Discovery of Inexact Concepts from Structural Data

Lawrence B. Holder and Diane J. Cook
Department of Computer Science Engineering
University of Texas at Arlington
Box 19015, Arlington, TX 76019
Email: holder@cse.uta.edu, cook@cse.uta.edu

I. Introduction

Automated knowledge discovery is essential for extracting information from databases [2]. However, complex databases have moved from the simplistic attribute-value representation assumed by recent knowledge-discovery systems to a structural data representation that reflects the relationships among objects. Discovering concepts in this structural data requires the identification of common substructures within the data. The motivation for this process is not merely to find substructures capable of compressing the data by abstracting instances of the substructure, but also to identify conceptually interesting substructures that enhance the interpretation of the data. Substructure discovery is the process of identifying concepts describing interesting and repetitive substructures within structural data. Once discovered, the substructure concept can be used to simplify the data by replacing instances of the substructure with a pointer to the newly discovered concept. The discovered substructure concepts allow abstraction over detailed structure in the original data and provide new, relevant attributes for interpreting the data. Iteration of the substructure discovery and replacement process constructs a hierarchical description of the structural data in terms of the discovered substructures. This hierarchy provides varying levels of interpretation that can be accessed based on the goals of the data analysis.

Discovering interesting concepts requires a cognitive evaluation component and the ability

to consider instances of the concept that do not exactly match the concept definition. We describe the SUBDUE system that utilizes psychologically-motivated heuristics and an inexact graph match to discover substructures which occur often in the data, but not always in the same form. This inexact substructure discovery can be used to formulate fuzzy concepts, compress the data description, and discover interesting structures in data that are found either in an identical or in a slightly convoluted form. Examples from the domains of scene analysis and chemical analysis demonstrate the benefits of the discovery technique.

II. SUBDUE

The SUBDUE system [3] discovers substructure in structured data based on four psychologically motivated heuristics: cognitive savings, connectivity, compactness and coverage. The *cognitive savings* of a substructure represents the net reduction in complexity when considering both the reduction in complexity of the input data after replacing each substructure instance by a single conceptual entity and the gain in complexity associated with the definition of the new substructure. Substructure *compactness* is defined as the ratio of the number of edges to the number of nodes in a graph representation of the substructure. *Connectivity* measures the amount of external connection in the instances of the substructure. *Coverage* measures the fraction of structure in the input data described by the substructure. SUBDUE evaluates a substructure based on the product of these four heuristic values, each weighted by a user-supplied exponent.

The substructure discovery algorithm used by SUBDUE is a computationally constrained best-first search guided by the heuristics. SUBDUE represents structured data as a directed graph. The algorithm begins with the substructure matching a single node in the graph. Each iteration through the algorithm selects the heuristically-best substructure and expands the

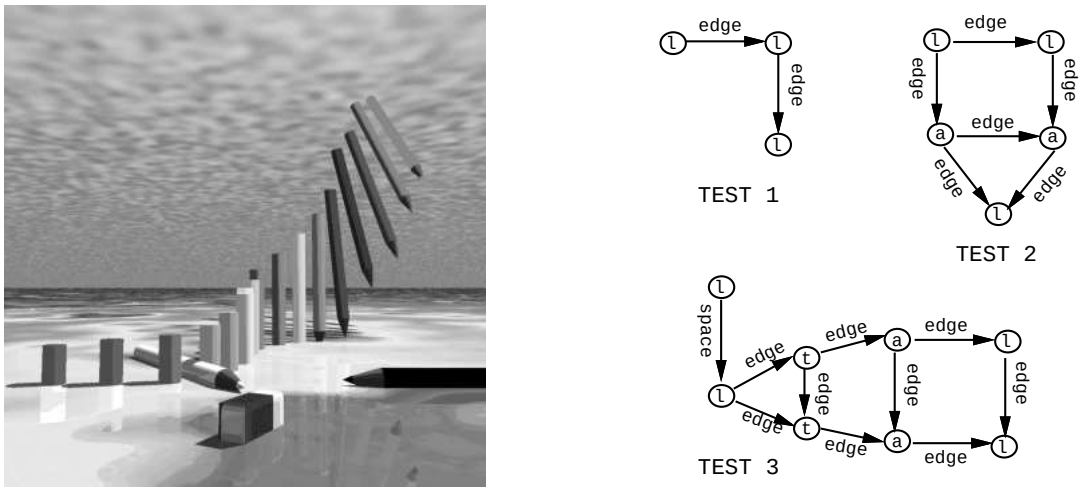

Figure 1: Scene analysis example.

instances of the substructure by one neighboring edge in all possible ways. The algorithm searches for the heuristically-best substructure until all possible substructures have been considered or the amount of computation exceeds a given limit. Holder *et al.* [4] describe the SUBDUE system in more detail.

III. Inexact Graph Match

Although exact structure match can be used to find many interesting substructures, many of the most interesting substructures show up in a slightly different form throughout the data. These differences may be due to noise and distortion, or may just illustrate slight differences between instances of the same general class of structures. Consider the image shown in Figure 1. The pencil and the cube would make ideal substructures in the picture, but an exact match algorithm may not consider these as strong substructures because they rarely occur in the same form and orientation throughout the picture.

We adopt the approach to inexact graph match developed by Bunke and Allermann [1], where each distortion of a graph is assigned a cost. A distortion is described in terms of basic

transformations such as deletion, insertion, and substitution of nodes and edges. SUBDUE finds the mapping between a substructure definition and instance that minimizes the cost, or amount of distortion, necessary to make the graphs isomorphic. The order of complexity of the inexact graph match is equivalent to that of exact graph match; however, the inexact match offers the additional benefit of assigning a similarity measure to each possible mapping. Integrating the inexact graph match into SUBDUE is accomplished by including as instances of a substructure all subgraphs in the input data that match the substructure definition with a match cost within a user-supplied threshold. The contributions of individual instances to the four heuristics are weighted according to their match cost.

IV. EXAMPLE 1 – SCENE ANALYSIS

Images provide a rich source of structure. Images that humans encounter, both natural and synthesized, have many structured subcomponents that draw our attention and that help us to interpret the data or the scene we are viewing. Applying SUBDUE to image data, we extract edge information from the image and construct a graph representing the scene. The graph representation consists of two types of arcs (*edge* and *space*), and three types of nodes (L, T and A). The *edge* arcs represent lines in the image, and the *space* arcs connect vertices from two objects that are closest together. Node labels come from the Waltz labeling [5] of the junctions of lines in the image, where A stands for an arrow junction.

Figure 1 gives an example of a scene which contains many similar substructures. Using the graph representation described above, a line drawing of this image would consist of rectangles (pencils stuck in the surface), partially occluded rectangles (overlapping pencils), and rectangles with triangles on the end (pencils with sharp points). In order to compare the types of substructures found, a variety of heuristic weights and inexact match thresholds

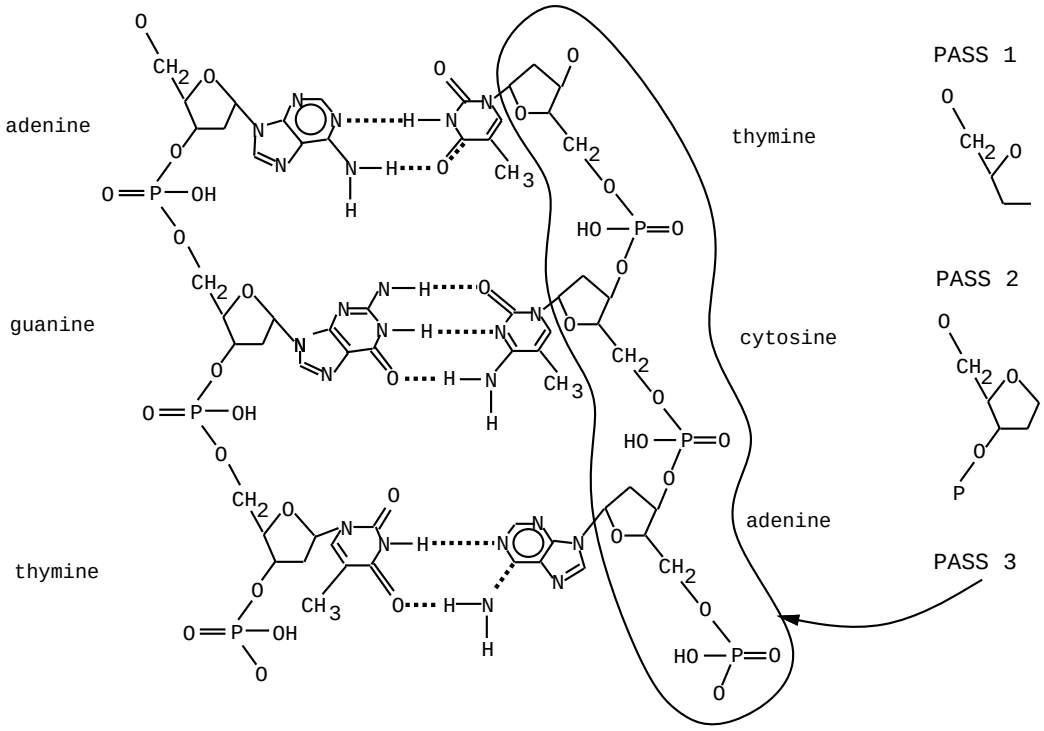

Figure 2: Portion of DNA molecule and discovered substructure.

were tested. Figure 1 shows the highest-valued substructure for three sets of parameter values. In Test 1, all heuristics are equally weighted. In Test 2, connectivity is stressed. In Test 3, connectivity is still emphasized, but the amount of variation between instances of the substructure is minimized. The results indicate that the amount of noise and the weights of the heuristics greatly affect the discovered substructure. Each result may be desired, based on whether the goal of discovery is image compression or interestingness.

V. Example 2 – Chemical Analysis

Identification of the common and interesting substructures within chemical compounds can benefit scientists by identifying recurring components, simplying the data description, and focusing on substructures that stand out and possibly merit additional attention.

Figure 2 shows a portion of DNA consisting of two chains of double helix, using three

pairs of bases which are held together by hydrogen bonds. For this example, we tested the ability of SUBDUE to find a hierarchy of substructures. Once SUBDUE selects a substructure, all nodes which comprise the exact instances of the substructure are replaced in the graph by a single node representing the discovered substructure. Edges connecting nodes outside the instance to nodes inside the instance now connect to the new node. Edges internal to the instance are removed. The program is then run a second time, with heavier weight given to substructures which utilize the previously discovered substructure. The increased weight reflects increased attention to this substructure. Figure 2 shows the results after each pass. Note that on the third pass SUBDUE linked together the instances of the substructure in the second pass to find the chains of the double helix. Results indicate that SUBDUE can discover pertinent substructures and find a hierarchical description of the input data by replacing previously-discovered substructures on successive passes.

VI. CONCLUSIONS

Automated knowledge discovery is essential for extracting information from databases [2]. Extracting knowledge from structural databases requires the identification of repetitive substructures in the data. The previous examples show how SUBDUE's heuristic search and inexact graph match can discover interesting and repetitive substructures in real structural domains. Applying SUBDUE to scene analysis assists in compression of the image and identification of similar objects in the scene. Application to chemical analysis assists the discovery of previously-unknown molecules and cognitive compression of the compound by abstracting over newly-discovered molecules. Further experimentation is underway in both artificial domains and other real domains in order to determine the effects of parameters and reduce the computational requirements of SUBDUE's substructure discovery algorithm.

References

[1] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.

[2] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus. Knowledge discovery in databases: An overview. In G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, chapter 1, pages 1–27. MIT Press, 1991.

[3] L. B. Holder. Empirical substructure discovery. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 133–136, 1989.

[4] L. B. Holder, D. J. Cook, and H. Bunke. Fuzzy substructure discovery. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 218–223, 1992.

[5] D. Waltz. Understanding line drawings of scenes with shadows. In P. H. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, 1975.