Coupling Two Complementary Knowledge Discovery Systems *

Lawrence B. Holder and Diane J. Cook
University of Texas at Arlington
Box 19015, Arlington, TX 76019-0015
Email: {holder,cook}@cse.uta.edu

Abstract

Most approaches to knowledge discovery concentrate on either an attribute-value representation or a structural data representation. The discovery systems for these two representations are typically different, and their integration is non-trivial. We investigate a simpler integration of the two systems by coupling the two approaches. Our method first executes the structural discovery system on the data, and then uses these results to augment or compress the data before being input to the attribute-value-based system. We demonstrate this strategy using the AutoClass attribute-value-based clustering system and the Subdue structural discovery system. The results of the demonstration show that coupling the two systems allows the discovery of knowledge imperceptible to either system alone.

Introduction

With the increasing amount and complexity of today's data, there is an urgent need to improve the discovery of knowledge in large databases. Numerous approaches have been developed for discovering knowledge in databases using a linear, attribute-value representation. Although much of the data collected today has an explicit or implicit structural component (e.g., spatial or temporal), few discovery systems are designed to handle this type of data (Fayyad, Piatetsky-Shapiro, & Smyth 1996). One reported method for dealing specifically with structural data is with the SUBDUE system (Cook, Holder, & Djoko 1996). SUBDUE provides a method for discovering substructures in structural databases using the minimum description length (MDL) principle introduced by Rissanen.

Although numerous approaches exist for either attribute-value-based or structural discovery, no system provides a general approach to performing discovery in the presence of both types of data. An alternative to a unified approach is to separate the attribute-value and structural components of the database, run appropriate discovery tools on each component, and combine the results. We demonstrate this approach using two specific systems: the AutoClass clustering system and the Subdue structural discovery system. The results of the demonstration show that coupling the two systems allows the discovery of knowledge imperceptible to either system alone.

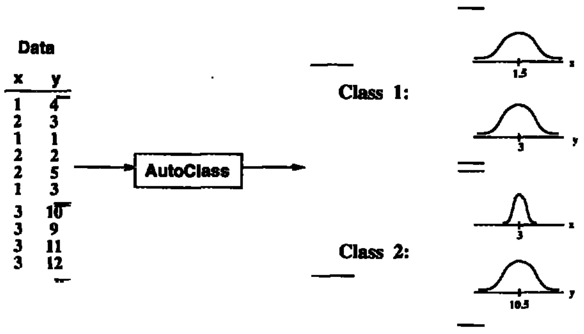


Figure 1: Sample input/output for AUTOCLASS.

AutoClass

The AUTOCLASS system (Cheeseman & Stutz 1996) is an attribute-value-based discovery tool that clusters the input data into classes describing natural partitions in the data. Input to AUTOCLASS consists of a set of tuples described by real-valued and discrete-valued attributes, a probabilistic model for each attribute, and a set of control parameters. The output of AUTOCLASS is a set of classes. The best classification is the one that maximizes the ability to predict the attribute values of a tuple in the data, given the correct class of the tuple.

Figure 1 shows a simple example of the typical input and output for AUTOCLASS. The input consists of two real-valued attributes X and Y whose probabilistic models are given as Gaussian. AUTOCLASS finds two classes in the data, each described by an instantiated Gaussian model for each attribute.

To perform the clustering, AUTOCLASS uses a

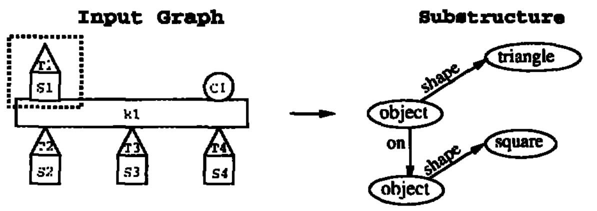


Figure 2: Example substructure in graph form.

Bayesian statistical technique. AUTOCLASS breaks the clustering problem into two parts: determining the number of classes or clusters and determining the parameters of the attribute probabilistic models that define the classes.

AUTOCLASS has classified data supplied by researchers from a number of domains. These domains include a DNA protein donor/acceptor site database, a 1024x1024 LandSat image of a county in Kansas, and Infrared Astronomical Satellite (IRAS) data. From the IRAS database, AUTOCLASS discovered 5,425 classes using 94 different attributes, and gave NASA scientists a new look at a database which is not thoroughly understood by domain experts. AUTOCLASS discovered classes which differed significantly from NASA's previous analysis but clearly reflect physical phenomena in the data.

Subdue

The SUBDUE system (Cook & Holder 1994; Cook, Holder, & Djoko 1996; Djoko, Cook, & Holder 1995) is a structural discovery tool that finds substructures in a graph-based representation of structural databases using the minimum description length (MDL) principle introduced by (Rissanen 1989). SUBDUE discovers substructures that compress the original data and represent structural concepts in the data. Once a substructure is discovered, the substructure is used to simplify the data by replacing instances of the substructure with a pointer to the newly discovered substructure. The discovered substructures allow abstraction over detailed structures in the original 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 specific goals of the data analysis.

SUBDUE represents structural data as a labeled graph. Objects in the data map to vertices or small subgraphs in the graph, and relationships between objects map to directed or undirected edges in the graph. A *substructure* is a connected subgraph within the graphical representation. This graphical representation serves as input to the substructure discovery system. Figure 2 shows a geometric example of such an input graph. The objects in the figure become labeled vertices in the graph, and the relationships become labeled edges in the graph. The graphical representation of the substructure discovered by SUBDUE from this data is also shown in Figure 2. One of the four instances of the substructure is highlighted in the input graph. An *instance* of a substructure in an input graph is a set of vertices and edges from the input graph that match, graph theoretically, to the graphical representation of the substructure.

Figure 3 shows a sample input database containing a portion of a DNA sequence. In this case, atoms and small molecules in the sequence are represented with labeled vertices in the graph, and the single and double bonds between atoms are represented with labeled edges in the graph. SUBDUE discovers substructure S_1 from the input database. After compressing the original database using S_1, SUBDUE finds substructure S_2, which when used to compress the database further allows SUBDUE to find substructure S_3. Such repeated application of SUBDUE generates a hierarchical description of the structures in the database.

The substructure discovery algorithm used by SUBDUE is a computationally-constrained beam search. The algorithm begins with the substructure matching a single vertex in the graph. Each iteration the algorithm selects the best substructure and incrementally expands the instances of the substructure. The algorithm searches for the best substructure until all possible substructures have been considered or the total amount of computation exceeds a given limit. Evaluation of each substructure is determined by how well the substructure compresses the description length of the database. Because instances of a substructure can appear in different forms throughout the database, an inexact graph match is used to identify substructure instances.

SUBDUE has been successfully applied with and without domain knowledge to databases in domains including image analysis, CAD circuit analysis, Chinese character databases, program source code, chemical reaction chains, Brookhaven protein databases, and artificially-generated databases.

Coupling Complementary Discovery Systems

One method for coupling a linear attribute-value-based discovery system (system A) and a structural discovery system (system S), depicted in Figure 4, is to run system S first on the structural component of the data and use the resulting structural patterns to modify the input to system A. The results of system S, which will typically link several tuples together, can be used to augment tuples

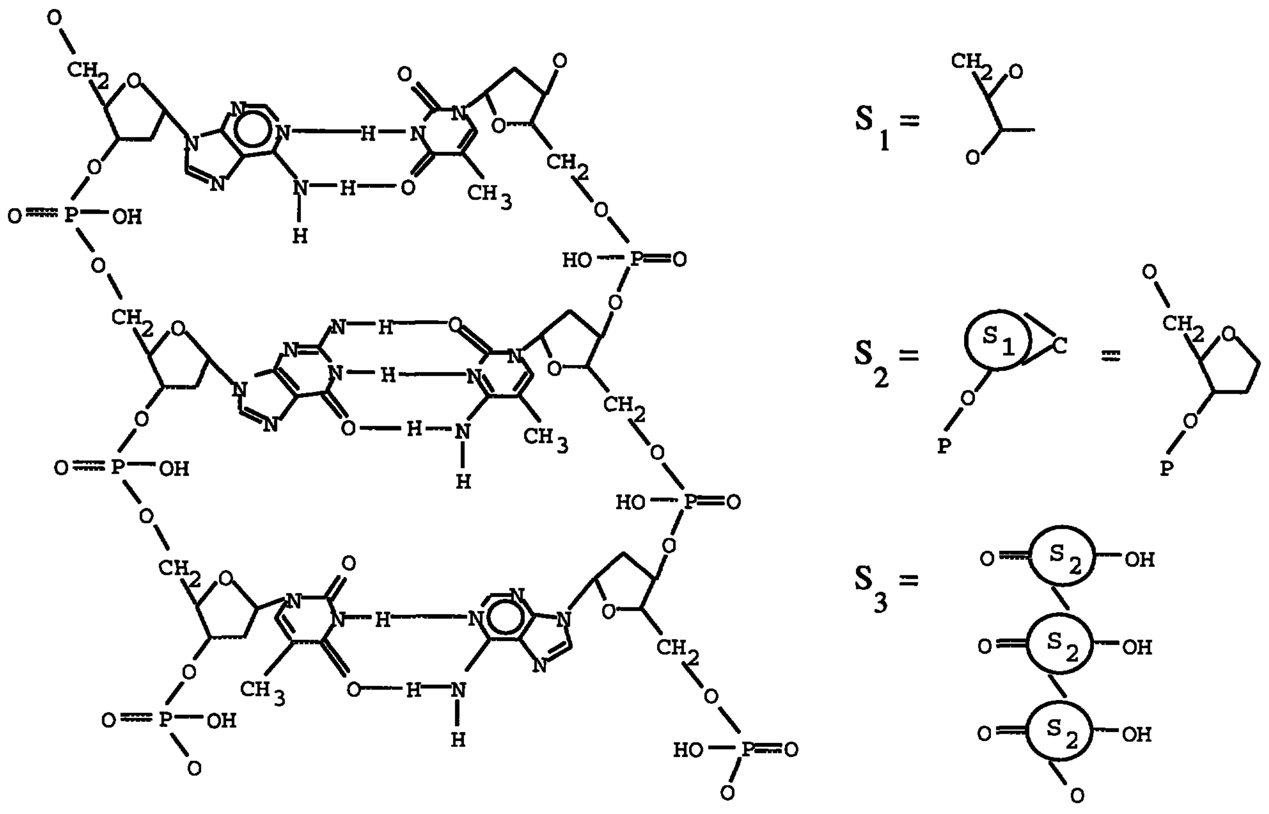
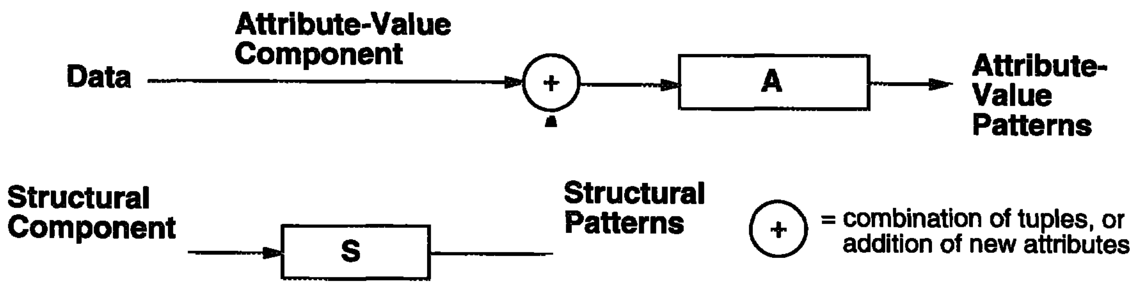


Figure 3: Sample results of Subdue on a protein sequence.



Data
Attribute-Value Component
A
Attribute-Value Patterns
Structural Component
S
Structural Patterns
= combination of tuples, or addition of new attributes

Figure 4: Coupling of complementary discovery systems ($S \rightarrow A$).

with attributes whose values indicate other tuples involved in the structural pattern or can be used to combine all the attributes of the participating tuples into a single, larger tuple.

As a demonstration of this coupling strategy, we perform discovery in the squares database shown in Figure 5. The squares database consists of 30 squares represented as 120 line segments. Each line segment consists of the line starting point (X1,Y1) and ending point (X2,Y2) (0-1023) along with the line's angle (0.0-10.0), length (1-99) and color (red, green or blue). In Figure 5, red lines are solid, green lines are dotted, and blue lines are dashed. The pattern we have artificially embedded in this database consists of three types squares based on line color: 10 red-red-green-green squares, 10 green-green-blue-blue squares, and 10 red-red-blue-blue squares. To demonstrate the benefits of the coupling strategy, we compare the results of this strategy with those of AutoClass alone and Subdue alone.

AutoClass represents this database using one tuple for each line. The tuple consists of attributes for X1, Y1, X2, Y2, Angle, Length and Color. Each attribute is modeled by a Gaussian. This includes the Color attribute using the values 1 for red, 2 for green and 3 for blue. We run AutoClass with this information and with added structural information in the form of adding a line number (LineNo) to each entry and two attributes (LineTo1 and LineTo2) whose values are the line numbers of the lines connected to the enpoints, (X1,Y1) and (X2,Y2), respectively. Given these 120 tuples, AutoClass alone finds the following 12 classes (numbers in parentheses are the number of tuples having highest probability of belonging to the class):

Class 0 (20): Color = green,
 LineNo = LineTo1 = LineTo2 = 98 ± 10
Class 1 (20): Color = red,
 LineNo = LineTo1 = LineTo2 = 99 ± 10
Class 2 (15): Color = red,
 LineNo = LineTo1 = LineTo2 = 12 ± 10
Class 3 (13): Color = blue,
 LineNo = LineTo1 = LineTo2 = 17 ± 10
Class 4 (12): Y1 = Y2, X1 = X2
Class 5 (10): Length = 80 ± 1,
 LineNo = LineTo1 = LineTo2 = 51 ± 1
Class 6 (8): Y1 = Y2,
 LineNo = LineTo1 = LineTo2 = 69 ± 1
Class 7 (8): X1 = X2,
 Length = 8.4 ± 1.2
Class 8 (4): Length = 59 ± 1,
 LineNo = LineTo1 = LineTo2 = 54 ± 1
Class 9 (4): Length = 90 ± 1,
 LineNo = LineTo1 = LineTo2 = 33 ± 1
Class 10 (3): Color = blue, LineNo = 3 ± 35
Class 11 (3): LineTo2 = 1 ± 13,
 Color = green

These classes show that AutoClass uses color as the main differentiator between classes. The concept of squares, and particularly two-colored squares, is absent from these concept descriptions.

Each line is represented as a vertex in the graph input to Subdue, and the attribute values are represented as directed edges from the line vertex to a vertex for that attribute's value. The one addition piece of information given to Subdue is the structural relation "touch" used to connect adjoining lines. The output of Subdue on the 30 squares corresponds to the four connected lines comprising the square with no attribute information. Thus, Subdue discovers the square, but not the fact that there are three different types of squares based on line color.

Completing the coupling strategy of Figure 4, we next use the structural pattern found by Subdue to modify the original attribute-value data for input to AutoClass. The modification involves combining the four line tuples comprising each square structure into a single larger tuple consisting of all the attributes of the four line tuples. Thus, the database is reduced to 30 tuples, one for each square, where each tuple has attributes corresponding to the four corners of the square (X1-X4, Y1-Y4) along with the angles, lengths and colors of the four lines. Given the modified database, AutoClass finds the desired classes that neither AutoClass or Subdue could find alone:

Class 0 (10): Color1 = red, Color2 = red,
 Color3 = green, Color4 = green
Class 1 (10): Color1 = green, Color2 = green,
 Color3 = blue, Color4 = blue
Class 2 (10): Color1 = blue, Color2 = blue,
 Color3 = red, Color4 = red

To further test this coupling strategy, we created a second artificial database consisting of three embedded concepts: two small structures and one larger structure. Each vertex of the structures is embellished with a numeric attribute. The numeric values vary between instances of the graphs but fall within specified ranges for each vertex of each structure. In the database we embed 9 instances of the smaller concepts and 5 instances of the larger concept.

AutoClass is given the dataset represented as a set of labeled data entries with numeric attributes and structural information indicating to what other data points each entry is connected in the structure. When AutoClass is run alone on the dataset, 12 classes are generated which partition the data mainly on the numeric values. Subdue run alone on this dataset finds the three structural concepts without any numeric information. We use Subdue's findings to compress the database and run AutoClass once more, at which point AutoClass finds the three classes containing the structural con-

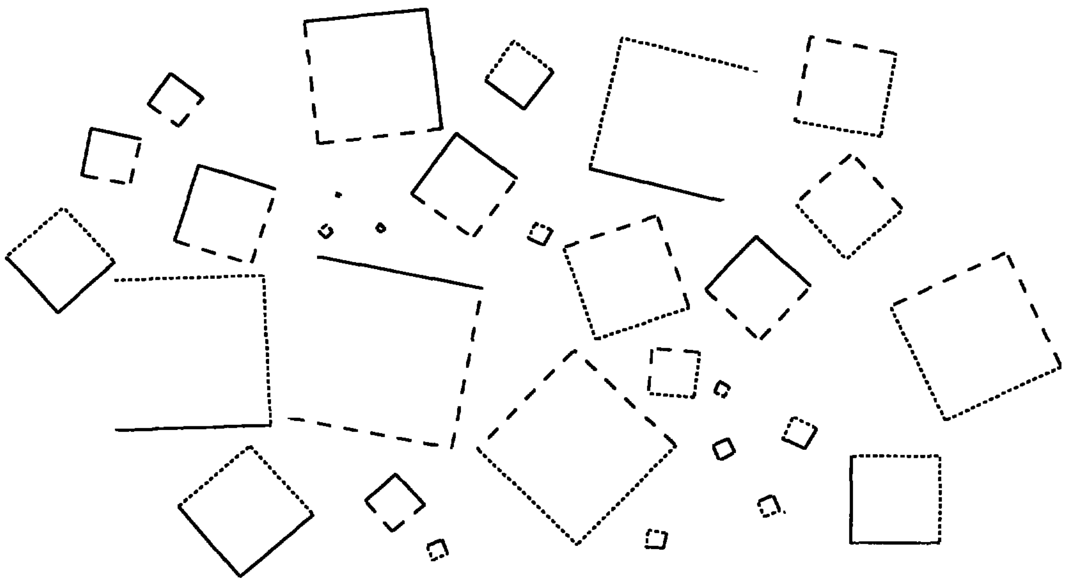


Figure 5: Squares database.

cepts with attached numeric ranges.

Conclusions

As this paper demonstrates, coupling two complementary knowledge discovery systems can yield patterns imperceptible to either system alone. Coupling the systems provides an easier alternative to integrating the biases of an attribute-value-based system into a structural discovery system or enhancing the attribute-value-based system's representation to include structural information.

We have only demonstrated the usefulness of a coupling strategy on two artificial databases. Much more evaluation remains to be done. One direction would be to evaluate the system on satellite images of cities containing buildings of various sizes and orientations. After extracting line segment information from the image, executing the coupled systems may be able to find relevant classes of buildings, instead of just different classes of line segments.

Acknowledgements

This work was supported by National Science Foundation grants IRI-9502260, DMI-9413923, and IRI-9615272.

References

Cheeseman, P., and Stutz, J. 1996. Bayesian classification (AutoClass): Theory and results. In Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining.* MIT Press. chapter 6, 153–180.

Cook, D. J., and Holder, L. B. 1994. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research* 1:231–255.

Cook, D. J.; Holder, L. B.; and Djoko, S. 1996. Scalable discovery of informative structural concepts using domain knowledge. *IEEE Expert* 10:59–68.

Djoko, S.; Cook, D. J.; and Holder, L. B. 1995. Analyzing the benefits of domain knowledge in substructure discovery. In *The First International Conference on Knowledge Discovery and Data Mining.*

Fayyad, U. M.; Piatetsky-Shapiro, G.; and Smyth, P. 1996. From data mining to knowledge discovery: An overview. In Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining.* MIT Press. chapter 1, 1–34.

Rissanen, J. 1989. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Company.