Graph-Based Hierarchical Conceptual Clustering

Istvan Jonyer, Lawrence B. Holder and Diane J. Cook

Department of Computer Science and Engineering
University of Texas at Arlington
Box 19015 (416 Yates St.), Arlington, TX 76019-0015
E-mail: {jonyer | holder | cook}@cse.uta.edu
Phone: (817) 272-2596
Fax: (817) 272-3784

Abstract

Hierarchical conceptual clustering has been proven to be a useful data mining technique. Graph-based representation of structural information has been shown to be successful in knowledge discovery. The Subdue substructure discovery system provides the advantages of both approaches. In this paper we present Subdue and focus on its clustering capabilities. We use two examples to illustrate the validity of the approach both in structured and unstructured domains, as well as compare Subdue to an earlier clustering algorithm.

Introduction

Cluster analysis has been studied and developed in many areas for a wide variety of applications. Among these are model fitting, hypothesis testing, hypothesis generation, data exploration, prediction based on groups, data reduction and finding true topologies [Ball 1971]. Clustering techniques have been applied in as diverse fields as analytical chemistry, geology, biology, zoology and archeology, just to mention a few. Many names have been given to this technique, among which are cluster analysis, Q-analysis, typology, grouping, clumping, classification, numerical taxonomy, mode separation and unsupervised pattern recognition, which further signifies the importance of clustering techniques [Everitt 1980].

The purpose of applying clustering to a database is to gain better understanding of the data, in many cases through revealing hierarchical topologies. An example of this is the classification of vehicles into groups such as cars, trucks, motorcycles, tricycles, and so on, which are then further subdivided into smaller and smaller groups based on some other traits.

In this paper we present Subdue, a structural knowledge discovery system, specifically focusing on its clustering capabilities. After acknowledging some earlier works, we describe Subdue, and present examples to highlight our results.

Related Work

Numerous clustering techniques have been devised in the past, among which are statistical, syntactic, neural and hierarchical approaches. In all cases, clustering is inherently an unsupervised learning paradigm, since it consists of identifying valuable groupings of concepts, or facts, which hopefully reveal previously unknown information. Most techniques have some intrinsic disadvantages, however. Statistical and syntactic approaches have trouble expressing structural information, and neural approaches are greatly limited in representing semantic information [Schalkoff 1992].

Nevertheless, many relatively successful clustering systems have been constructed. An example of an incremental approach is Cobweb, which successively considers a set of object descriptions, while constructing a classification tree [Fisher 1987]. Labyrinth [Thompson and Langley 1991], an extension to Cobweb, can represent structured objects using a probabilistic model. AutoClass [Cheeseman at al. 1988] is an example of a bayesian classification system, which has a probabilistic class assignment scheme. It can deal with real, discrete, or missing values. Yet another algorithm, called Snob, uses the Minimum Message Length (MML) principle to do mixture modeling—another synonym for clustering [Wallace 1968].

There also exist hierarchical approaches that work on databases containing data in Euclidian space. Among these are agglomerative approaches that merge clusters until an optimal separation of clusters is achieved based on intra-, and inter-cluster distances. Divisive approaches split existing clusters until an optimal clustering is found. These approaches usually have the disadvantage of being applicable only to metric data, which excludes discrete-valued and structured databases. Examples of these are Chameleon [Karypis, Han and Kumar 1999] and Cure [Guha, Rastogi and Shim 1998].

Conceptual Clustering Using Subdue

Subdue [Holder and Cook 1993] is a knowledge discovery system that can deal with structured data—a very important feature in more and more applications. Subdue expects a graph as its input, hence a database needs to be represented as a graph before passing it to Subdue. This graph representation includes vertex and edge labels, as well as directed and undirected edges, where objects and data usually map to vertices, and relationships and attributes map to edges (see Figure 2 for an example).

Subdue's discovery algorithm discovers interesting, repetitive substructures in the input graph. Our Graph-Based Hierarchical Conceptual Clustering (GBHCC) algorithm begins with an empty lattice and calls Subdue to find a substructure S that maximally compresses the input graph G. If S achieves some compression of G, then S is added to the lattice and used to compress the graph G. The compressed graph is passed again to Subdue to find another substructure. This iterative approach on successively more compressed graphs allows Subdue to find new substructures defined in terms of previously discovered substructures. Therefore, when substructures are added to the lattice, their parents may include other, non-root nodes in the lattice. If a substructure is composed of two of the same previously-discovered substructures, then there will be two links from the parent to the child in the lattice.

Subdue's discovery algorithm discovers substructures in the input graph. Subdue uses a beam search that–starting with single-node subgraphs–incrementally expands the substructures that seem the best thus far. During the expansion process, a substructure is expanded in all possible ways by its neighboring vertices, and all instances of these new substructures are found. This discovery process continues iteratively until all possible subgraphs have been considered, or the algorithm reaches a user-specified limit. After the best substructure is found and the graph is compressed, the process starts all over, finding another best substructure. This search is guided by the Minimum Description Length (MDL) principle, originally developed by Rissanen [Rissanen 1989]. According to the evaluation heuristic, the best substructure is the one that minimizes the description length of the graph when compressed by the substructure. When compressing the graph, all instances of the substructure are replaced by a single vertex, which is a pointer to the substructure's definition.

This approach imposes more and more hierarchy on the database with each successive iteration. The definition of the best substructure after a single iteration yields the description of a cluster. After identifying a cluster, it is inserted into the *classification lattice* (see Figure 5). Previous works on clustering suggested the use of classification trees, however, in structured domains the strict tree representation is inadequate. We realized that in certain domains a lattice-like structure emerges instead of a tree.

Subdue searches the hypothesis space of all classification lattices. During each iteration of the search process, numerous local minima are encountered, where the global minimum tends to be one of the first few minima. For clustering purposes the first local minimum is used as the best partial hypothesis. The reason for this is easy to see. Subdue starts with all the single-vertex instances of all unique substructures, and iteratively expands the best ones by a single vertex. The local minimum encountered first is therefore caused by a smaller substructure with more instances than the next local minimum, which must be larger, and have fewer instances. A smaller substructure is more general than a larger one, and should be a parent node in the classification lattice for any more specific clusters. Even though it is entirely possible to use the global minimum as the best substructure, we found that if the global minimum is not the first local minimum it may produce overlapping clusters. Overlapping clusters are those that include the same information. For example, in a particular clustering of the vehicles domain two clusters may include the information *"number of wheels: 4"*. This suggests that perhaps a better clustering may be constructed in which this information is part of a cluster at a higher level.

Subdue supports biasing the discovery process. Predefined substructures can be provided to Subdue, which will try to find and expand these substructures, this way "jump-starting" the discovery. The inclusion of background knowledge proved to be of great benefits [Djoko, Cook and Holder 1997]. Inexact graph matching is also provided by Subdue to account for slight variations of a substructure. The user is given control of the degree of similarity between the substructures to be considered the same. Subdue also supports supervised learning, where positive and negative examples are provided to the system. Substructures found that are similar to positive examples are given a higher value, while substructures similar to the negative examples are penalized. This way of influencing the discovery process has proven successful, an example of which is the application of Subdue to the chemical toxicity domain [Chittimoori, Holder and Cook 1999].

Experiments

A small experiment devised by Fisher can serve as an example of Subdue's performance on unstructured data, as well as offer a brief comparison to Cobweb. The database used for the experiment is given in Table 1. Cobweb produces the classification tree shown in Figure 1, as suggested by Fisher [Fisher 1987].

The animal domain is represented in Subdue as a graph, where attribute names (like *Name* and *BodyCover*) were mapped to edges, and attribute values (like *mammal* and *hair*) were mapped to vertices. In unstructured databases

Table 1 Animal Descriptions

Name	*Body Cover*	*Heart Chamber*	*Body Temp.*	*Fertilization*
mammal	hair	four	regulated	internal
bird	feathers	four	regulated	internal
reptile	cornified-skin	imperfect-four	unregulated	internal
amphibian	moist-skin	three	unregulated	external
fish	scales	two	unregulated	external

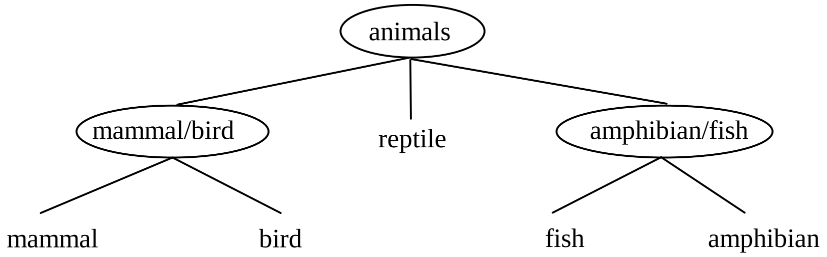

Figure 1 Hierarchical clustering over animal descriptions by Cobweb.

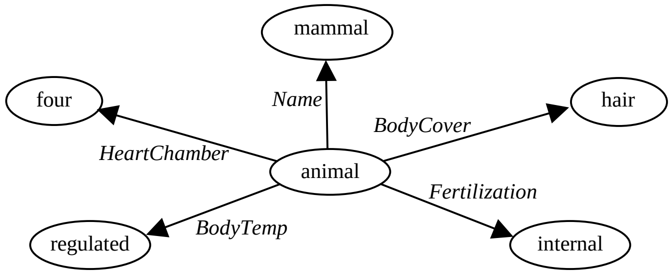

Figure 2 Graph representation of an animal description

like this the data set translates to a collection of small, star-like, connected graphs. Intuitively, we would map the "main" attribute–*Name* in this case–to the center node and all other attributes would be connected to this central vertex with a single edge. We found, however, that a more general representation yields better results. In this representation the center node becomes a very general description of the example. In this case the center node can be *animal*. Note that the *Name* attribute becomes just a regular attribute (see Figure 2). In the most general case, the center node could be named *entity*, or *object*, since the designation is quite irrelevant to the discovery process—the purpose is proper structural representation.

Subdue generated the hierarchical clustering shown in Figure 3. Subdue actually produces a hierarchical graph as its output that can be viewed with a visualization software. This also allows us to include informative details in the classification lattice. A node in the classification hierarchy includes the description of vertices and edges that form the cluster.

Subdue's result is similar to that of Cobweb's. The "*mammal/bird*" branch is clearly the same. Amphibians and fish are grouped in the same cluster based on their external fertilization, which is done the same way by Cobweb. Subdue, however, incorporates reptiles with amphibians and fish, based on their commonality in unregulated body temperature. This clustering of the animal domain seems better, since Subdue eliminated the overlap between the two clusters (*reptile* and *amphibian/fish*) by creating a common parent for them that describes this common trait.

To illustrate Subdue's main advantage–the ability to work with structured data–we present a task that involves describing a DNA sequence by clustering. A portion of the

DNA is shown in Figure 4. To represent the DNA as a graph, atoms and small molecules are mapped to vertices, and bonds are represented by undirected edges. The edges are labeled according to the type of bond, single or double. A portion of the classification lattice is shown in Figure 5. For better understanding, we show the chemical compounds the clusters define, rather than the textual description extracted from the graph representation of the DNA (like in Figure 3).

The lattice closely resembles a tree, with the exception that two nodes (bottom-left) have two parents. The lattice in Figure 5 describes 71% of the DNA sequence shown in Figure 4. As the figure shows, smaller, more commonly occurring compounds are found first that compose the first level of the lattice. These account for more than 61% of the DNA. Subsequently identified clusters are based on

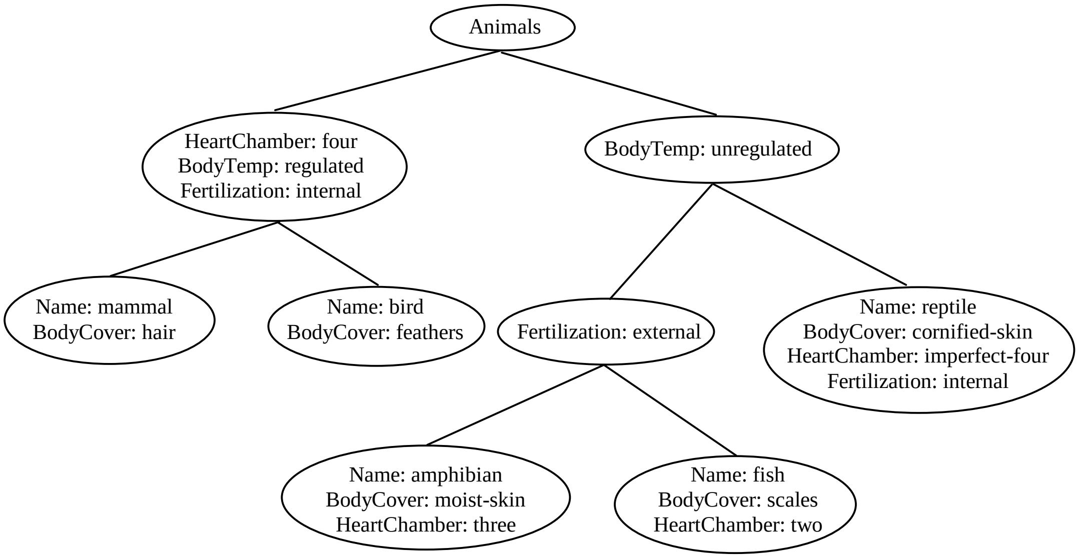

Figure 3 Hierarchical clustering over animal descriptions by Subdue.

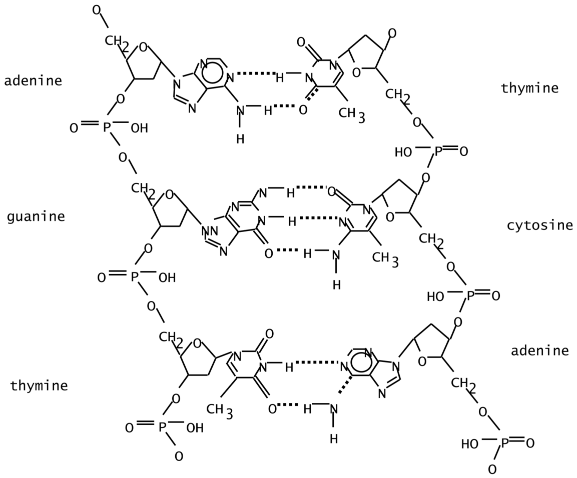

Figure 4 Portion of a DNA molecule

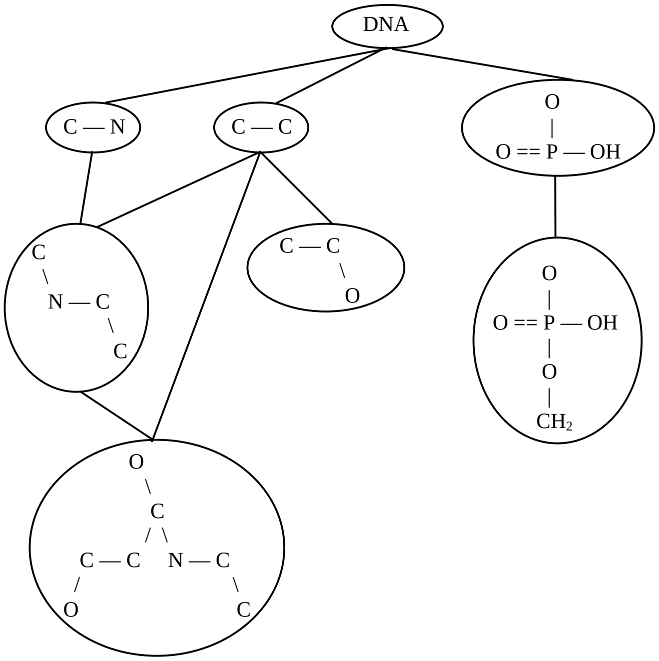

Figure 5 Partial hierarchical clustering of a DNA sequence.

these smaller clusters that are either combined with each other, or with other atoms or molecules to form a new cluster. The second level of the lattice extends the conceptual clustering description such that an additional 7% of the DNA is covered. Evaluation of the domain-relevance of Subdue's findings will require the assistance of a domain expert.

The best clustering is usually the one that has the minimum number of clusters, with minimum number of overlaps between clusters, such that the entire data set is described. Too many clusters can arise if the clustering algorithm fails to generalize enough in the upper levels of the hierarchy, in which case the classification lattice may become shallow with a high branching factor from the root, and a large number of overlaps. On the other extreme, if the algorithm fails to account for the most specific cases, the classification lattice may not describe the data entirely. Experimental results indicate that Subdue finds clusterings that effectively trade off these extremes.

Conclusions

Most previous efforts at clustering work with unstructured databases that simply enlist object descriptions. Subdue overcomes this restriction by representing databases using graphs, which allows for the representation of a large number of relationships between items, which is an integral part of defining clusterings.

We have demonstrated that Subdue's performance on unstructured datasets parallels one of the most prominent algorithms so far, perhaps even outperforming it. We also showed Subdue's applicability to chemical domains—specifically, a conceptual clustering of a DNA sequence.

Future work on Subdue will include defining hierarchical clusterings of real-world domains, and comparisons to other clustering systems. Since data mining researchers have to rely on experts' opinion to evaluate the effectiveness of their clustering algorithm, it would be extremely useful to devise metrics for objective evaluation of clusterings.

References

[Ball 1971] Ball, G. H. *Classification Analysis*. Stanford Research Institute SRI Project 5533, 1971.

[Cheeseman at al. 1988] Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., & Freeman, D. *Autoclass: A bayesian classification system*. In Proceedings of the Fifth International Workshop on Machine Learning, 1988, pp. 54--64.

[Chittimoori, Holder and Cook 1999] R. Chittimoori, L.B. Holder, and D.J. Cook. Applying the Subdue Substructure Discovery System to the Chemical Toxicity Domain. In the *Proceedings of the Twelfth International Florida AI Research Society Conference*, 1999, pages 90-94.

[Djoko, Cook and Holder 1997] S. Djoko, D. J. Cook, and L. B. Holder. *An Empirical Study of Domain Knowledge and Its Benefits to Substructure Discovery*. In the IEEE Transactions on Knowledge and Data Engineering, Volume 9, Number 4, 1997.

[Everitt, 1980] Everitt, B. S. *Cluster Analysis*. Wiley & Sons, New York, 1980.

[Fisher 1987] Fisher, D. H. *Knowledge Acquisition Via Incremental Conceptual Clustering, Machine Learning.* Kluwer, The Netherlands, 1987.

[Guha, Rastogi and Shim 1998] S. Guha, R. Rastogi, K. Shim. *CURE: An Efficient Clustering Algorithm for Large Databases*. ACM SIGMOD International Conference on Management of Data, 1998.

[Holder and Cook 1993] L. B. Holder and D. J. Cook. *Discovery of Inexact Concepts from Structural Data*. In IEEE Transactions on Knowledge and Data Engineering, Volume 5, Number 6, 1993, pages 992-994.

[Karypis, Han and Kumar 1999] G. Karipis, E. Han, V. Kumar. *Chameleon: Hierarchical Clustering Using Dynamic Modeling.* In Computer, August, 1999, pages 68-75.

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

[Schalkoff, 1992] R. Schalkoff. *Pattern Recognition*. Wiley & Sons, New York, 1992.

[Thompson and Langley 1991] Concept formation in structured domains. In Fisher, D.H., & Pazzani, M. (Eds.), *Concept Formation: Knowledge and Experience in Unsupervised Learning,* chap. 5. Morgan Kaufmann Publishers, Inc., 1991.

[Wallace 1968] Wallace, C.S. and Boulton, D.M., *An Information Measure for Classification*, Computer Journal, Vol.11, No.2, 1968, pp 185-194.