Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 39 (2006) 573–586
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Recovery of missing information in graph sequences by means of reference pattern matching and decision tree learning

Horst Bunke[a,*], Peter Dickinson[b], Christophe Irniger[a], Miro Kraetzl[b]

[a]*Institute of Computer Science and Applied Mathematics, University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland*
[b]*Intelligence, Surveillance and Reconnaissance Division, Defence Science and Technology Organisation, Edinburgh SA 5111, Australia*

Received 13 October 2005

Abstract

Algorithms for the analysis of graph sequences are proposed in this paper. In particular, we study the problem of recovering missing information and predicting the occurrence of nodes and edges in time series of graphs. Two different recovery schemes are developed. The first scheme uses reference patterns that are extracted from a training set of graph sequences, while the second method is based on decision tree induction. Our work is motivated by applications in computer network analysis. However, the proposed recovery and prediction schemes are generic and can be applied in other domains as well.

Keywords: Graph sequence analysis; Recovery of missing information; Computer network analysis; Machine learning; Decision tree classifier; Reference pattern matching

1. Introduction

The aim of graph matching is to find an assignment of the nodes and edges of two given graphs such that some optimality criterion is satisfied. Special instances of the graph matching problem allow us to compute, among other quantities, distance measures between two given graphs. A large body of work on graph matching, dealing with both theory and applications, has been published in the literature. For representative collections of recent work see Refs. [1–4]. However, almost all published papers address the case of only two graphs being matched with each other. In this paper, we provide an extension and address the analysis of graph sequences.

The work described in this paper is motivated by applications in computer network monitoring. The basic idea is to represent a computer network by a graph, where the clients and servers are modelled by nodes and the physical connections correspond to edges. If the state of the network is captured at regular points in time and represented as a graph, a time series of graphs is obtained that formally represents the network. In our previous work we have developed various procedures for the detection of anomalous events and network behaviour [5]. These procedures are based on the observation that abnormal network behaviour corresponds to large distance between two consecutive graphs in a time series. In the current paper we address a different problem, viz. the recovery of incomplete network knowledge. Due to various reasons it may happen that the state of a network node or a network link cannot be properly captured during network monitoring. This means that it is not known whether or not a certain node or an edge is present in the graph sequence at a certain point in time. In this paper we describe procedures that are able to recover missing information of this kind. These procedures are capable of making a decision as to the presence or absence of such network nodes and edges. Information recovery procedures of this kind can also be used to predict, at time t, whether a certain node or a certain link will be present, i.e. active, in the network at the next point in time, $t + 1$. Such procedures are

* Corresponding author. Tel.: +41 31 631 44 51.
E-mail addresses: bunke@iam.unibe.ch (H. Bunke), Peter.Dickinson@dsto.defence.gov.au (P. Dickinson), irniger@iam.unibe.ch (C. Irniger), Miro.Kraetzl@dsto.defence.gov.au (M. Kraetzl).

0031-3203/$30.00
doi:10.1016/j.patcog.2005.10.011

useful in computer network monitoring in situations where one or more network probes have failed. Here the presence, or absence, of certain nodes and edges is not known. In these instances, the network management system would be unable to compute an accurate measurement of network change and would thus be unable to recognize abnormal network behaviour. The techniques described in this paper can be used to determine the likely status of this missing data and hence improve reliability of abnormal change detection.

There exist a number of papers that are concerned with modelling the topology of computer networks, especially the Internet. In Ref. [6] a number of heuristics are described that seem to be suitable to create an approximate map of the Internet from a number of probes. In Refs. [7,8] it is attempted to build a formal model of the Internet topology, for example in terms of a power-law random graph [9]. In contrast with these works, we do not intend to build any model or any map and assume that the topology of the underlying network is known. This assumption is justified in applications that involve intranets. The goal of our work is the recovery of missing information in such an environment. That is, we want to make a statement as to whether or not a particular node or link with unknown status is active or not at a certain point in time.

Analysis of time series and prediction is a field that has been intensively studied in the literature. Particular attention has been paid to problems such as time series segmentation [10], retrieval of sequences or partial sequences [11], indexing [12], classification of time series [13], detection of frequent subsequences [14], periodicity detection [15] and prediction [16–19]. However, in all these previous works a time series is given in terms of symbols, numbers or vectors [20]. In the current paper we go a step further and consider prediction schemes that operate on sequences of graphs. Node as well as edge prediction in time series of graphs will be addressed.

The paper is organized as follows. Basic terminology and notation will be introduced in the next section. Then, in Sections 3 and 4, we will describe two novel information recovery and prediction procedures. Results of a number of experiments with these new schemes will be presented in Section 5. Finally, conclusions will be drawn in Section 6.

2. Basic concepts and notation

A labeled graph is a 4-tuple, $g = (V, E, \alpha, \beta)$, where V is the finite set of nodes, $E \subseteq V \times V$ is the set of edges, $\alpha : V \to L$ is the node labelling function, and $\beta : E \to L'$ is the edge labeling function, with L and L' being the set of node and edge labels, respectively. In this paper we focus our attention on a special class of graphs that are characterized by unique node labels. That is, for any two nodes, $x, y \in V$, if $x \neq y$ then $\alpha(x) \neq \alpha(y)$. Properties of this class of graphs have been studied in Ref. [21]. In particular it has been shown that problems such as graph isomorphism, subgraph isomorphism, maximum common subgraph, and graph edit distance computation can be solved in time that is only quadratic in the number of nodes of the larger of the two graphs involved.

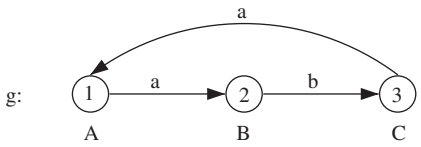

Fig. 1. A graph with unique node labels.

To represent graphs with unique node labels in a convenient way, we drop set V and define each node in terms of its unique label. Hence a graph with unique node labels can be represented by a 3-tuple, $g = (L, E, \beta)$ where L is the set of node labels occurring in g, $E \subseteq L \times L$ is the set of edges, and $\beta : E \to L'$ is the edge labeling function [21]. The terms "node label" and "node" will be used synonymously in the remainder of this paper.

As an example, consider graph g in Fig. 1. Using traditional notation, this graph is represented by the 4-tuple $g = (V, E, \alpha, \beta)$, where $V = \{1, 2, 3\}$; $E = \{(1, 2), (2, 3), (3, 1)\}$; $\alpha : 1 \mapsto A, 2 \mapsto B, 3 \mapsto C$; $\beta : (1, 2) \mapsto a, (2, 3) \mapsto b, (3, 1) \mapsto a$. Because all node labels are unique, we can alternatively represent graph g by the 3-tuple $g = (L, E, \beta)$, where $L = \{A, B, C\}$; $E = \{(A, B), (B, C), (C, A)\}$; $\beta : (A, B) \mapsto a, (B, C) \mapsto b, (C, A) \mapsto a$.

In this paper we will consider time series of graphs, i.e. graph sequences, $s = g_1, g_2, \ldots, g_N$. The notation $g_i = (L_i, E_i, \beta_i)$ will be used to represent individual graph g_i in sequence s; $i = 1, \ldots, N$. Motivated by the computer network analysis application considered in this paper, we assume the existence of a universal set of node labels, or nodes, $\mathscr{L}$, from which all node labels that occur in a sequence s are drawn. That is, $L_i \subseteq \mathscr{L}$ for $i = 1, \ldots, N$ and $\mathscr{L} = \bigcup_{i=1}^{N} L_i$.[1]

As an example, consider sequence $s = g_1, g_2, g_3$, where graphs g_1, g_2 and g_3 are depicted in Fig. 2. These graphs are formally represented as follows:

- $g_1 = (L_1, E_1, \beta_1)$; $L_1 = \{A, B, C\}$; $E_1 = \{(A, B), (B, C), (C, A)\}$;
- $g_2 = (L_2, E_2, \beta_2)$; $L_2 = \{A, B, D\}$; $E_2 = \{(A, B), (B, D), (D, A)\}$;
- $g_3 = (L_3, E_3, \beta_3)$; $L_3 = \{A, D, C\}$; $E_3 = \{(A, D), (D, C), (C, A)\}$.

We assume that $\beta_1 = \beta_2 = \beta_3 = const$ and omit the edge labels in Fig. 2. In this example we have $\mathscr{L} = \{A, B, C, D\}$.

Given a time series of graphs, $s = g_1, g_2, \ldots, g_N$, and its corresponding universal set of node labels, $\mathscr{L}$, we can repre-

[1] In the computer network analysis application $\mathscr{L}$ will be, for example, the set of all unique IP host addresses in the network. Note that in one particular graph, g_i, usually only a subset is actually present. In general, $\mathscr{L}$ may be any finite or infinite set.