Chapter 10

Data Driven Encoding of Structures and Link Predictions in Large XML Document Collections

Markus Hagenbuchner
University of Wollongong, Australia

Chung Tsoi
Macau University of Science and Technology, China

Shu Jia Zhang
University of Wollongong, Australia

Milly Kc
University of Wollongong, Australia

ABSTRACT

In recent years there have been some significant research towards the ability of processing related data, particularly the relatedness among atomic elements in a structure with those in another structure. A number of approaches have been developed with various degrees of success. This chapter provides an overview of machine learning approaches for the encoding of related atomic elements in one structure with those in other structures. The chapter briefly reviews a number of unsupervised approaches for such data structures which can be used for solving generic classification, regression, and clustering problems. We will apply this approach to a particularly interesting and challenging problem: The prediction of both the number and their locations of the in-links and out-links of a set of XML documents. In this problem, we are given a set of XML pages, which may represent web pages on the Internet, with in-links and out-links. Based on this training dataset, we wish to predict the number and locations of in-links and out-links of a set of XML documents, which are as yet not linked to other existing XML documents. To the best of our knowledge, this is the only known data driven unsupervised machine learning approach for the prediction of in-links and out-links of XML documents.

DOI: 10.4018/978-1-61350-356-0.ch010
INTRODUCTION

The traditional approach to processing data is by using numeric vectors of some fixed dimension as a means of representing the data, where the vector elements describe features of an associated data item (Haykin, 1994). Such representation is suitable when the data items are independent or when any existing dependencies are not relevant for the solving of a given problem. A common situation in which data items are dependent is in time series information processing (Haykin, 1994). In this case, a data entry may depend on another data entry which occurred in a prior time instance, and such dependency may be relevant for a given problem (Haykin, 1994). Data items which are sequentially organized are referred to as sequences. For example, in financial forecasting or in speech recognition, it is important that such dependency can be represented and encoded (Haykin, 1994). Note that unlike data vectors which are often of the same dimension, in the cases where they are not of the same dimension, then the technique of "zero padding" is used to ensure that all data vectors are of the same dimension (Haykin, 1994), there is not normally a restriction to the length of sequences. When data items are “related” to several other data items, then such dependencies are popularly described by a graph, hence such instances are referred to as graph data structures. Note that here, the data items are “related” in that the data objects, represented by nodes of a graph, are connected by links, directed or un-directed, cyclic, or acyclic, which express the relationships among the data objects. A very large number of practical problems can be represented as a graph data structure. For example, problems in molecular chemistry, image processing, computer security systems, weather forecasting, etc. can be addressed by processing the dependencies which are represented in graph structured data. Since sequences are a special case of graphs (graphs with an out-degree of 1), and since vectors are also a special case of graphs (a graph with a single node), graph data structures provide the most generic means for data representation. Any approach capable of encoding graphs can also solve time series problems and problems in which the data can be represented in vector spaces.

Simple graphs consist of two basic elements, namely, nodes and the binary relations called links. The nodes represent atomic entities of a domain. For example, a document may be represented by a node. A node may be labeled to describe the associated entity. For example, a node label may contain descriptive features of the associated document. These features could be the number of words in the document, the occurrence of certain words in the document, the frequency with which certain words occurred in the document, etc. A link exists between any two nodes if these nodes are “related” in some way. For example, a document may contain a hyperlink to another document. In this case, there is a directed link from the node representing the source document to the node which represents the target document of the hyperlink. A link may also be labeled in order to describe its properties. For example, a label associated with a link may indicate the strength of a link. Links may be directed (such as in the hyperlink case) or undirected (such as the links indicating atomic bindings of a molecule). The total number of all outgoing links of a node is called the outdegree of the node, while the total number of all incoming links is called the indegree of a node. In the case of undirected links, this is referred to as the degree of a node. The size of a graph $G$ is the total number of nodes in $G$ and is denoted by $n=|G|$. If there exists a path along the links in $G$ from one node to another then these nodes are said to be connected, otherwise they are disconnected. The shortest path along the link structure between a pair of nodes in $G$ is called the distance between these nodes.\(^1\)

The importance of an ability to process graph structured information is highlighted by the fact that most documents are created electronically nowadays. The content of a document is stored as