From law-like knowledge to concept hierarchies in data

Molly Troxel, Kim Swarm, Robert Zembowicz, and Jan M. Żytkow

Computer Science Department
Wichita State University
Wichita, KS 67260-0083
{mltroxel, ksswarm, robert, zytkow}@wise.cs.twsu.edu

Keywords: discovery, regularities, concept formation, concept hierarchies, predictive power, empirical contents.

Abstract

In this paper we analyze relationships between different forms of knowledge that can be discovered in the same data matrix (database): regularities, concept descriptions and conceptual clusters (hierarchies). These relationships, very important for our understanding of knowledge, have not received sufficient attention, neither in the domain of machine learning nor from the perspective of knowledge based systems. We argue for the basic role of regularities (law-like knowledge) and we show how a subset of the discovered regularities, made of regularities which approximate logical equivalences, can be used to construct concept hierarchies. We show how each of those regularities leads to an element of the conceptual hierarchy and how those elements are linked to form elements of higher empirical contents. One-way implications can also contribute to the empirical contents of hierarchy elements. Next we show how to combine hierarchy elements into concept hierarchy. Different hierarchies are possible, leading to the question of choice between hierarchies, for which we provide our optimality criteria. The algorithm is illustrated by a walk-through application on the soybean database. We compare our results with results obtained earlier by the COBWEB clustering approach.

1 Introduction: concepts and regularities discovered in data

Relational tables in databases (data matrices in statistics, collections of examples in the machine learning research) have been used for a long time to seek different kinds of knowledge, for instance, concepts, taxonomies, and regularities. The relationships between these forms of knowledge, however, did not receive sufficient attention.

We argue that concepts and taxonomies are a limited form of knowledge, compared to regularities, which are also called law-like knowledge. We demonstrate that useful boolean concepts and their hierarchies can be typically inferred from especially simple types of regularities discovered in data. Other forms of knowledge are both important and non-reducible to concepts and taxonomies.

1.1 Concept learning and concept discovery

In technical terms of logic, concepts are predicates which include free variables. They name objects, properties or patterns, but they are not statements, as they are neither true nor false. Truth values

can be assigned to statements, which use concepts and which have all variables bound by quantifiers, either in explicit or implicit way. Statements are claims about the world. With the exception of tautologies, true and universally quantified statements are typically called laws or regularities.

A proven model of concept discovery comes from science. Concept discovery in science is not an isolated activity, because concepts are justified by feedback from knowledge. Concepts can be viewed as investments which produce payoff when they allow us to express regularities and laws. Better investments, that is, better concepts can be recognized by analyzing regularities that they permit to express. Among an unlimited number of concepts that can be proposed, science uses a very limited number, choosing them based on the generality, predictive power, accuracy, and number of laws in which they occur. In machine discovery we also use the same feedback (Langley, Simon, Bradshaw, & Zytkow, 1987; Nordhausen & Langley, 1993; Shen, 1993). In our paper we use predictive strength, the scope of applications, and the number of laws to guide concept formation.

Concept learning from examples can be viewed as a very limited search for regularities. Membership in the target class is described by the target attribute, which indicates for each record in a relational table, whether it belongs to that class or not, that is, whether it is an example or a counterexample. The learner seeks the best definition of the target class in terms of other attributes. Such a definition has a truth value. If true, it shares many features of regularities, for instance, it can be used to predict class membership. The target class is externally defined and a learner searches only for a class definition. In contrast, a discoverer must explore various target attributes, search for regularities for each, and evaluate the concepts based on the number of discovered regularities, their predictive strength, and the range of data they cover. While a learner may not understand the reasons why a concept has value, a discoverer would, because the focus on regularities gives it a good foundation for the autonomous acceptance of concepts.

1.2 Clustering as limited discovery

Clustering is a step towards autonomy in concept learning. Here the task is more open, aimed at the autonomous creation of classes. Given a data matrix, clustering seeks to divide all records into classes and to find a description of each class. The concern for regularities in data has been notably absent in early clustering systems, and resultant taxonomies have had little scientific value. A new generation of clustering systems guides the clustering process by predictivity of clusters (Fisher, 1987). The resultant cluster hierarchies demonstrate predictive power when regularities are present in the data. In addition to knowledge that is contained in concept definitions, additional knowledge is implicit in cluster hierarchies when they are exhaustive and disjoint.

Knowledge included in a taxonomy falls into the category of monadic logic; membership criterion for each class is represented by a unary predicate, while empirical contents of each class and relations between classes are represented by equivalences and implications between such predicates. Knowledge represented by monadic predicates is, of course, very limited.

Regularities for two-dimensional and many-dimensional relations, however, are poorly represented by clusters. A regularity does not separate existing objects into classes, but instead, it specifies a pattern obeyed by all objects. Many classes may be needed to represent predictivity of a simple pattern. For instance, a simple proportionality between attributes x and y must be approximated by many clusters, rather than by a simple regularity $y = ax$. This applies also to relationships between non-numerical attributes, when these attributes have many values. In contradistinction to clustering, the main goal of many discovery systems is to find regularities in the data, while new concept construction has merely instrumental role in the search for regularities. So even if clustering is a limited form of discovery, the global regularities as such are overlooked, while pieces of regularities are captured locally, in different combinations, by clusters.

2 From regularities to useful concepts

Elsewhere (Żytkow & Zembowicz, 1993) we argue that equations and contingency tables are two basic forms of regularities. The generic form of regularity is "Pattern P holds in domain D". A pattern that holds in a domain distinguishes records that are possible in that domain from records that are impossible. Contingency tables, which express statistical regularities, distinguish statistically probable combinations of records from those improbable. The majority of patterns do not imply new concepts. Take an equation in the form $y = f(x)$. It does not "naturally" lead to conceptual distinctions among the values of x or y, because all values of both variables are treated uniformly by the equation. The majority of contingency tables do not lead to "natural" concepts, either, but we will demonstrate that there is a subcategory of contingency tables which gives strong reasons for concept formation. Furthermore, when a number of contingency tables in that category is inferred from data, and if these tables share some common attributes, we can use these regularities to form a concept hierarchy. Such a concept hierarchy captures relationships discovered between different tables.

2.1 Concepts inferred from contingency tables

Consider two types of tables, depicted in Figure 1, labelled with values of two-valued attributes, A1 and A2. Non-zero numbers of occurences of particular value combinations are indicated by n1, n2, and n3. Zeros in the cells indicate that the corresponding combinations of values do not occur in data from which the table has been generated. The upper table in Figure 1, for instance, shows that 0 objects are both A1 and non-A2 (labelled -A2 in the table), while n2 objects are neither A1 nor A2. From the zero values we can infer inductively, with significance that increases as we consider more data, that these value combinations do not occur in the population represented by data.

A1	0	n1
¬A1	n2	0
	¬A2	A2

The regularity expressed in this table is equivalence:
For all x, (A1(x) if and only if A2(x))
2 classes can be defined: (1) A1 and A2, (2) non-A1 and non-A2

A1	0	n1
¬A1	n2	n3
	¬A2	A2

The regularity expressed in this table is implication:
For all x, (if A1(x) then A2(x)) or equivalently:
For all x, (if non-A2(x) then non-A1(x))

Figure 1. Contingency tables that lead to conclusions about concepts.

The upper table motivates the partition of all data into two classes: (1) of objects which are both A1 and A2, and (2) of objects which are neither A1 nor A2. Each class has empirical contents. We can test class membership by the values of one attribute, and predict the value of the other attribute.

The lower table in Figure 1 leads to weaker conclusions. Only the values A1 and non-A2 carry predictive contents. For objects, which are A1, it enables the inference that they are also A2. Equivalently, for objects which are non-A2, they are non-A1.

The interpretation of zeros, illustrated in Figure 1 can be generalized so that it applies to each zero that occurs in any table, but for large tables the inferred concepts and their properties may be too many and too weak.

2.2 Approximate inference

In real-world situations, rarely we see regularities without exceptions. Instead of cells with zero counts, we can expect cells with small numbers, that is, numbers small compared to those in other cells. A robust method should be able to handle exceptions.

We could compare the numbers in different cells, to determine whether some are small in comparison to others, but in our approach we use Cramer's V, set at a threshold close to 1.0. The Cramer's V coefficient is a measure based on χ^2, which measures the distance between tables of actual and expected counts. For a given $M_{row} \times M_{col}$ contingency table, Cramer's V is defined as

$$V = \sqrt{\frac{\chi^2}{N \min(M_{row} - 1, M_{col} - 1)}},$$

where N is the number of records.

Cramer's V can be treated as the measure of the predictive power of a regularity. The regularity between x and y has a larger predictive power if for a given value of x the value of y can be predicted more uniquelly. The strongest predictions are possible when for each value of one attribute there is exactly one corresponding value of the other attribute. For ideal correlation, χ^2 is equal to $N \min(M_{row} - 1, M_{col} - 1)$, so Cramer's $V = 1$. On the other extreme, when the actual distribution is equal to expected, then $\chi^2 = 0$ and $V = 0$. Cramer's coefficient V does not depend on the size of the contingency table nor on the number of records. Thus it can be used as a homogenous measure on regularities found in different subsets and for different combinations of attributes. In addition to Cramer's V, we use significance to qualify regularities for further analysis.

3 From regularities to taxonomies

As a walk-through example that illustrates our method for taxonomy formation we selected the small soybean database of 47 records and 35 attributes, because it has been studied extensively, for instance by Michalski and Chilausky (1980), by Stepp (1984) and Fisher (1987).

We used the 49er system (Żytkow & Zembowicz, 1993) to discover two-dimensional regularities in soybean data, for all combinations of attributes and for a large number of subsets of records. Systems such as EXPLORA (Hoschka & Kloesgen, 1991, Kloesgen, 1992) and other systems developed in the field of knowledge discovery in databases, described in collections edited by Piatetsky-Shapiro & Frawley (1991), Piatetsky-Shapiro (1991, 1993) could be also applied.

In our walk-through example, Cramer's V threshold of ≥ 0.90 was used. All regularities in the form of contingency tables, discovered by 49er, have been examined, and those with the V values ≥ 0.90 were retained. For example, in Table 1 a regularity between the two attributes stem-cankers and fruiting-bodies is reported, with the Cramer's V rating of 1. Many such regularities have been found, suggesting strongly that the database is conducive to taxonomy formation.

3.1 Hierarchy elements generated from equivalencies

Each regularity, for which Cramer's V meets the threshold requirement, is used to build an elementary hierarchical unit, which is a simple tree, comprised of 3 classes: the root and two children. The root is labeled with the description of the class of records, in which the regularity holds. Each child is labeled with all the descriptors known to define that class. An example of a descriptor is Stem-Cankers(0,1,2), which means the statement that the values of Stem-Canker are 0, 1, or 2. The two "children" classes are approximately disjoint and they exhaustively cover the range of the regularity. Each class is assigned the corresponding property values of both attributes. In

FRUITING-BODIES

1	0	0	0	10
0	10	18	9	0
	0	1	2	3

STEM-CANKERS

Range: All records (47)
Cramer's V = 1.0
Chi-square = 47.0

Table 1. A regularity found in the small soybean dataset by 49er's search for regularities. The numbers in the cells represent the numbers of records with combinations of values indicated at the margins of the table. Note that the value of FRUITING-BODIES (0 or 1) can be uniquely predicted for each value of STEM-CANKERS.

our example, the contingency table of Table 1 contains the knowledge that Fruiting-Bodies (the vertical coordinate) has the value of 1, if and only if Stem-Cankers (the horizontal coordinate) has the value 3. Knowing all the other values of both attributes, this is equivalent to "the value of Fruiting-Bodies is 0 if and only if the value for Stem-Cankers falls in the range of 0,1,2". The corresponding elementary hierarchy unit is depicted in the left part of Figure 2. Each class in that element contains all the attribute/value combination from the corresponding contingency table.

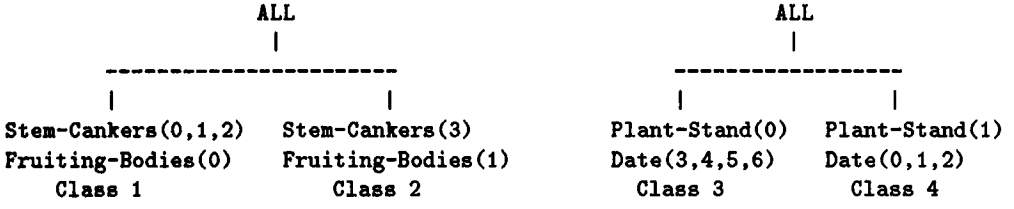

Figure 2. Hierarchical elements built from regularities. Classes 1 and 2 are formed from the regularity in Table 1. Classes 3 and 4 are added from another regularity. Both regularities hold for all data, hence the root is ALL in both cases.

As each further regularity is considered, a new hierarchical element is created. Another example is depicted in the right side of Figure 2. Both regularities in Figure 2 hold over the entire dataset, denoted by the ALL node at each root, but a regularity can hold over a subrange of all records.

3.2 Merging the hierarchy elements

If the same class can be obtained from different regularities, it can be characterized by many descriptors, and can occur under different names in different hierarchy elements. To identify different occurences, after each hierarchy element is created, it is compared to each other element over the same range of records, in search for common descriptors. If they have a common descriptor (the same attribute and equal value sets), the classes are identical (approximately identical, because of exceptions; see above). Both hierarchy elements are collapsed into one and their descriptors are merged (Figure 3).

Two other relations may hold between classes in two hierarchy elements which have an attribute in common: subset and intersection. We will consider the case of subset on example from soybean database provided in Figure 4. For a common attribute Stem-Cankers, the values of Stem-Cankers in Class 2 are a subset of values of Stem-Cankers in Class 3. This means that Class 2 is a subset of Class 3. The corresponding subset link is shown in the upper part of Figure 4 between classes 2 and 3.

Class 4, complementary to Class 3, is a subset of Class 1, complementary to Class 2. To keep

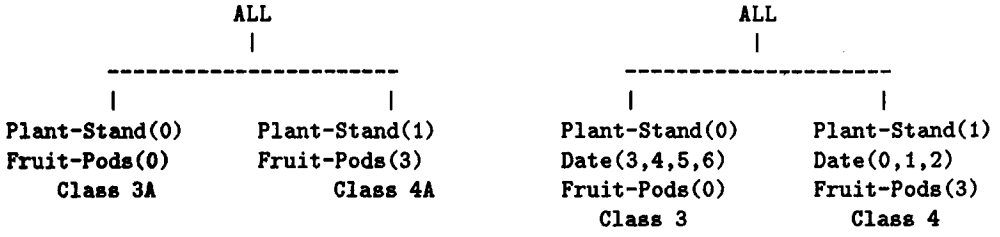

Figure 3. Classes 3 & 4 after the addition of a regularity found between the attributes Plant-Stand and Fruit-Pods, represented by the hierarchy element on the left. As Classes 3 and 4 contained the attribute Plant-Stand and the value ranges for Plant-Stand in Classes 3A & 3B are identical, the corresponding descriptors based on Fruit-Pods is added to the existing Classes 3 & 4.

Figure 4 simple, this subset link is not shown in the top part of Figure 4. The bottom part of Figure 4 describes the situation after the subset links are introduced. If the values of a common attribute in Class 2 and Class 3 would intersect, none of Class 1, Class 2, Class 3, and Class 4 would be a subset of any other, and no construction step would be made.

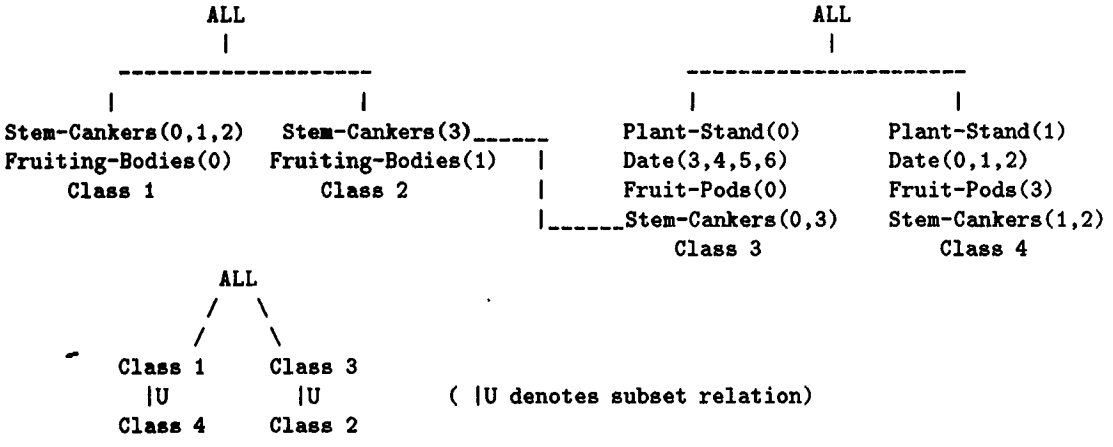

Figure 4. The effect of addition of a regularity between Plant-Stand and Stem-Cankers to Classes 3 & 4. Although Stem-Cankers already described Classes 1 & 2, the value ranges in classes 1,2,3, and 4 are not identical, and no classes could be merged. Special link is formed to indicate the subset relationship between the Classes 2 & 3, detected through the values of the attribute Stem-Cankers. Class 4 is a subset of Class 1, but this link has not been shown in the upper part of the Figure.

New regularities are used to build hierarchy elements, and to link them together, until all regularities found over the full dataset and meeting the Cramer's V threshold are exhausted. The search through the soybean database produced 15 regularities over the entire dataset, that initially led to 30 new classes. After merging, 8 classes remained. In Figure 5 we record the subset relationship between these 8 classes, showing the attribute values of Stem-Cankers and Canker-Lesion. The information about the number of equivalent descriptors for each class indicates the empirical contents of that class.

The same algorithm applies recursively to regularities found in subsets of all data. For a regularity in a subset described by condition C, the root of the hierarchy element is labelled by C (example in Figure 6).

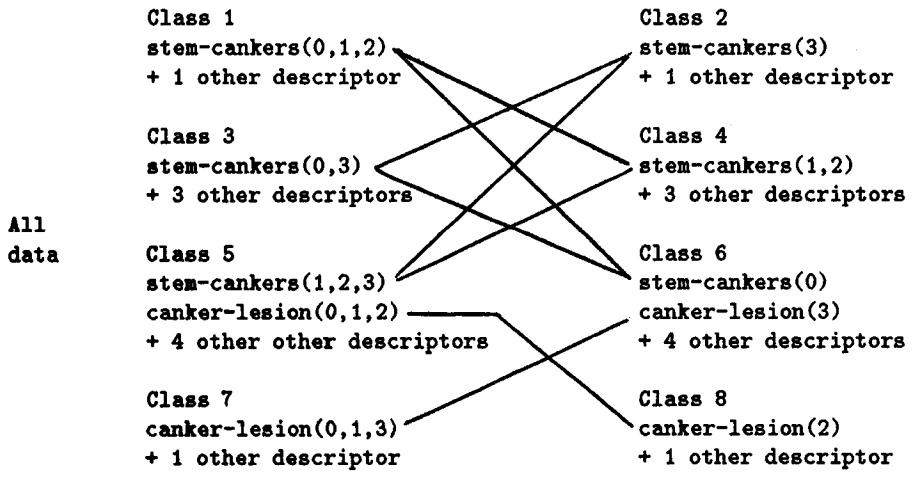

Figure 5. Eight classes generated by strong contingency tables for all data in the soybean database. Links show the subset relation.

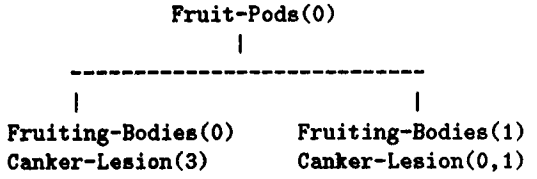

Figure 6. A hierarchy element over a subrange of the soybean data defined by the descriptor Fruit-Pods(0).

3.3 Definitional and inferred properties.

The children classes in each hierarchy element are associated with a number of descriptors. In our soybean example, for instance, after the merging of hierarchy elements has been completed, Classes 3 and 4 contain four descriptors (Figure 5), while Classes 5 and 6 contain six other descriptors each. Each descriptor for class C can be used to define C. We call it a definitional descriptor. If a definitional descriptor D is used to define membership in C, all other descriptors can be deduced, leading to predictions of property values for members of C.

If a class C possesses many definitional descriptors, each can be used as a definition (recognition procedure). Since we allow exceptions, a conjunction of two definitional descriptors may offer a more reliable recognition.

The choice available among definitional descriptors offers flexibility in selection of the recognition procedure. Depending on the observable concepts available in various real life situations, one or another definitional descriptor can be used. Since alternative recognition procedures can be applied, missing values do not pose a problem in our approach in contrast to many machine learning systems, until all recognition procedures fail to apply for the lack of data.

It must be noted that each definitional descriptor D is sufficient to determine whether a given record belongs to a given class C only within the range of the hierarchy element, to which it belongs. To obtain a complete definition of C, we must use D in conjunction with the definition of the range of the hierarchy element. Similarly, conjunctions of two or more descriptors must be used to define the extent of a node at lower levels of taxonomy, each of the descriptors definitional for one node on the path from the root. Of course, making a choice of a definitional descriptors at each level, we can assemble the complete definition in a very flexible way.

If every object in class C satisfies descriptor D, but not the other way, D can be used to infer

properties of objects in C, but not to define membership in C. We will call such a descriptor D an inferred property. Regularities in the form of contingency tables of the "implication" type (see section 2.1) can be used to obtain inferred properties.

In taxonomy formation, definitional descriptors of concepts may lead to inferred descriptors for the concepts above them in the taxonomy, as explained in the section on taxonomy formation and in the comment (*) in the caption of Table 2.

In our approach, the knowledge about definitional and inferred descriptors guides taxonomy formation, as we explain later.

3.4 Empirical contents of a concept

Each concept can be characterized by its extent and intent. The extent is the set of all objects which are instances of the concept, while the intent is a set of property values (represented by descriptors) possessed by all objects in the extent.

We postulate that the empirical contents, which can be also called predictive contents of a concept, is proportional to the number of descriptors which can be deduced about a single object in C, after the membership has been determined. We can further postulate that the empirical contents is also proportional to the cardinality of the extent of the concept, because for each object in the extent, once it is recognized as a member of C by testing one definitional property, other descriptors can be deduced.

Empirical contents measures the significance of a concept. It can be used to decide on concept acceptance, and to make choices between concepts. By totaling the empirical contents of individual concepts we can also define the empirical contents of the whole taxonomy.

3.5 Taxonomy formation

We will now describe the process of transformation from the graph depicted in Figure 5 into a multi-level taxonomy, which is exhaustive and disjoint at each level. Arbitrarily, at the top of the taxonomy we may place any hierarchical element, gradually connecting other elements to the leaves of the nascent taxonomy. We position the classes with the greatest number of descriptors above those with less descriptors, to minimize the number of times each descriptor must occur in the taxonomy.

Based on the belief that the best concepts are those with highest empirical contents, we used a greedy algorithm to build the hierarchy. The algorithm searches for the hierarchy element with the largest number of shared attributes (In our example this is the element including Class 5 & 6), and places it at the uppermost level in the taxonomy. Under each of Class 5 and Class 6 the algorithm places the hierarchy element with the next greatest number of attributes (Class 3 & 4). Table 2 illustrates the placements. The greedy algorithm continues placing class pairs as above until all are exhausted. Some of the nodes, however, can be empty. Our algorithm examines this possibility as soon as each level is formed. It computes intersection of the value sets for each common attributes from each newly created node upward to the root of the taxonomy. If for a common attribute this intersection is empty, we know that no objects in the dataset could possibly belong to the lower class. This class is then eliminated. Class 6 contains the attribute Stem-Cankers with the value range (0), and under it Class 4 contains the same attribute with the value range (1,2). No records in the data can possibly be in both these classes simultaneously, therefore the intersection of Class 6 with Class 4 is eliminated (crossed off in Table 2).

All nodes an a given path must be compared, to detect empty intersections. For example in Table 2, Class 5 holds Stem-Cankers with a value range (1,2,3). In the next level, in Class 3 are only

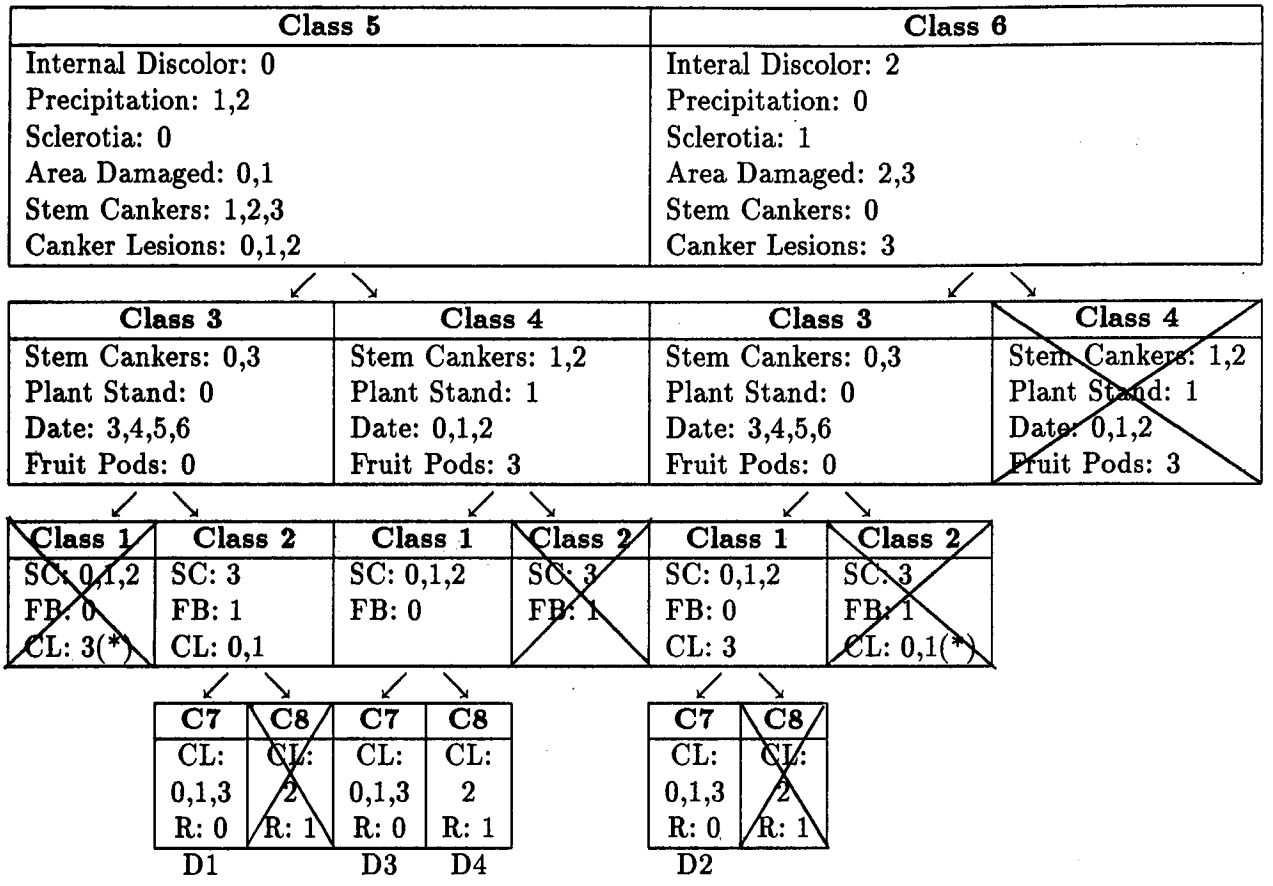

Table 2. The taxonomy generation process, depicted from the top (Classes 5 and 6) till the bottom (Classes 7 and 8). Empty classes are crossed out. Abbreviations: FB = Fruiting-Bodies; R = Roots; SC = Stem-Cankers; CL = Canker-Lesions
(*) The values of Canker-Lesion (CL-3 and CL-0,1) are added to Classes 1 and 2 under Class 3 due to a regularity found in the subset of data defined by the descriptor Fruit-Pods=0. That regularity, depicted in Figure 6, links Canker-Lesion to Fruiting-Bodies. Since the same values of Fruiting-Bodies define Class 1 and Class 2, the corresponding values of Canker-Lesion become inferred descriptors in the subclasses of Class 1 and Class 2 within Class 3.

those records from Class 5 where Stem-Cankers value was (3), the only value in common between the two classes. There can be no records in this part of Class 3 with Stem-Cankers(0), as none were in the class above. Going down one level more, to Class 1 and Class 2, we find that Class 1 under Class 3 may be terminated, as Stem-Cankers does not contain (3). Similar situations hold in several other paths, which also end with an empty set.

In Table 2 we have shown only the use of regularities found for all data, with one exception. The values of Canker-Lesion, shown under Class 3, come from a regularity found in the subset of data defined by the descriptor Fruit-Pods=0. That regularity, depicted in Figure 6, links Canker-Lesion to Fruiting-Bodies. Since the same values of Fruiting-Bodies define Class 1 and Class 2, the corresponding values of Canker-Lesion become inferred descriptors in the subclasses of Class 1 and Class 2 within Class 3.

When it is discovered that one of the classes in a class pair is empty and is eliminated, the descriptors of the remaining class are added to the descriptor list of the parent class, expanding the intent and the empirical contents of the parent node. The descriptors acquired from the lower class become inferred properties in the parent class, because in this situation all objects in the parent

class also belong to the remaining lower class. We see in Table 2 that Class 4 is eliminated under Class 6, therefore any object with Class 6's descriptors must also hold in Class 3. However, Class 3 also contains objects that belong in Class 5, so the merged properties from Class 3 to Class 6 cannot be definitional, but are inferred. We can similarly infer the descriptors of Class 7 and Class 1 under Class 6 into the set of descriptors of Class 6. Table 3 includes all definitional and inferred descriptors of Class 6, produced in our walk-through example.

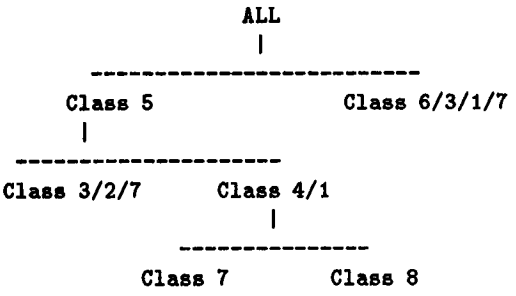

Figure 7. The finished taxonomy, after pruning empty nodes in Table 2 and merging the remaining classes upward. While the leaves correspond to the natural diseases of the soybean data, internal nodes may describe 'superclasses' of diseases.

After all the pruning, we are left with the four nodes at the bottom level in Table 2. These nodes, along with three internal nodes (ALL, Class 5, Class 4&1 under Class 5) form the taxonomy. We can hypothesize that this taxonomy describes the natural divisions of the soybean database's diseases. It turns out that the extents of the four leaf concepts in our taxonomy are equal to the four diseases, listed under each leaf concept in Table 2 as D1 through D4. The intents of each leaf concept describes the properties common for that disease. We can hypothesize that the internal nodes correspond to natural classes of diseases. Our approach leads to a claim about empirical contents for each node in the hierarchy.

Our algorithm places concepts with higher empirical contents at the top of the taxonomy the narrower range attributes higher up in the taxonomy. This way the number of descriptors which must be stored in the taxonomy is minimized. Our taxonomy, shown in Table 2 requires 33 descriptors, while a taxonomy which puts classes with the fewest attributes at the top (class 7&8, followed by class 1&2, class 3&4, and class 5&6 at the bottom, requires 44 descriptors.

4 Comparisons

Our method is based on the idea that regularities are of prominent importance among the types of knowledge inherent in data. Concept taxonomies are secondary, as they can be formed from regularities in the form of special contigency tables. Previous conceptual clustering systems use various methods to find cluster taxonomies, but as they do not consider regularities, their approximation methods may miss important knowledge useful in taxonomy formation.

Linear Clustering, introduced by Piatetsky-Shapiro & Matheus (1991), uses regularities, but in another way. When several linear patterns are detected in the same domain, clusters are formed that capture the subdomains in which each linear pattern is unique.

Several approaches combining Bayesian and expert knowledge in the form of domain heuristic measures for classifying data (Wu, et al., 1991). In distinction to those approaches which require expert domain knowledge to guide classification heuristics, our approach requires no prior domain knowledge to determine meaningful classes in the data.

	OUR SYSTEM		COBWEB		
Attribute	Value	Role	Value	[P(value|D2), P(D2|value)]	Role
Precipitation	0	Definitional	0	[1.0 , 1.0]	Definitional
Temperature	1,2	Inferred	2	[0.6 , 1.0]	Partial Def.
Stem Cankers	0	Definitional	0	[1.0 , 1.0]	Definitional
Fruit Pods	0	Inferred	0	[1.0 , 0.5]	Weak
Canker Lesion	3	Definitional	3	[1.0 , 1.0]	Definitional
External Decay	0	Inferred	0	[1.0 , 0.48]	Weak
InternalDisclr	2	Definitional	2	[1.0 , 1.0]	Definitional
Sclerotia	1	Definitional	1	[1.0 , 1.0]	Definitional
Area Damaged	2,3	Definitional			Not included
Plant Stand	0	Inferred			Not included
Date	3,4,5,6	Inferred			Not included
Fruiting Body	0	Inferred			Not included
Roots	0	Inferred			Not included

Table 3. An example of one concept (Disease 2 - Charcoal Rot) found by building our regularity hierarchy. The results of the COBWEB system are similar, but our method finds more inferred descriptors, as well as one definitional descriptor, Area Damaged(2,3). The attributes Temperature and External Decay were found as inferred attributes for this concept in a subrange of records by our method, and were not included in Table 2 due to space restrictions. P(A|B) means the probability of being in A for the objects in B.

In Table 3 we compare the results of our algorithm for clustering regularities found in the soybean data set to the available results obtained by COBWEB (Fisher, 1987). The comparison concerns Class 6, which corresponds to Disease 2, which is called Charcoal Rot. Notice a substantial increase of empirical contents reached in our approach, which shows in the number of additional predictions possible for five attributes of the inferred type, not included by COBWEB.

COBWEB employs a probabilistic approach to determine classes. Objects are incorporated into classes based on normative values for attributes in a given class. Since objects are added incrementally to the conceptual hierarchy, change operators are used to merge, split and delete nodes from the hierarchy. This is in contrast to our algorithm, which is not incremental. Because we are forming a hierarchy from regularties already discovered by a database mining system, we have found that hierarchy change operators are unneccessary when forming a taxonomy of regularities. Where empty classes are detected in the nascent taxonomy, a merge of descriptors occurs during hierarchy formation. If none of a pair of classes added at each step is empty, a natural split occurs in the hierarchy formation.

5 Summary

In this paper we have presented a theory of and an algorithm for conceptual hierarchy formation from law-like knowledge presented in the form of contingency tables. We used the soybean database in our walk-through example. It turned out that four diseases hidden in the soybean data coincide with the four leaves in the taxonomy generated by our algorithm.

We argued that regularities are the basic form of knowledge and that knowledge contained in concepts is secondary to knowledge contained in regularities. Empirical contents of regularities is

typically greater than empirical concepts of concepts. Autonomous discoverer will see reasons to claim empirical contents of some concepts when coexisting properties are revealed in the form of regularities. We argued that better concepts are those with higher empirical contents and therefore a knowledge discovery system which can notice that contents, can play central role in concept formation.

In contrast to fixed rules which define concepts in many machine learning systems, in our approach the choice available among definitional descriptors offers flexibility in selection of the recognition procedure. The same taxonomy can be used in different ways in various applications. Depending on the observable concepts available in a given situation, different definitional descriptors can be used. Since alternative recognition procedures can be applied, missing values do not pose a problem in our approach in contrast to many machine learning systems.

References

Fisher, D.H. 1987 . Knowledge Acquisition Via Incremental Conceptual Clustering *Machine Learning 2* 139-172.

Hoschka, P. & Klösgen, W. 1991. A Support System for Interpreting Statistical Data, in: Piatetsky-Shapiro G. & Frawley W. eds *Knowledge Discovery in Databases*, Menlo Park, CA: AAAI Press, 325-345.

Klösgen, W. (1992). Patterns for Knowledge Discovery in Databases in: ed *Proceedings of the ML-92 Workshop on Machine Discovery (MD-92)*, Aberdeen, UK. July 4, p.1-10.

Langley, P., Simon, H. A., Bradshaw, G. L. & Żytkow, J. M. 1987. *Scientific discovery: Computational explorations of the creative processes.* Cambridge, MA: MIT Press.

Michalski, R.S. & Chilausky, R.L. 1980. Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis, *Int.J. of Policy Analysis and Info. Systems*, 4, 125-161.

Nordhausen, B. & Langley, P. 1993. An Integrated Framework for Empirical Discovery, *Machine Learning*, 12, 17-47.

Piatetsky-Shapiro, G. ed. 1991. *Proc. of AAAI-91 Workshop on Knowledge Discovery in Databases*, Anaheim, CA, July 14-15, 1991.

Piatetsky-Shapiro, G. & Frawley, W. eds. 1991. *Knowledge Discovery in Databases*, Menlo Park, CA: AAAI Press.

Piatetsky-Shapiro, G. & Matheus, C. 1991. Knowledge Discovery Workbench: An Exploratory Environment for Discovery in Business Databases, in Piatetsky-Shapiro G. ed. *Proc. of AAAI-91 Workshop on Knowledge Discovery in Databases*, 11-24.

Shen, W. 1993. Discovery as Autonomous Learning from the Environment. *Machine Learning*, 12.

Stepp, R.E. 1984. Conjunctive Conceptual Clustering: A methodology and experimentation, Ph.D. dissertation, Dept. of Computer Science, University of Illinois, Urbana.

Wu, Q., Suetens, P. & Oosterlinck, A. 1991. Integration of Heuristic and Bayesian Approaches in a Pattern-Classification System, in Piatetsky-Shapiro G. & Frawley W. eds. *Knowledge Discovery in Databases*, Menlo Park, CA: AAAI Press, 249-260.

Żytkow, J., & Zembowicz, R., (1993) Database Exploration in Search of Regularities, *Journal of Intelligent Information Systems*, 2, p.39-81.