

In the classical period of graph based pattern recognition the overall aim was to modify algorithms, originally developed for vector based pattern representations, such that they become applicable to graphs. To this end, several mathematical concepts had to be transferred from the vector space to the graph domain. In this section, three prominent and important examples of such concepts are reviewed and discussed, namely graph distance (Section 2.1), median graph and weighted mean of graphs (Section 2.2).