

MULTI-LEVEL DECOMPOSITION OF PROBABILISTIC RELATIONS

Stanislaw Grygiel, Martin Zwick, Marek Perkowski

December 2, 2002

Abstract

Two methods of decomposition of **probabilistic relations** are presented. They consist of splitting relations (blocks) into pairs of smaller blocks related to each other by **new variables** generated in such a way as to minimize a cost function which depends on the size and structure of the result. The decomposition is repeated iteratively until a stopping criterion is met. Topology and contents of the resulting structure develop dynamically in the decomposition process and reflect relationships hidden in the data.

1 INTRODUCTION

There exist two main approaches to the analysis of complex systems: **probabilistic** and **non-probabilistic**. Probabilistic approach assumes a knowledge of probability distribution over the variables of the system and the decomposition consists on determination of a set of simplest possible marginal probabilities. Non-probabilistic approach requires specification of the global relation over the variables of the system and the decomposition consists on determination of a set of simplest possible projected relations describing the system.

A system is here described by a contingency table. Each cell of the table contains the frequency observed for a particular combination of variable values. These frequencies can be normalized to the total number of observations and used to approximate the true probability distribution over the variables of the system. The system is referred to as probabilistic system.

In many situations it may be impossible or unreasonable to collect frequency information which is statistically reliable but it is relatively easy to collect meaningful information on the set-theoretic relation which exists between variables of the system. This corresponds to the situation where the cell frequency is either 0 or 1. This approach is also justified if cells of the contingency table contain only two distinct values of frequency (or values that are close to two distinct values) which may be assigned to two classes 0 or 1. In such situations the system is referred to as non-probabilistic.

Systems are also characterized as being either directed or neutral. In directed systems, variables are distinguished as being either independent variables (inputs) or dependent variables (outputs); in neutral systems no such distinction is made.

The decomposition of a complex system into an organized set of subsystems is motivated by the belief that a sim-

pler decomposed structure will better describe unobserved data (Occam razor principle) and will make it easier to understand relationships hidden in the data. Each subsystem can be viewed as defining a certain concept and the whole structure can be viewed as a higher level relation expressed in terms of these concepts (variables).

In this paper both probabilistic and non-probabilistic approaches will be considered and a new method of their decomposition will be presented.

The paper is organized as follows: Section 2 presents the related work, Section 3 presents the decomposition algorithms, Section 4 presents the cost function used in this paper, Section 5 discusses results and Section 6 concludes the paper.

2 RELATED WORK

The decomposition of complex systems was analyzed by many researchers in the past. In the terminology of systems science both decomposition and composition are known under the name of **reconstructability analysis** [Klir, 1985].

The approach presented by Ashby [Ashby, 1965], Klir [Klir, 1976], Krippendorff [Krippendorff, 1979], and Conant [Conant, 1972] consists of generating a lattice of possible decomposition structures and evaluating them in terms of both complexity and accuracy using either a **set-theoretic** (non-probabilistic) or an **information-theoretic** (probabilistic) approach. Both approaches are based on uncertainty measures, the first on Hartley's entropy [Hartley, 1928] and the second on Shannon's entropy [Shannon and Weaver, 1949]. A structure that results in the smallest complexity and yet describes the data with a high accuracy is selected to be the best solution.

An overview of decomposition approaches developed within the framework of general systems methodology (reconstructability analysis) is presented in [Zwick, 2001] and

an extended bibliography of reconstructability analysis (RA) as a whole is provided in [Klir, 1996]. Reconstructability analysis of directed systems was further clarified in [Zwick, 1995a], and some additional details on set-theoretic RA are presented in [Conant, 1981] [Zwick, 1995b].

In standard (both set-theoretic and information-theoretic) approaches to reconstructability analysis, the number of system variables remains unchanged in the process of decomposition. By contrast, the methods presented in this paper, which are based on ideas used in the decomposition of binary functions, introduce **new variables** in the decomposition process to reduce complexity. These methods, while inherently non-probabilistic in nature [Grygiel, 2000], can be applied also as approximate techniques for probabilistic systems.

3 DECOMPOSITION

Decomposition of relation consists in general on splitting a larger relational block into a number of smaller, possibly interrelated, blocks (see Figure 1).

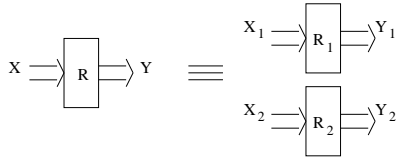


Figure 1: *Decomposition*

We will focus in this paper on decomposing one block into two smaller blocks in such a way as to reduce a certain cost measure. This process can be iteratively repeated until termination criterion is satisfied. The cost measure will be discussed in more details in Section 4.

In Figure 1 X denotes a set of independent and Y set of dependent variables of the relation. In the decomposed structure, if $X_1 \cap X_2 \neq \emptyset$ then decomposition will be called **non-disjoint**, otherwise we will call it **disjoint**. In the most general case both R_1 and R_2 have both dependent and independent variables. It is also possible that dependent variables of one block are independent in another block, for instance it may be $Y_1 \cap X_2 \neq \emptyset$.

The presentation of the decomposition algorithms in this paper will be based on tabular representation of relations (contingency tables). The software implementation of the algorithm, however, uses Ir-partition representation which is more memory efficient for manipulation of large multiple-valued relations [Grygiel et al., 1997],[Grygiel and Perkowski, 1998].

Other notations used in this paper are as follows: uppercase characters will denote sets and lowercase characters will denote variables. $|X|$ is the cardinality of the set X and $|x|$ is the cardinality of the variable x (number of values the

variable x can take). A relational/functional block with a set X of independent variables and set Y of dependent variables will be denoted by (X, Y) .

3.1 Relations

The following definition of relation will be used in this paper:

Definition 1 (relation) Let $S = \{S_i\}$ be a set of sets S_i . A subset R of the Cartesian product $S_1 \times S_2 \times \dots \times S_k$ will be called an **k -ary relation**. \square

Such a defined relation can always be represented by a two dimensional contingency table based on the fact that the Cartesian product operation is associative and Cartesian product is a set so we can reduce a k -ary relation to a binary relation $R \subseteq S_a \times S_b$ where S_a and S_b are sets of n -ary and m -ary tuples respectively and $n + m = k$.

Cells of the contingency table representing relation can either contain 0s and 1s or any numbers. The first case corresponds to non-probabilistic relations, 1s and 0s denoting tuples which are and are not contained in a given relation. The second case corresponds to probabilistic relations and numbers represent probabilities or frequencies associated with the corresponding tuples.

3.2 Decomposition Type I

This type of decomposition is always non-disjoint, i.e. the sets of independent variables of the decomposed blocks are non-disjoint.

Let $X = \{x_i\}, i = 1, \dots, n$, be a set of variables, X_1, X_2 be a partition of X , and Q_{x_i} be a set of values the variable x_i can take. If R is a relation based on the set of variables X then $R \subseteq Q_{x_1} \times \dots \times Q_{x_n} = Q_{X_1} \times Q_{X_2}$, where $Q_{X_j} = \{q_{kX_j}\}$, q_{kX_j} is a tuple (combination of values) for the variables of set X_j , and Q_{X_j} is a set of all tuples for the set X_j . Such defined relation R can be represented by a contingency table of $|Q_{X_1}|$ columns and $|Q_{X_2}|$ rows, each column corresponding to different tuple $q_{kX_1} \in Q_{X_1}$ and each row to a different tuple $q_{kX_2} \in Q_{X_2}$. Each cell of the contingency table contains 1 if the corresponding combination of tuples q_{iX_1}, q_{jX_2} belongs to the relation and 0 if it doesn't.

Definition 2 (column multiplicity) Column multiplicity μ is a number equal to the number of distinct columns in the contingency table. \square

Definition 3 (row multiplicity) Row multiplicity μ is a number equal to the number of distinct rows in the contingency table. \square

Column multiplicity μ is greater or equal to 1 (it is equal to 1 if all the columns are identical) and less then or equal to $|Q_{X_1}|$ (all the columns are different).

Our goal is to decompose the original relation R into two sub-relations R_1 and R_2 .

Let's create a new variable a such that $|a| = \mu$ and label each of μ sets of identical columns with a different value a_j of variable a .

Let $R_1 \subseteq Q_{X_1} \times Q_{\{a\}}$ be a relation created by extending every tuple $q_{iX_1} \in Q_{X_1}$ with the value a_j of variable a assigned to the column q_{iX_1} so that $R_1 = \{q_{iX_1} a_j\}$. To achieve our goal, R_2 has to be created in such a way that the composition of R_1 and R_2 results in R . The process of creation of R_2 will be defined by the following theorem:

Theorem 1 (decomposition) *Relation $R_2 \subseteq Q_{X_2} \times Q_{\{a\}}$ meeting the above conditions can be represented by a contingency table created from the original table for relation R by combining the identical columns of the table. The new columns will correspond to the tuples $q_{jA} \in Q_A$.*

PROOF It is enough to show that for every pair of tuples $q_{iX_1} a_k \in R_1$ and $q_{jX_2} a_k \in R_2$, the pair of tuples $q_{iX_1} q_{jX_2} \in R$ is part of the relation R ($q_{iX_1} q_{jX_2} \in R$).

Let's assume that there exists a pair of tuples $q_{iX_1} a_k \in R_1$ and $q_{jX_2} a_k \in R_2$, such that $q_{iX_1} q_{jX_2} \notin R$. The condition $q_{iX_1} q_{jX_2} \notin R$ means that the intersection of column q_{iX_1} and row q_{jX_2} in the original contingency table contains 0. The condition $q_{jX_2} a_k \in R_2$ means that the intersection of row q_{jX_2} and column a_k in the contingency table corresponding to R_2 contains 1. By the construction of R_1 , column a_k corresponds to the set of identical columns containing column q_{iX_1} . Hence, by the condition $q_{jX_2} a_k \in R_2$, intersection of the row q_{jX_2} and column q_{iX_1} contains 1 which is in contradiction with the assumption $q_{iX_1} q_{jX_2} \notin R$. This completes the proof. ■

The following example shows the process of decomposition of a relation (Figure 2).

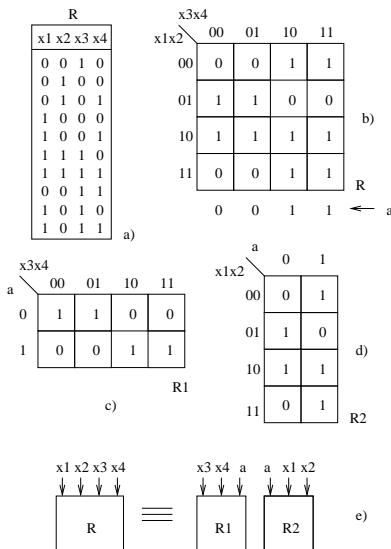


Figure 2: Decomposition type I.

Relation R is represented by tables in Figures 2a and 2b. A cell in the table in Figure 2b contains 1 if the corresponding tuple belongs to the relation and 0 otherwise. The column multiplicity index of the table in Figure 2b is equal to 2 and so is the cardinality of the new variable a . The table in Figure 2c corresponds to the block R_1 in Figure 2e, a cell of the table contains 1 if the corresponding combination of variable values exist in the table in Figure 2b. For instance, the columns $x_3x_4 = 00$, $x_3x_4 = 01$ in Figure 2b are labeled with $a = 0$ and $x_3x_4 = 10$, $x_3x_4 = 11$ with $a = 1$ so the cells corresponding to these combinations of values will contain 1 in table R_1 in Figure 2c. Other combinations of values of x_3x_4 and a will yield 0 in the table R_1 .

The table in Figure 2d which corresponds to the block R_2 in Figure 2e is created from the table in Figure 2b by combining identical columns and replacing variables x_3x_4 with a new variable a .

The same decomposition method can be used to decompose probabilistic relations, i.e. relations with probability or frequency associated with each tuple. For this kind of relations however, the probabilities have to be discretized before decomposition can be performed. The most often used discretization method, uniform binning, divides the space of each variable values into a number of equally sized bins. Another type of discretization methods are methods based on the entropy measure [Catlett, 1991], [Fayyad and Irani, 1993] which use minimum entropy criterion to assign values to different bins. They often yield better results.

Figure 3 depicts decomposition process of such a relation.

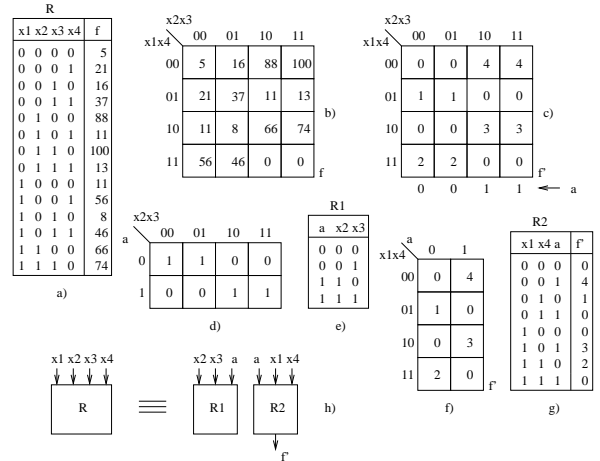


Figure 3: Decomposition type I, probabilistic relation.

3.3 Decomposition Type II

The decomposition described in the previous section resulted always in a non-disjoint solution. In this section we will describe a decomposition which may result in either disjoint or non-disjoint solutions. The main distinction of the decompo-

sition described in this section is that it is a functional decomposition. We decompose not the relation itself but the probability density function defined by the frequencies or probabilities in the contingency table describing the relation. The result of the decomposition can again be viewed as a neutral relation. The type II decomposition procedure is depicted in Figure 4.

The relation used in this example is the same as the one in Figure 3. The relation to be decomposed is defined in Figures 4a and 4b. The result of uniform binning to five values is shown in Figure 4c. The decomposition alone is performed in the manner similar to the decomposition described in the previous section. The difference between the two is the way the block R_1 is created. For decomposition of type I the new variable a is always an independent variable in R_1 . For decomposition of type II described in this section the new variable a is always a dependent variable in R_1 . In block R_2 , variable a is independent in both, type I and type II decompositions.

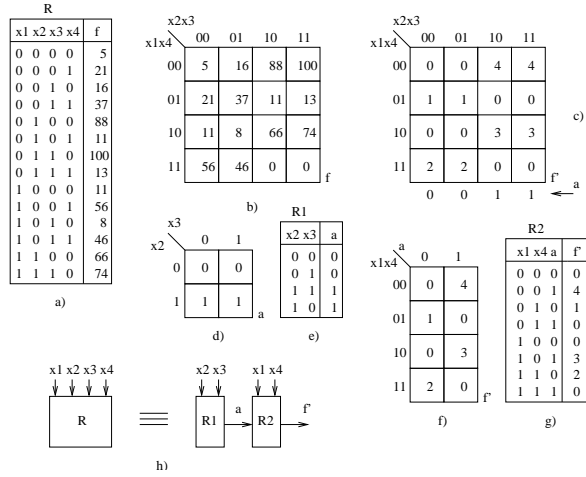


Figure 4: Decomposition type II.

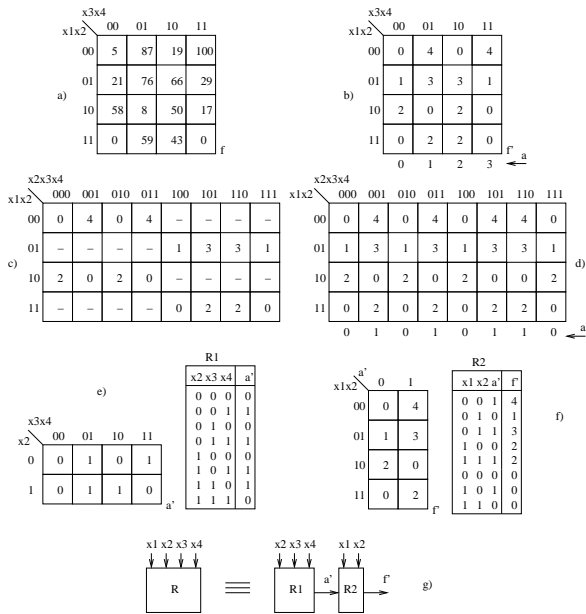


Figure 5: Non-disjoint decomposition type II.

The decomposition in Figure 4 is a disjoint decomposition because the sets of independent variables of blocks R_1 and R_2 are disjoint. In Figure 5 we show the type II non-disjoint decomposition procedure. Since for the type II decomposition the extra variable a can't be shared between R_1 and R_2 then the only way to achieve non-disjoint decomposition is to share some of the independent variables from the set X .

The disjoint decomposition with $X_1 = \{x_1, x_2\}$ and $X_2 = \{x_3, x_4\}$ in Figure 5 leads to $|a| = 4$ and doesn't simplify the original structure. Selecting non-disjoint sets $X_1 = \{x_1, x_2\}$, $X_2 = \{x_2, x_3, x_4\}$ leads to the table in Figure 5c. Some of the cells in this table correspond to impossible combinations of variable values for instance, a variable taking values 0 and 1 at the same time. These cells are denoted by '-' and correspond structural zeroes as defined in [Krippendorff, 1986]. Since structural zeroes correspond to impossible observations we can replace them with any values for the sake of the column multiplicity computations. Selecting the values as in Figure 5d results in column multiplicity equal to 2. This value is smaller than the value of column multiplicity of the table in Figure 5b corresponding to the non-disjoint case. Relations R_1 and R_2 can now be determined the same way as for the disjoint case. The result of the decomposition is shown in Figures 5e, 5f, and 5g.

The same procedure can also be used in type I decomposition to increase the number of shared variables if needed.

4 COST MEASURE: CARDINALITY

The cost measure used in this paper is based on the measure proposed by Abu-Mostafa in [Abu-Mostafa, 1988]. He defined complexity of a binary function (functional block) as a number of tuples describing it:

$$C = 2^{|X|} |Y| \quad (1)$$

where X and Y are sets of independent and dependent variables respectively.

The cost of a combination of functional blocks was defined as a sum of costs of particular blocks.

Notice, that according to his definition the number of tuples, which is determined by the set of independent variables X , is multiplied by the number of dependent variables. This is due to the fact that each dependent variable corresponds to a separate function defined on the same set of independent variables.

We will extend Abu-Mostafa definition to the multiple-valued case and call it **cardinality** (see [Grygiel, 2000]):

$$C = \prod_{x_i \in X} x_i \cdot \log_2 \prod_{y_i \in Y} y_i \quad (2)$$

where $\prod_{x_i \in X} x_i$ is the number of tuples and $\log_2 \prod_{y_i \in Y} y_i$

is a normalized number of dependent variables, i.e. an equivalent number of binary variables corresponding to the set Y of multiple-valued variables.

If there is no dependent variables we will assume:

$$C = \prod_{x_i \in X} x_i \quad (3)$$

This is justified by the fact that every neutral relation (only independent variables present) can be always transformed to a function with one binary dependent variable which takes value 1 if the corresponding combination of values of independent variables belongs to the relation and takes value 0 if it doesn't. For one binary variable y the expression $\log_2 \prod_{y_i \in Y} y_i$ in Equation 2 is equal to 1 and the Equation 3 follows.

Let us consider decomposition of the block (X, Y) into blocks (X_1, Y_1) and $(Y_1 \cup X_2, Y)$. The complexity of the decomposed structure is equal to:

$$C = p_{X_1} \log_2 p_{Y_1} + p_{Y_1} p_{X_2} \log_2 p_Y \quad (4)$$

where: $X_1 \cup X_2 = X$

$$\begin{aligned} p_{X_1} &= \prod_{x_i \in X_1} x_i \\ p_{X_2} &= \prod_{x_i \in X_2} x_i \\ p_{Y_1} &= \prod_{y_i \in Y_1} y_i \\ p_Y &= \prod_{y_i \in Y} y_i \end{aligned}$$

Comparing Equation 4 to the complexity of the original structure we can easily show that in order to achieve complexity reduction the following necessary condition must be true:

$$p_{Y_1} < p_{X_1} \quad (5)$$

in fact, if $p_{Y_1} \geq p_{X_1}$ then:

$$\begin{aligned} p_{X_1} \log_2 p_{Y_1} + p_{Y_1} p_{X_2} \log_2 p_Y &\geq \\ p_{X_1} \log_2 p_{X_1} + p_{X_1} p_{X_2} \log_2 p_Y &\geq \\ p_{X_1} p_{X_2} \log_2 p_Y = p_X \log_2 p_Y & \end{aligned}$$

and decomposition increases, instead of decreasing, the complexity of the structure.

5 RESULTS

In this section we will present type II decomposition of a small real life example (Ries-Smith data [Ries and Smith, 1963]) and compare complexities of few simple decomposition examples presented in the previous sections.

Figure 6a depicts contingency table of the Ries-Smith data.

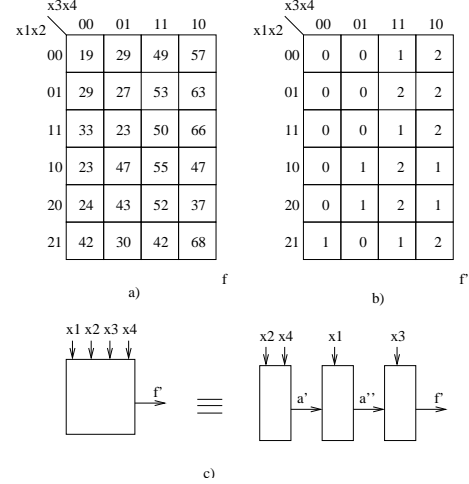


Figure 6: *Ries-Smith data*

We have four independent variables here, each combination of variable values is associated with a frequency in the table in Figure 6a. In Figure 6b the result of uniform binning into three equally sized bins is shown. We performed uniform binning for the number of bins ranging from 2 to 10 but only for the 3 bins case our program was able to find a decomposition.

The decomposed structure is portrayed in Figure 6c. In the decomposition process (disjoint decomposition of type II) three blocks were extracted from the original data and two new variables a' and a'' added. The tables in Figure 7 describe relations between variables in these three blocks.

x_2	x_4	a'
0	1	0
0	0	1
1	1	1
1	0	2

x_1	a'	a''
0	2	0
0	1	0
0	0	1
1	2	0
1	1	1
1	0	2
2	2	2
2	1	1
2	0	2

x_3	a''	f'
0	0	0
0	1	0
1	1	1
1	2	2
1	0	2
0	2	1

Figure 7: *Ries-Smith data: decomposed blocks.*

The complexity of original structure in Figure 6b is equal to $C = |x_1||x_2||x_3||x_4| \log_2 |f'| = 3 \cdot 2 \cdot 2 \cdot 2 \cdot \log_2 3 = 38.04$, the decomposed structure complexity is smaller and equal to $C_{II} = |x_2||x_4| \log_2 |a'| + |x_1||a'| \log_2 |a''| + |x_3||a''| \log_2 |f'| = 2 \cdot 2 \cdot \log_2 3 + 3 \cdot 3 \cdot \log_2 3 + 2 \cdot 3 \cdot \log_2 3 = 30.11$, which makes for 21% complexity reduction.

The Table 1 summarizes complexity gains for different structures discussed in this paper.

	Type		C'	C''	drop
Figure 2	I	d np	16.00	16.00	0%
Figure 3	I	d p	37.15	26.58	28%
Figure 4	II	d p	37.15	22.58	39%
Figure 5	II	nd p	37.15	26.58	28%
Figure 6	II	nd p	38.04	30.11	21%

Table 1: *Complexity drop.*

where d and nd denote disjoint and non-disjoint decompositions, p and np denote probabilistic and non-probabilistic relations, C' is the complexity of the initial structure, C'' is the complexity of the structure after decomposition and Equations 2, 3, and 4 were used to calculate complexity. Remember also that Figures 3, 4, and 5 present decomposition of the same relation but using different decomposition methods.

As we can see in Table 1 for the decomposition in Figure 2 complexity of the decomposed structure is the same as that of the initial structure. However, if we count the number of tuples in the original and decomposed structures we will obtain 10 and 9 respectively (10%). This means that the complexities calculated using Equations 2, 3, and 4 are equal to the maximum number of tuples that can be used to describe a given structure. The real complexity (number of tuples) can be in fact smaller. In other words, the values of C' and C'' in Table 1 compare the size of the state space of the original table (the binned table, if the data are probabilistic) and the sum of the sizes of the state spaces of the tables which the decomposition gives.

The second observation we can make is that non-disjoint decompositions usually result in higher complexity structure than the disjoint ones (see Figures 3 and 5 for non-disjoint and Figure 4 for disjoint decompositions). The disjoint decomposition however, is harder to find and the non-disjoint one may be the best we can get. The non-disjoint decomposition may not exist either but the chance of finding it is higher than for disjoint one.

6 SUMMARY

In this paper we presented two types of decomposition that can be used for decomposing relations both probabilistic and non-probabilistic. The decomposition of type I can be applied to relations directly and leads to non-disjoint decomposed structures. The type II decomposition is a functional decomposition so we apply it to probability density function or frequency distribution specified for a given relation.

Both decompositions act on discrete values only so if they are applied to probabilistic relations the continuous values of probabilities or frequencies have to be discretized before decomposition. All decompositions are “lossless” in the sense that they yield the binned table exactly (but they’re not lossless relative to original tables with unbinned frequencies). No analysis has yet been done of the loss of information which occurs when frequencies are binned.

The decomposition process is driven by a cost function which assures that the decomposed structure is of lower complexity than the decomposed one. The cost function used in this paper (cardinality) defines a relation’s complexity as number of its tuples. Other cost functions could be used as well (see for instance [Grygiel, 2000]) but more detailed discussion of this subject is beyond the scope of this paper.

We also presented few simple decomposition examples to illustrate the algorithms used. In one of the examples we decomposed a small real life data set (Ries-Smith data) running our software implementation of type II decomposition method. The data values were discretized using uniform binning method and decomposed iteratively into three blocks with two independent variables each.

Summarizing, we think that the methods of decomposition presented in this paper can serve as a useful alternative to the uncertainty based methods most often used for the decomposition of probabilistic relations.

References

- [Abu-Mostafa, 1988] Abu-Mostafa, Y. (1988). *Complexity in Information Theory*. Springer-Verlag, New York.
- [Ashby, 1965] Ashby, W. (1965). Measuring the internal informational exchange in a system. *Cybernetica*, 1(8):5–22.
- [Catlett, 1991] Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In Kodratoff, Y., editor, *Proc. of the European Working Session on Learning*, pages 164–178, Berlin, Germany. Springer-Verlag.
- [Conant, 1972] Conant, R. (1972). Detecting subsystems of a complex system. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 374–377.
- [Conant, 1981] Conant, R. (1981). Set-theoretic structure modeling. *International Journal of General Systems*, 7(38):93–107.
- [Fayyad and Irani, 1993] Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann.
- [Grygiel, 2000] Grygiel, S. (2000). *Decomposition of Relations as a new Approach to Constructive Induction in Machine Learning and Data Mining*. PhD dissertation, Portland State University, Portland, Oregon.
- [Grygiel and Perkowski, 1998] Grygiel, S. and Perkowski, M. (1998). New compact representation of multiple-valued functions, relations, and non-deterministic state machines. In *ICCD-98*, Austin, Texas.
- [Grygiel et al., 1997] Grygiel, S., Perkowski, M., Marek-Sadowska, M., Luba, T., and Jozwiak, L. (1997). Cube diagram bundles: a new representation of strongly unspecified multiple-valued functions and relations. In *Proc. of ISMVL'97*, pages 287–292, Halifax, Nova Scotia, Canada.
- [Hartley, 1928] Hartley, R. (1928). Transmission of information. *The Bell Systems Technical Journal*, 7(3).
- [Klir, 1976] Klir, G. (1976). Identification of generative structures in empirical data. *International Journal of General Systems*, (3):89–104.

- [Klir, 1985] Klir, G. (1985). *Architecture of Systems Problem Solving*. Plenum Press, New York.
- [Klir, 1996] Klir, G. (1996). Reconstructibility analysis bibliography. *Int. J. General Systems*, 24:225–229.
- [Krippendorff, 1979] Krippendorff, K. (1979). On the identification of structures in multivariate data by the spectral analysis of relations. In *Proc. 23th Annual Meeting of SGSiL*, Houston, Texas.
- [Krippendorff, 1986] Krippendorff, K. (1986). *Information Theory: Structural Models for Qualitative Data*. Sage Publications, Inc.
- [Ries and Smith, 1963] Ries, P. and Smith, H. (1963). The use of chi-square for preference testing in multidimensional problems. *Chem. Eng. Progress*, 59:39–43.
- [Shannon and Weaver, 1949] Shannon, C. and Weaver, W. (1975 (first published in 1949)). *The Mathematical Theory of Communication*. University of Illinois Press.
- [Zwick, 1995a] Zwick, M. (1995a). Control uniqueness in reconstructability analysis. *International Journal of General Systems*, 23(2).
- [Zwick, 1995b] Zwick, M. (1995b). Set-theoretic reconstructability of elementary cellular automata. *Advances in System Science and Application, Special Issue I*, pages 31–36.
- [Zwick, 2001] Zwick, M. (2001). Whole and parts in general systems methodology. In Wagner, G., editor, *The Character Concept in Evolutionary Biology*. Academic Press.