Granularity refined by knowledge: contingency tables and rough sets as tools of discovery

Jan M. Żytkow
Department of Computer Science, Univ. of North Carolina, Charlotte, NC 28223
and Institute of Comp. Science, Polish Academy of Sciences, Warsaw, Poland

ABSTRACT

Contingency tables represent data in a granular way and are a well-established tool for inductive generalization of knowledge from data. We show that the basic concepts of rough sets, such as concept approximation, indiscernibility, and reduct can be expressed in the language of contingency tables. We further demonstrate the relevance to rough sets theory of additional probabilistic information available in contingency tables and in particular of statistical tests of significance and predictive strength applied to contingency tables. Tests of both type can help the evaluation mechanisms used in inductive generalization based on rough sets. Granularity of attributes can be improved in feedback with knowledge discovered in data. We demonstrate how 49er's facilities for (1) contingency table refinement, for (2) column and row grouping based on correspondence analysis, and (3) the search for equivalence relations between attributes improve both granularization of attributes and the quality of knowledge. Finally we demonstrate the limitations of knowledge viewed as concept approximation, which is the focus of rough sets. Transcending that focus and reorienting towards the predictive knowledge and towards the related distinction between possible and impossible (or statistically improbable) situations will be very useful in expanding the rough sets approach to more expressive forms of knowledge.

Keywords: knowledge discovery; knowledge refinement; automated discovery; granularity; indiscernibility; approximation; contingency tables; rough sets.

1. ROUGH SETS REPRESENTATION BY CONTINGENCY TABLES

Both rough sets and contingency tables are founded on a similar idea of granular empirical data. Both approaches use the representation of empirical objects by n-tuples or vector of attribute values. This representation is common in statistics, databases, machine learning, pattern recognition and many other areas. Specific to rough sets and contingency tables approaches are indiscernibility classes. Objects are indiscernible if their property tuples are the same. That can happen in practice when domains of attributes contain small numbers of values. That happens in many databases, especially when attribute values are determined with limited accuracy. But if that is not the case, binning also known as discretization, can be used.

Rough sets were introduced in 1980's by Zdzislaw Pawlak (1991), while the history of contingency tables is several decades older. They play a major role in the theory and practice of statistics (Gokhale & Kullback, 1978; Fienberg, 1980; Whittaker, 1990). Contingency tables have been used as a general tool for expressing knowledge and for knowledge refinement in 49er (Zembowicz and Zytkow, 1993, 1996).

In this paper we place rough sets in the framework of contingency tables. We argue that contingency tables provide an added value of statistical techniques and a broader perspective on knowledge that can help to transcend the current rough sets applications (Ziarko, 1994; Polkowski & Skowron, 1998, 1998A)

1.1. Contingency tables

In this paper we will consider data organized into a single relational table R with the attributes A_1, A_2, ..., A_M. Let V_1, V_2, ..., V_M be the corresponding sets of values for each attribute. This will ensure compatibility with the rough sets treatment of data.

The space W of all possible events (situations) is the Cartesian product $W = V_1 \times V_2 \times \ldots \times V_M$, which includes all possible combinations of attribute values. While a data table is a convenient way of storing data, a Cartesian product provides an important theoretical perspective. We will explain that perspective throughout this paper. Contingency

Further author information: e-mail: zytkow@uncc.edu; http://www.cs.uncc.edu/žytkow

tables are defined on Cartesian products of values of all attributes. For each relational table R, a frequency table is a mapping from W to the set of natural numbers, associating each possible event $(v_1, \ldots, v_n)$, $v_1 \in V_1$, ..., $v_M \in V_M$, with the number of occurrences of the corresponding record in the database. Frequency tables, also called contingency tables, or tables of actual counts, are very useful, because they can represent regularities in the domain represented by data. For instance, fields which are occupied by one or more records indicate possible events, while if sufficiently many data are available, empty fields (with zero records) indicate impossible events. Different patterns that distinguish combinations of values that occur in the domain, vs. those that do not occur, lead to different forms of knowledge, such as equations, equivalence and subset relation.

But in many domains the knowledge is not black-and-white. All events are possible but not equally probable. Contingency tables are commonly used to express statistical relations between attributes. Consider university records that include for each student information such as high school grade point average (HSGPA) and total credit hours taken at the university in the entire course of study (CURRHRS). The following contingency table provides, for each pair of values of CURRHRS and HSGPA indicated at the margins, the number of students in the university records who share these values. The numbers describe a cohort of students admitted in a particular year. For example, there are 92 records with CURRHRS of 120+ and HSGPA=B.

CURRHRS						
120 +	0	11	102	92	73	
90-119	0	13	67	26	32	
60-89	0	6	54	25	25	
30-59	0	34	100	32	22	
1-29	4	164	243	60	29	
0	0	14	17	5	3	
	F	D	C	B	A	HSGPA

Such a table has been also called actual distribution, cross-tabulated data, cross-classified data, two-way frequency table.

Although 2D tables are the most commonly used, CTs can be n-dimensional. 1D table is a histogram, which is a mapping from the set of attribute's values to the number of occurrencies of each. In our example those numbers are the totals of all rows or all columns. Many-dimensional tables were rarely used in practice, as their size grows exponentially with the number of dimensions, as they were sparse for relational tables with small numbers of records that were common in the past, and as their printing and viewing is awkward.

A typical size of 2-10 values per attribute makes a 2D table manageable and suitable for moderate-size data. In our example, the number of original values of HSGPA and CURRHRS is above 100 for each attribute. Those values have been grouped into 5 and 6 categories shown in the table. The use of computers and large data sets will increase the admissible number of values. Attribute values can be discretized (binned, grouped) when numbers of values are large. Grouping may be aided by prior knowledge: an attribute may be on the ordering scale, a taxonomy of values may be provided or an equivalence to another attribute can be used.

2. ROUGH SETS AND CONTINGENCY TABLES

In the theory of rough sets, the source of data is also a single relational table R. It is called an information system. The domain of each attribute A used in R is expressed as V_A. In typical applications, the cardinality of each V_A is small, for the reasons similar to contingency tables (cf. many articles in Ziarko, 1994; Polkowski & Skowron, 1998, 1998A).

Indiscernibility relation on R is an equivalence relation on tuples such that all identical tuples belong to the same class. A B-indiscernibility relations can be defined for each subset B of attributes in R.

In a contingency table, each n-tuple of values forms a class of indiscernibility. Some of those classes may not contain any data. Such classes provide an important message, for they allow us to distinguish what can exist from what cannot. When those empty classes are ignored, we miss the empirical perspective on knowledge. Recognition that some logically possible situations are physically not possible leads to knowledge that has predictive value. Each possibility that is excluded narrows down the predictions and makes knowledge more deterministic.

The paradigmatic application of rough sets is concept learning from examples, called concept approximation. Given a subset C of tuples in R, the task is finding the most accurate description of C in terms of attributes available in R. Such a description should capture the tuples in C and exclude all the remaining tuples.

The task is essentially the same as the main task in machine learning, which is concept learning from examples. The treatment of indiscernibility, however, is different. In machine learning, two tuples which are equal on all attributes but differ on their class assignment are called inconsistent. In distinction to machine learning which treats data inconsistency as inconvenience which must be avoided, for instance by expert advise, rough sets developed a formalism that systematically uses data inconsistency. All unproblematic (consistent) examples of C determine its lower approximation C_*, while all inconsistent cases belong to the boundary. The lower approximation together with the boundary form the upper approximation C^* . The rough sets approach distinguishes between strong rules that capture lower approximation and possible rules that define the upper approximation.

2.1. Indiscernibility in CTs

Data indiscernibility can be rare when each tuple includes values of many attributes. It becomes common, however, in case of data reduction when only a subset of attributes is considered in attempts at a possibly simple knowledge. B-indiscernibility applies to tuples that become equal on a reduced set B of attributes. Many rough sets techniques are used for data reduction, by determining special subsets of attributes called reducts. A classical reduct retains a minimal set of attributes that retains indiscernibility of the original data, but other reducts eliminate further attributes and thus increase indiscernibility.

Let CT(B) be a contingency table on the set B of attributes. Each cell in a CT(B) represents a potential class of B-indiscernibility. While in the theory of Rough Sets, indiscernibility is defined for concrete tuples available in the data, a CT(B) expands B-indiscernibility to the space of all logically possible events, defined by the Cartesian product of domains of attributes in B.

2.2. Representing concept approximations in CTs

Rough sets techniques were mainly applied in concept approximation, which is concept learning from examples. Many worthwhile results were reached, competitive in accuracy with approaches of machine learning.

Let us consider the following contingency table. It was created for a simple comparison of a statistical approach to concept learning based on contingency tables with that of rough sets.

Number of occurrences of C	5	3	4	2	5	9	6	2	0	0
Number of occurrences of ¬C	0	0	1	2	3	8	7	8	10	4
Values of A	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
	lower approx.		boundary of C						complement of C	
	upper approximation of C									

Table 1. Concept C learned with the help of concept A

This made-up table emphasizes the phenomenon common in probabilistic situations, where concept boundaries cover almost all indiscernibility classes.

We can distinguish the following situations:

1. Concept boundary extends to all values of A (A can be a Cartesian product of more than one attribute). Rough sets approach is not going to work, as the lower approximation and the complement of C is empty. Statistical methods and probabilistic predictions are the best tool.

2. Concept boundary is non-empty but relatively narrow. This is the main area of applications of rough sets techniques. Here rough sets offer a simple yet competitive approach to concept approximation by probabilistic distributions. It seems that in this area they can provide results better than machine learning.

3. Concept boundary is empty. Here rough sets approaches lose competitive advantage over machine learning, but they still apply.

Notice that the analytical methods based on contingency tables and used by 49er can handle all three situations. In the third case, for instance, 49er would notice the equivalence relation between C and a binary partition of the values of A.

Notice an advantage of statistical significance analysis. When the numbers of occurrences per cell are small, the qualification of a cell (indiscernibility class) to the lower approximation, the boundary, or the complement of C is objectionable. Additional records are very likely to change that qualification.

2.3. Representing reducts in CTs

Reduction of the number of attributes reduces the volume of data considered in concept learning and also improves performance of many learning methods which depend heavily on the number of attributes. We can distinguish two basic strategies:

Strategy 1: remove the attributes which are not used to make distinctions between tuples in C and in $\neg C$. This can be often done in many ways, and the numbers of reducts can be very large.

Strategy 2: (49er, ID3, etc.) Add attributes, as needed to discern between tuples in C and in $\neg C$.

As reducts can be statistically superficial, the second strategy has distinct advantages. Suppose that the data include a few random attributes. Those random attributes can and probably will be used in a reduct, as they provide distinctions between the tuples that exist in the data. Those distinctions, however, are statistically insignificant and should not be made.

But reducts are objectionable on another ground, too. Attributes that are redundant, because other attributes provide similar distinctions, may still be very valuable as alternatives that can be jointly used to make a definition more robust. We can improve Strategy 2:

Strategy 2A: (49er) If two attributes are approximately equivalent, use both of them in the definition. Such definitions ar resilient to missing data and are statistically significant even on smaller sets of data.

3. INDUCTIVE GENERALIZATION

It is important to understand the method of data collection before attempting at generalization which would make claims about a real-world domain. Data can be

1. a set of carefully selected examples; this was typical at early stages of machine learning, when a few examples were typically used. Near misses were particularly to avoid over-generalization.
2. a set collected for another purpose, typically to support the business operation of data owners; such data may not be a representative sample, so we must be careful about the target of generalization.
3. a statistically valid sample of a population; here statistical methods are particularly useful.
4. data about natural system, which do not represent a population. We typically do not seek statistical results, and even quantum indeterminism is statistically different from population statistics in ecology or sociology.

While concept approximation methods are recently successfully applied to large datasets, at early stages, similarly to early machine learning, the rough sets mechanism was used to learn from small well-prepared datasets, often called decision table. Data reduction to decision tables is still one of the key methods in the repertory of rough sets techniques.

It is obvious that a contingency table represents the dataset R from which it was generated. But can it be used to infer knowledge about the domain represented by R? Let A_{ab} be the number of records that include values a and b of two attributes. Let n be the total number of records in data. We can make an instantaneous inductive generalization and claim that $p_{ij} = A_{ij}/n$ estimates joint probability distribution of both attributes in the population represented by R.

Such a generalization can be very useful, but is not always valid. First, the number of data must be sufficient so that probability values are significant. As a rule of thumb, the average number of data per category (cell) should be larger than some minimum of three or four.

Second, not every dataset represents a population. In engineering and sciences such as physics, chemistry, the notion of population is rarely useful. When data have been procured by an experimentation or observation strategy which selects the values of attribute A, the histogram of A represents the experimentation strategy, not a population. Instead of a joint probability distribution, conditional probabilities of other attributes, given the value of A, represent knowledge of D.

Third, populations can change. As always, how far can we generalize is subject to future empirical verification.

But even if drawn from the same population, the actual frequency tables may be different in different samples. On the flip side, a given table can come from many distributions. The attributes used in the table can be independent, while an appearance of dependence has occurred by a random fluctuation. Since attribute independence is common, it is customary to evaluate that possibility. A 2D table shows the joint distribution of the values of both attributes. This information is additional to histograms of both attributes. When attributes A and B are independent, their joint distribution should be close to the product of histograms of both variables. The expected number of records with $A = a$ and $B = b$ is

$$E_{ab} = \frac{h(A,a) \cdot h(B,b)}{n},$$

where $h(A,a)$ is the number of records in dataset D with the value a of attribute A, and similarly $h(B,b)$. E_{ab} is usually called the expected distribution.

How different can A_{ab} be from E_{ab} so that we can still claim that E_{ab} is the true distribution? Here comes the theory of statistics. It can tell how probable it is that a sample generated from a specified probabilistic distribution has a parameter value which exceeds a specified threshold. An efficient, but approximate way of making such an estimate starts from the value of chi-square:

$$\chi^2 = \sum_{a,b} \frac{(A_{ab} - E_{ab})^2}{E_{ab}}$$

Given the χ^2 and the number ν of degrees of freedom in the table (in our example $\nu = (M_{row} - 1)M_{col} - 1) = 20$), the probability Q can be computed (Jobson, 1991), estimating the likelihood that a table such as A_{ab} has been randomly drawn from distribution E_{ab}.

In our example, $\chi^2 = 229.0, Q = 1.66 \cdot 10^{-32}$. Such a low value of Q tells that it is practically impossible that HSGPA and CURRHRS can be independent. Thresholds of significance vary from traditional 0.01 or 0.05 to much smaller, more demanding numbers in massive search for regularities in KDD (Zembowicz and Zytkow, 1996).

When a previously estimated joint distribution of A and B is available, we may use it as E_{ab} to estimate the probability that A_{ab} comes from that distribution. Lacking knowledge of the relation between the attributes, the current A_{ab} table can be treated as an exact representation of the population.

Notice that all the problems with generalization of a contingency table are also problems of all other inductive generalization techniques, including rough sets.

4. PREDICTIVE POWER OF CTS

A contingency table can be used to reason about objects in the domain (Gokhale & Kullback, 1978; Bhattacharyya & Johnson, 1986). Treated as a regularity, contingency table can provide many predictions. In a 2D table, p_{ab} are probabilities that randomly selected objects will have the pair of values a and b. If the value of one attribute is known, then the row or column corresponding to that value can be used to make predictions of values of the other attribute.

For instance, for an incoming student with the value A of HSGPA, we can predict probabilistically the value of CURRHRS. The value 120+ will occur with probability 0.4 (73/184).

Predictive strength of a CT can be measured by various criteria, such as Cramer's V (Jobson, 1991) and lambda measures (Goodman and Kruskal, 1954). For a given $M_{row} \times M_{col}$ contingency table, Cramer's V, is defined as

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

In our example, $V = 0.19$. When the values of one attribute can be uniquely predicted from values of the other, Cramer's $V = 1$. On the other extreme, when the actual distribution is equal to the expected, then $\chi^2 = 0$ and $V = 0$. V does not depend on the size of the contingency table nor on the number of records. Thus it can be used to compare regularities found in different subsets and for different combinations of attributes.

Missing values can be ignored and they do not harm many statistical tests. But in addition to that, contingency tables are convenient for determining why values are missing. When missing values are treated as special values that are included in a table, it is possible to find why the values are missing.

The additional information available in contingency tables provides advantage over other inductive methods, including rough sets, because it allows to assess in probabilistic cases the predictive power of statistical regularities.

5. IMPROVING GRANULARITY BY FEEDBACK FROM KNOWLEDGE

Discretization and binning can be done apriori or can be based on background knowledge. But in many cases, techniques based on search for knowledge with the use of contingency tables provide a more informed binning, that leads to better quality knowledge. We will briefly outline two such methods.

5.1. Determination of a functional relation in data

In databases, typically a small discrete set of values is permitted for each attribute. Given a small set V_A of A values, and a small set V_B of B values, the product $V_A \times V_B$ is computationally manageable, as well as the corresponding frequency table F which is a mapping

$$F : V_A \times V_B \rightarrow N,$$

where N is the set of natural numbers, and $F(A_0, B_0) = n$ when n is the number of datapoints with $A = A_0$ and $B = B_0$.

If the number of distinct values of A and/or B becomes too large to compute the frequency table, the values of A and B can be grouped into bins $b(V_A)$ and $b(V_B)$, respectively. Aggregating the values of A into bins $b(V_A)$ of equal size ΔA means that the point (A_0, B_0) is replaced by a pair of integer numbers (k_A, k_B) such that A_0 is in the range from $A_{min} + k_A \Delta A$ to $A_{min} + (k_A + 1)\Delta A$, and B_0 is in the range from $B_{min} + k_B \Delta B$ to $B_{min} + (k_B + 1)\Delta B$, where A_{min} and B_{min} are the smallest values of A and B, respectively. The frequency table $F(k_A, k_B)$ can be interpreted as a grid $b(V_A) \times b(V_B)$ imposed on the data (A, B) and defined by ΔA and ΔB.

Binning the original variables A and B into bins $b(V_A)$ and $b(V_B)$ helps to determine functionality in data which include error and/or noise. If the grid size ΔB is comparable to the error δ_B, the requirement of maximum difference between B values corresponding to the same value of A can be replaced by: all points in the same A-bin k_A must lie in the adjacent B-bins (for example, in bins $k_B - 1$, k_B, $k_B + 1$). This works only if the bin sizes ΔA and ΔB are not smaller than corresponding errors; otherwise the functionality test may fail because points in the same A-bin may not lie in *adjacent* B-bins, even if the original data follow a functional dependency plus error. On the other hand, if the sizes ΔA and ΔB are too large, the test could assign functionality to data that intuitively should not be described by a function. In the extreme case, when ΔB is larger than $B_{max} - B_{min}$, all points always lie in the same B-bin, because one bin includes all values of B.

The problem of background noise can be alleviated if one tests the adjacency only for cells that contain an above average number of points. This noise subtraction works effectively only if the background noise is not stronger than the regularity itself. But if the noise is stronger, there is hardly any chance to find a regularity.

If the errors δ_A and δ_B are known, they can be used as the corresponding bin sizes ΔA and ΔB. But if they are unknown or uncertain, the proper grid sizes must be estimated. Let us define the "density" ρ as the average number of points in all cells which contain at least one point. As long as the grid size ΔA is smaller than Δ, ρ is equal to one. The density ρ starts to grow when ΔA becomes greater than Δ. Note that in distinction to the true density equal the number of data points divided by the total number of cells, ρ does not depend on the number of A-bins

in the following sense. If we extend the range of A by adding more points which are evenly distributed, and keeping ΔA constant, then in our ideal example, ρ will have exactly the same values. This is important because it means that the "density" measure ρ does not depend on the range of A or B as long as bin sizes are the same. Therefore ρ can be used to determine ΔA and ΔB.

Let us consider an algorithm that determines the grid size. It starts from some small initial sizes ΔA and ΔB and changes these sizes until a criterion for "density" ρ is satisfied.

Note that for the linear dependency $A = aB + b$ and evenly distributed points the "density" ρ becomes larger than 1 when $\Delta A > \Delta = V_A/N$ or $\Delta B > a\Delta = V_B/N$, where N is the number of points. Based on these observations, we have developed the following algorithm to determine of the grid size in the case of unknown error. ρ_0 is the minimum required "density" of points in non-empty cells.

Algorithm: Determine grid size
$\Delta A \leftarrow V_A/2N\rho_0, \quad \Delta B \leftarrow V_B/2N\rho_0$
$\rho \leftarrow N$ / (# of non-empty cells)
if $\rho \leq \rho_0$ **then**
  repeat
    $\Delta A \leftarrow 2\Delta A, \quad \Delta B \leftarrow 2\Delta B$
    $\rho \leftarrow N$ / (# of non-empty cells)
  until $\rho > \rho_0$
else
  repeat
    $\Delta A \leftarrow \Delta A/2, \quad \Delta B \leftarrow \Delta B/2$
    $\rho \leftarrow N$ / (# of non-empty cells)
  until $\rho < \rho_0$
  $\Delta A \leftarrow 2\Delta A, \quad \Delta B \leftarrow 2\Delta B$
end if
end algorithm

The initial values of ΔA and ΔB are chosen to be $V_A/2N\rho_0$ and $V_B/2N\rho_0$, respectively, because for monotonic functions with more or less evenly distributed points the resulting density ρ would be close to ρ_0. The additional factor 1/2 was introduced to avoid decreasing the grid size in cases when initial ρ is only slightly larger than ρ_0. Decreasing the grid size is more costly than increasing it, because the latter operation can be performed on the existing grid from the previous step, while in the former case the density grid must be build from data. From the definition of ρ one can see that its value is never smaller than 1. If the grid size is too small, the value of ρ is about 1.

The initial values of ΔA and ΔB are chosen based on the case of a monotonic function with evenly distributed points. However, as our algorithm does not depend on these assumptions, in many situations ΔA and ΔB would be then changed: either increased or decreased, until the resulting "density" parameter ρ gets close to the required value ρ_0.

When ρ becomes significantly greater than 1, say its value is about 2, it means that there is on average about two points per each non-empty cells. At that moment for most data one can analyze functionality based on that grid size, therefore a good default value for ρ_0 is around 2. However, when the distribution of data is very uneven, ρ_0 should be increased.

5.2. Refinement of 2*2 CT patterns and concepts

2*2 contingency tables are based on attribute values aggregated into two disjoint and complementary subsets of "lower" and "upper" values for each attribute. They are an important tool for summarizing regularities, similar to linear correlations, because they are easy to interpret, to use for predictions or decisions, and take little space even if expanded to more than two dimensions. 49er makes the initial aggregates *a priori*, using the histogram of each attribute. By the changes of the concept definitions for the "lower" and "upper" values, an initial 2*2 pattern can be strengthened. New concepts of "lower" and "upper" values obtained as a result of refinement can be more meaningful.

The refinement of 2*2 tables uses the hill-climbing search control, which stops when strength of the regularity can be improved no more. For ordinal attributes the resulting algorithm is fast ($O(k^2 \times n)$, where n is the number of values of the attribute, k is the number of attributes). Concepts that result from CT refinement can be very useful, but at the same time the final CTs are the strongest regularities.

6. KNOWLEDGE BEYOND CONCEPTS

The main target of learning in the rough sets community are concept definitions, also called concept approximations. Some concepts can be functions, which are a special type of sets. Machine learning considers, in addition to concept learning, also dividing data into clusters and construction of cluster hierarchies. Concepts, clusters and their hierarchies are defined by systems of rules and alternative descriptions such as trees that can jointly describe many concepts. Both in machine learning and in rough sets communities the emphasis on concepts is complemented by the claim that concepts are the main target of learning and discovery.

In sciences such as physics and chemistry, that developed the most advanced forms of knowledge, concepts are secondary to laws and models. Discovery of a new law is important, while introduction of a new concept derives meaning from laws that can be expressed by that concept. Concept and cluster definitions are inferior in their predictive capabilities or even void of empirical, predictive contents. Let us consider the reasons in detail.

Empirical data are not just any set of tuples. Consider a simple case of one argument function $y = f(x)$ that captures the dependence of attribute y on x. The relevant empirical data, a set of pairs $F = \{(x_i, y_i) : i = 1, ..., n\}$ must represent real-world situations or events. Each datum must be obtained by observation or experiment. When the data are obtained by experiments, at most one variable can be controlled by the experimenter, while the value of the other is measured as a response of the empirical world to the situation created by the experimenter.

A set F of pairs $\{(x_i, y_i), i = 1, ..., n\}$ represents a function in the set-theoretic sense iff for all pairs in F, when $x_i = x_j$ then $y_i = y_j$. This definition does not make any claim about the points outside of F. Empirical data that justify a functional regularity such as an empirical equation, are empirical data that meet stronger requirements. The value of y should be unique not only in F, but in the domain represented by data. Experiments must recreate many times the situation in which the value x_i holds, and each time determine that the resultant value $y_i = \phi(x_i)$ of y is the same within measurement error. Notice that $\phi(x_i)$ indicates one result of experiment for the value x_i of the control variable x. At different times the result of experiment, and thus the value of $\phi(x_i)$ can be different. Experiments must determine that a unique value of y corresponds to every value of x:

$$\forall x \, (\exists y \, (y = \phi(x)),$$

$$\forall x (y_1 = \phi(x) \, \& \, y_2 = \phi(x) \; \rightarrow \; y_1 = y_2).$$

Let K be a piece of knowledge. Knowledge with empirical contents has several appealing properties. First, since empirical contents of K is non-tautological, situations inconsistent with K are logically possible. They should not occur or K is false. In other terms, there is no empirical contents in K if it does not exclude any logically possible situation. Second, every observational consequence can be viewed as a prediction.

Let us illustrate these notions, using the sentence $\forall x(A_1(x) \rightarrow A_2(x))$. This sentence can be used to express empirical contents, since it is not a tautology and is made of observational terms A_1 and A_2. Concrete predictions, $A_1(r) \rightarrow A_2(r)$, can be inferred for each record or entity r. Empirical contents of those predictions is very specific. Notice that $A_1(r) \rightarrow A_2(r)$, or its equivalent $\neg A_1(r) \vee A_2(r)$, do not predict an individual observation, but if it is further known that $A_1(r)$, a concrete prediction of $A_2(r)$ follows. Concrete individual observations have the logical form of ground literals (atomic sentences or their negations), which we call facts.

6.1. Concepts and knowledge

Formally, a concept can be represented by a predicate, such as $D(x)$. Since $D(x)$ contains x as a free variable, it does not have a truth value. $D(x)$ is satisfied by objects which belong to the extension of D and not satisfied by objects in the complement of D. Satisfaction of $D(x)$ by object r does not lead to any extra observational statement about r. In contrast, statements without free variables are either true or false. Consider the regularity "All ravens

are black," formally expressed as $\forall x(R(x) \rightarrow B(x))$. It is a statement which is false if a non-black raven exists. For each object r, the observational conclusion is $Rr \rightarrow Br$ or equivalently $\neg Rr \vee Br$. Knowing that r is a raven, we can predict that r is black.

Any observational language provides room for many concepts. For instance, in the language of R and B, we can define a concept of black non-raven: $\neg R(x) \vee B(x)$, a concept of black raven, a concept of raven which is non-black, and so forth. None of them contains any claims about the situation in the world. Some of them can be empty. Many are not useful. Empirical contents is present in regularities but not in concepts understood as predicates. Huge amount of concepts can be conceived in any dataset R, arguably all subsets of R. Concept definitions can be any predicated defined by attributes available in R and values of those attributes. But only a small subset of that enormous variety is useful.

Arguably, concept to be learned in machine learning and rough sets are useful, but that must be guaranteed by a human operator. In sciences and in mathematics concepts can be viewed as investments and they can be evaluated autonomously. They demonstrate their value by qualities of laws and theorems expressed in their terms. Generality, accuracy and utility of laws (theorems and models) justify the investment made by introduction of a concept used in those laws (theorems, models). Automated scientific discovery systems (BACON: Langley et al. 1987; IDS: Nordhausen & Langley, 1993; FAHRENHEIT: Zytkow, 1996) explore this view of concepts, keeping them only when justified by the simultaneously discovered knowledge.

6.2. Definitions by equivalence: empirical contents

Consider a classical definition of C that can be a result of concept learning

$$\forall x(C(x) \equiv D(x)),$$

where $D(x)$ is a Boolean expression formed from descriptors (statements such as $A_1(x) = a$) that use the attributes $A_1, \ldots, A_n$, and their values. Such a definition can be viewed as a special case of regularity. It can be empirically verified on the provided examples and counterexamples. Whenever an expert (teacher) is available, C is observational, so the definition has empirical contents. One prediction, of $C(r)$ or $\neg C(r)$, can be made and verified for each record. The definition can be used to predict class membership for other records, which haven't been classified by the teacher, but then it acts as a norm, not as a descriptive regularity. When C is not observational, $\forall x(C(x) \equiv D(x))$ does not have empirical contents, as no conclusion can be expressed purely in terms of $A_1, \ldots, A_n$. Such a definition cannot be falsified if we have no method other than $D(x)$ to assess membership in C. In conclusion, a concept definition by equivalence provides one prediction per object but may have no empirical contents if that prediction cannot be independently verified.

A knowledge miner will be well advised to seek multiple definitions of concept C, by predicates $E_1(x), \ldots, E_n(x)$. Jointly, multiple definitions of C possess empirical contents, expressed by the statement of their equivalence

$$\forall x(E_1(x) \equiv \ldots \equiv E_n(x)).$$

One observation $E_i(r)$ for object r leads to $n - 1$ predictions of facts. Alternative definitions make such concepts resistant to missing data.

In conclusion, it is possible to discover a concept with significant empirical contents by accumulating different definitions by equivalence, but there is little interest in ML, rough sets, and KDD communities in this approach. Exceptions are systems such as COBWEB (Fisher, 1987) and 49er (Zembowicz & Żytkow, 1996).

References

Bhattacharyya, G.K. & Johnson, R.A. 1986. *Statistical Concepts and Methods*, Wiley, New York, NY.

Fienberg, S.E. 1980. *The analysis of cross-classified categorical data*, MIT Press.

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

Gokhale, D.V. & Kullback, S. 1978. *The Information in Contingency Tables*, New York: M.Dekker.

Goodman, L. &Kruskal, W. 1954. Measure of Association for Cross Classification. Springer Series in Statistics, Springer-Verlag, 1979. Reprinted from the Journal of The American Statistical Association, **49**, 732-764.

Jobson, J.D. 1991, *Applied Multivariate Data Analysis*, Springer-Verlag.

Langley, P.W., Simon, H.A., Bradshaw, G., & Zytkow J.M. 1987. *Scientific Discovery; An Account of the Creative Processes*. MIT Press.

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

Pawlak, Z. 1991. *Rough Sets – Theoretical aspects of Reasoning about Data*, Kluwer.

Polkowski, L. & Skowron, A. (eds.) 1998. Rough Sets in Knowledge Discovery 1: Methodology and Applications, Physica-Verlag.

Polkowski, L. & Skowron, A. (eds.) 1998A. Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Physica-Verlag.

Whittaker, J. 1990. *Graphical Models in Applied Multivariate Statistics*, John Wiley & Sons.

Zembowicz, R. & Żytkow, J. 1996. From Contingency Tables to Various Forms of Knowledge in Databases. In Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI Press.

Ziarko, W. (ed) 1994. Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag.

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

Żytkow, J. 1996. Automated Discovery of Empirical Laws, *Fundamenta Informaticae, 27*, 299-318.