Edit Distance Based Kernel Functions for Attributed Graph Matching

Michel Neuhaus and Horst Bunke

Dept. of Computer Science, University of Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{mneuhaus, bunke}@iam.unibe.ch

Abstract. In this paper we propose the use of a simple kernel function based on the graph edit distance. The kernel function allows us to apply a wide range of statistical algorithms to the problem of attributed graph matching. The function we describe is simple to compute and leads to several convenient interpretations of geometric properties of graphs in their implicit vector space representation. Although the function is not generally positive definite, we show in experiments on real-world data that the kernel approach may result in a significant improvement of the graph matching and classification performance using support vector machines and kernel principal component analysis.

1 Introduction

In recent years results from statistical learning theory [1] have successfully been applied to various pattern recognition problems. The Support Vector Machine (SVM) classifier, among other kernel machines, has performed particularly well on a number of data sets [2,3]. The development of specific kernel functions such as diffusion kernels, convolutional kernels, and marginalized kernels lead to the application of kernel machines to the structural domain of graphs and strings [4,5,6,7,8]. In contrast to feature vectors, graphs allow for a more powerful representation of structured data. To measure the similarity of graphs, various formalisms have been proposed, ranging from exact methods such as subgraph isomorphism and maximum common subgraph computation to error-tolerant methods based on continuous optimization theory and the spectral decomposition of graph matrices [9,10,11,12,13]. Another graph matching approach that has become quite popular is the concept of the graph edit distance [14,15,16]. In graph edit distance, the main idea is to model graph difference through a sequence of edit operations, which leads to a dissimilarity measure on graphs.

In the present paper, we propose to use kernel functions based on the edit distance to classify graphs, instead of performing a simple nearest-neighbor classification in the original graph space. In Sect. 2 and 3, graph edit distance and kernel machines are briefly reviewed. The proposed kernel function is described in Sect. 4. Experimental results are presented in Sect. 5, and some concluding remarks follow in Sect. 6.
2 Graph Edit Distance

In graph matching, patterns represented by graphs are classified based on their structural similarity. For this purpose, several similarity measures have been defined for different classes of graphs. In the following, we consider the case of numerically labeled graphs. That is, nodes and edges may contain vector attributes to make the graph representation more powerful. One of the most common similarity measures for general attributed graphs is the graph edit distance [14][15][16]. The key idea is to describe variations in the structure of graphs by sequences of basic edit operations. A standard set of edit operations consists of insertion, deletion, and substitution operations for nodes and edges. To allow for application-specific similarity measures, it is common to introduce edit costs for edit operations, so that the edit distance of two graphs can be defined by the least expensive sequence of edit operations, or edit path, that transforms one graph into the other.

After computing distances of an input graph to a number of known graphs, the classification of the input pattern is often performed by means of a nearest-neighbor approach, where the neighborhood of every known graph, according to the edit distance, is associated with its class. Theoretical results and practical experiments [17][18] suggest that nearest-neighbor algorithms are appropriate for pattern recognition problems of various difficulty. Yet, the distances of the input graph to some or all of the known graphs could also be regarded as a vector representation of the graph, thus embedding the input graph in a vector space. As a major advantage, a vector space representation allows us to apply a large number of powerful statistical algorithms to the classification problem. In the remainder of this paper, we will explore the feasibility of using simple kernel functions to apply such algorithms to attributed graphs. In this context, the vector space embedding of the graphs follows implicitly and need not be explicitly constructed.

3 Kernel Machines

The recent widespread use of kernel machines is primarily due to their successful application to various pattern recognition problems [2][3]. Instead of solving a classification problem in its original domain, the basic idea is to map patterns into a Hilbert space (vector space with an inner product) and find a solution for the vector representation of the patterns [1][19]. While linear algorithms in the original pattern space may be efficient and easy to compute, they are unable to take non-linear pattern relations into account. By using non-linear transformations when mapping from the original domain into the vector space, we obtain non-linear generalizations of well-known linear algorithms. Cover’s theorem on the separability of patterns [20] states that a complex pattern classification problem cast non-linearly into a high-dimensional space is more likely to be linearly separable than in a low-dimensional space. A principal component analysis (PCA) in the original pattern space, for instance, will only detect linear struc-