

2004

An Overview of Reconstructability Analysis

Martin Zwick

Portland State University, zwick@pdx.edu

Let us know how access to this document benefits you.

Follow this and additional works at: http://pdxscholar.library.pdx.edu/sysc_fac

 Part of the [Databases and Information Systems Commons](#), [Multivariate Analysis Commons](#), and the [Theory and Algorithms Commons](#)

Citation Details

Martin Zwick, (2004) "An overview of reconstructability analysis", *Kybernetes*, Vol. 33, No. 5/6, pp. 877 - 905

This Post-Print is brought to you for free and open access. It has been accepted for inclusion in Systems Science Faculty Publications and Presentations by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

AN OVERVIEW OF RECONSTRUCTABILITY ANALYSIS

Martin Zwick

Systems Science Ph.D. Program, Portland State University, Portland, OR 97207-0751
zwickm@pdx.edu http://www.sysc.pdx.edu/Faculty/Zwick

Keywords: reconstructability analysis, information theory, log-linear modeling, decomposition, general systems methodology, multi-valued logic, logic design, data mining, machine learning

ABSTRACT

This paper is an overview of reconstructability analysis (RA), an approach to discrete multivariate modeling developed in the systems community. RA includes set-theoretic modeling of relations and information-theoretic modeling of frequency and probability distributions. It thus encompasses both statistical and non-statistical problems. It overlaps with logic design and machine learning in engineering and with log-linear modeling in the social sciences. Its generality gives it considerable potential for knowledge representation and data mining.

I. INTRODUCTION

This paper is an overview of reconstructability analysis (RA), a discrete multivariate modeling methodology developed in the systems literature; an earlier version of this tutorial is (Zwick 2001). RA derives from Ashby (1964), and was developed by Broekstra, Cavallo, Cellier, Conant, Jones, Klir, Krippendorff, and others (Klir 1986, 1996). RA resembles and partially overlaps log-linear (LL) statistical methods used in the social sciences (Bishop et al 1978; Knoke & Burke 1980). RA also resembles and overlaps methods used in logic design and machine learning (LDL)

in electrical and computer engineering (e.g., Perkowski 1997). Applications of RA, like those of LL and LDL modeling, are diverse, including time-series analysis, classification, decomposition, compression, pattern recognition, prediction, control, and decision analysis.

RA involves the set-theoretic modeling of relations and mappings and the information-theoretic modeling of probability/frequency distributions. Its different uses can be categorized using the dimensions of variable, system, data, problem, and method-types shown in Table 1. These will now be briefly discussed. Section II explains RA in more detail. Section III gives examples, Section IV discusses software, and Section V offers a concluding discussion.

1. Variable-type: Nominal, ordinal, and quantitative

RA applies to multivariate data involving nominal variables or quantitative variables which are converted into nominal variables by being discretized. Variables need not be binary (dichotomous) but can be multi-valued. Nominal variables, whose states are discrete and unordered, are the most general type of variable, and so methods which apply to them encompass ordinal and quantitative variables as well. Continuous quantitative variables can be discretized either by quantization (non-overlapping binning intervals) or by fuzzification (Zadeh, 1965; Cellier et al, 1995), which is less sensitive to the boundaries of the bins. Although discretization loses information, this loss is offset by the fact that RA can detect nonlinearities and interaction effects which might be missed by standard methods. Moreover, it is not necessary to hypothesize *specific* nonlinear and interaction effects to detect their existence. The subject of discretization is mentioned only to emphasize the generality of nominal variable methods; it is outside the scope of this paper.

2. System-type: directed vs. neutral

To relate RA to a familiar LDL problem, consider the task of decomposing a logic function $Z=g(A, B, C)$, where variables are either binary or multivalued. In RA terminology this is a *directed* system, since inputs and outputs ("independent variables" and "dependent variables")

Table 1. Aspects of RA. (Prototypical RA task shown in bold.)

1. VARIABLE TYPE	nominal (discrete) [binary/ multi-valued] ordinal (discrete) quantitative (typically continuous)
2. SYSTEM TYPE	directed system (has inputs & outputs) deterministic vs. non-deterministic neutral system (no input/output distinction)
3. DATA TYPE	information theoretic RA (freq./prob. distribution) set-theoretic RA (set-theoretic relation/function)
4. PROBLEM TYPE	reconstruction (decomposition) confirmatory vs. exploratory (data analysis/ mining) identification (composition)
5. METHOD TYPE	variable-based modeling (VBM) state-based modeling (SBM) latent variable-based modeling (LVBM)

are distinguished. Directed systems are further classified as deterministic or stochastic. While most RA applications involve predictive, dynamic, or causal (hence directed) relationships between variables, sometimes variables have equal status; these systems are called *neutral*.

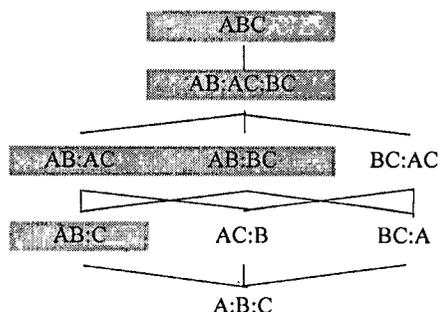
RA can be applied to directed -- both deterministic and stochastic -- and neutral systems. By contrast, in LL modeling, stochastic systems are usually the focus. In LDL modeling, deterministic directed systems are the rule, and neutral systems are rarely considered.

3. Data-type: Information-theoretic and set-theoretic RA

RA has two versions: a set-theoretic (here called SRA), or more precisely, a "crisp possibilistic," version which applies to set-theoretic relations and mappings, and an information-theoretic (here called IRA) "probabilistic" version which applies to frequency (and probability) distributions (Conant, 1981; Klir, 1985; Krippendorff, 1986). IRA can also be applied to quantitative functions of nominal variables by rescaling these functions so that they can be treated as probability distributions (Jones, 1985a). SRA and IRA are similar in many respects, and together constitute a coherent framework. Moreover, probabilistic and crisp possibilistic analyses are encompassed within a "generalized information theory" (Klir & Wierman, 1998), which includes also fuzzy possibilistic and probabilistic distributions.

The same model structures are considered in both IRA and SRA. Let ABC represent a set-theoretic relation or mapping or a probability or frequency distribution for a three-variable system, with projections AB, AC, and BC, and A, B, and C. Define a structure as a non-redundant set of projections. If ABC is the data, the possible model structures shown in Figure 1. At the top of the lattice is the data, also called the "saturated model." At the bottom is A:B:C, called the "independence model." (In IRA, the bottom model may alternatively chosen to be the uniform distribution.)

Figure 1. Lattice of Specific Structures for a 3-variable neutral system. The shaded sublattice is for a directed system, with inputs A & B and output C.



The figure shows that the lattice of structures for a directed three variable system (with two inputs and one output) is a sublattice of the lattice for neutral systems. For directed systems with output C, the independence model is AB:C, not A:B:C, and only the 5 shaded structures need to be considered. Each of these 5 structures contains an AB component (relation or distribution). A directed system model always has one component which collects together all inputs, allowing for but ignoring the possible presence of constraint among them. Every other component includes at least one output. Directed system models can thus be characterized by their number of predicting components. For inputs A, B, and output C, model AB:AC has one predicting component, AC. (We do not here employ a notation which explicitly shows directedness, e.g., $A \rightarrow C$, any relation or distribution written as a string of letters may be either neutral or directed.) Model AB:AC:BC has two predicting components, which are "independent" in a "maximum uncertainty" sense, to be described later. Only in model ABC do A and B interact in their joint effect on C.

The most commonly used version of RA is IRA. Here the problem is typically the decomposition of frequency or probability distributions, where RA does statistical analysis. This is the main subject of this paper. Consider a frequency distribution $f(A, B, C, Z)$ for a directed system, where A, B, and C are inputs and Z is an output. RA decomposes such distributions into projections, such as $f_1(A, B, Z)$ and $f_2(B, C, Z)$, and models are assessed for statistical significance, usually with the Chi-square distribution. This use of RA overlaps considerably with LL modeling but has no parallel in LDL, where "statistical" considerations can arise in that functions or relations may be *partially specified*, e.g., due to sparse sampling.

Where IRA and LL overlap, they are equivalent, but each has distinctive strengths. The LL literature is more advanced statistically, includes latent variable techniques (discussed below), and offers methods to analyze ordinal variables. Well tested LL software exists (e.g., in SPSS and SAS). On the other hand, in IRA, graph-theoretic methods are used to define explicitly various lattices of possible models and to suggest heuristic techniques to search these lattices. RA makes extensive use of the uncertainty (Shannon entropy) measure, which is conceptually transparent because of its similarity to variance, and includes innovations like state-based modeling (discussed below), absent in the LL literature. However, the RA and LL communities have been only dimly aware of -- and have not benefitted much from -- each other's existence, despite early work which linked the two (Kullback, 1959; Ku & Kullback, 1968).

While IRA is statistical in its overlap with LL, it includes also non-statistical applications. For example, the k-systems methodology of Jones (1985a) is used primarily for function approximation and compression. IRA can also be used to analyze set-theoretic relations and functions (this is done, for example, in the analyses of cellular automata reported below). SRA, in contrast, is completely and inherently non-statistical. It is the natural RA approach to set-theoretic relations and functions. SRA here overlaps with LDL. While it appears to be different from any particular LDL technique, it resembles LDL methods which decompose functions into *generalized* (arbitrary) as opposed to *specific* components (like 'and' and 'or' gates).

RA thus bridges two very disparate fields: log linear modeling in the social sciences and logic design and machine learning in electrical and computer engineering..

4. Problem-type: Reconstruction vs. identification

RA includes *reconstruction* and *identification* (Klir, 1985). In reconstruction ABC (see Figure 1) is the data, and one goes down the lattice until decomposition losses are unacceptable. Or, one can start at the bottom with A:B:C (the independence model for neutral systems) or AB:C (the independence model for a directed system with output C), and ascend until model error relative to the data is too great or the model is unacceptably complex. Descending the lattice is especially natural for neutral systems, while ascending the lattice is more natural for directed systems.

Thus, in reconstruction, a distribution or relation is decomposed (compressed, simplified) into projected distributions (also called "margins") or relations. ABC might, for example, be decomposed into AB and BC, written as structure, AB:BC. Taken together, the two linked bivariate projections would constitute a model of the data which is less complex (has fewer degrees of freedom) than the data. By maximum-entropy (uncertainty) composition of these projections, the model yields a *calculated* trivariate $ABC_{AB:BC}$ distribution or relation which may differ from the observed ABC data. The difference (error) represents loss of information in the model. By definition, the data itself has 100% information (0% error). Models are also assessed in complexity, where complexity for IRA is df, degrees of freedom, the number of parameters needed to specify a model. For convenience df values may be normalized so that the data is 100% complex and the independence model 0% complex. Reconstruction decomposes data by finding less complex models which preserve either all of its information (lossless decomposition) or a sufficient amount of its information (lossy decomposition), where sufficiency is assessed either statistically or by other standards.

Reconstruction is done in either a confirmatory or exploratory mode. In the confirmatory mode for IRA, a specific model or a small number of models, proposed *a priori* on the basis of theoretical considerations, are tested statistically. In the exploratory mode, one has no prior idea about what model might be suitable, and one examines many structures to find a best model or a family of best models. LL modeling is normally done in the confirmatory mode and, indeed, in the social sciences, exploratory modeling is normally frowned upon. The situation is quite different in machine learning, a field explicitly devoted to exploratory modeling.

Identification is pure composition. For example, the observed data might be the two distributions AB and BC. Because they are not derived from a single ABC, AB and BC can be inconsistent if they disagree in their common B projection. If such an inconsistency can be resolved, a *calculated* ABC can be generated. Identification methods exist which resolve such inconsistencies and make possible the integration of multiple data sets (as in a data-base merge) coming from different sources. The LL and LDL literatures have not articulated the identification problem and are focused exclusively on reconstruction.

Reconstruction in the exploratory mode is the typical RA problem. For convenience, call modeling with "few" variables *data analysis* and modeling with "many" variables *data mining*. This paper discusses both, but is motivated by the task of data mining, for which adequate RA software is not yet available. (LDL techniques have been used for data mining, but LL methods, not typically implemented for many variables or for exploratory searching, are rarely mentioned in the data-mining literature.) "Few" variables here means that *exhaustive* evaluation of all possible RA models can be done. This allows us to be certain of the choice of a best single model or it might be done for a very different purpose, namely to characterize the data by the set of errors for all possible decompositions. The limit of exhaustive search is roughly about 7 or 8 variables, in round numbers, 10. *Data mining* here means exploratory modeling beyond this threshold, i.e., with 10's, 100's, perhaps even 1,000's of variables. Exhaustive search is then no longer possible, and *heuristic* techniques, which consider only a subset of possible models, must be used instead.

This threshold of somewhat less than 10 variables marks the limit of exhaustive analysis of all models, but there is a second threshold involving the number of *possible* states of the system (as opposed to the number of states observed in the data), which poses limits even for heuristic search. This 2nd threshold presently precludes the use of *multi-predictive*-component RA models for more than about 20 variables, but *single-predictive* component models can still

be used. For single-predicting-component models, one can treat 10's, 100's, and perhaps even 1000's of variables, with computing time and space requirements depending on the size of the data, not the size of the state space. However, multi-predicting-component models for directed systems are cyclic (have "loops"), and at present cyclic models in IRA need computation on the entire state space, without regard to how sparse the data is. For 20 binary variables, the state space is about 24 Mbytes. Adding many variables beyond 20 is impractical, though *approximate* RA computation, which varies with the data and not the state space, might extend this range. Two possible approximation approaches are mentioned at the end of this paper.

5. Method-type: Variable-based, state-based, and latent variable-based RA

Reconstruction as explained above illustrates *variable*-based modeling (VBM), which decomposes data into subsets of variables. This is the most common situation. Two other method-types are available: state-based modeling (SBM) and latent variable-based modeling (LVBM).

SBM is less developed than LVBM. Originally an aspect of the "k-systems analysis" of Jones (1985a,b, 1986), SBM is now being more integrated into the standard RA framework (Johnson & Zwrick 2000; Zwrick & Johnson 2003). VBM reveals information-rich sets of *variables*, and a variable-based model is a set of *complete* projections. By contrast, SBM selects information-rich *states*, i.e., salient conditions (e.g., Shaffer, 1987, 1988). A state-based model is a set of frequency values selected from the original data and its projections, but complete projections do not have to be included. For example, for f(A,B,C) data, the SB model could be the frequencies, f(A₂,B₁,C₃), f(A₁,B₂), and f(C₂). SBM resembles rule-based methods in logic programming and fuzzy control. It also resembles Crutchfield's e-machines (Feldman & Crutchfield, 1998).

Even though log-linear methods overlap with IRA, there is nothing equivalent to SBM in the LL literature. Something like SBM seems standard in LDL, where decompositions involving sums of products having varying numbers of variables are widely used.

In latent variable models complexity is reduced or

new constructs are introduced by adding additional, unmeasured, variables. For example, an AB distribution might be modeled by the simpler AQ and QB projections of an AQ:QB model. LV models are absent in the RA literature but widely used in the log-linear field (Hagenaars, 1993; Vermunt, 1997). However, latent variable LL software which is usable for exploratory data mining is not available. In LDL, latent variables are standard, functions being typically decomposed using both "free" (observed) and "bound" (latent) variables (Grygiel, 2000; Grygiel, Zwrick, & Perkowski, 2003).

Since the objective of this paper is to explain RA (especially IRA) methodology, no survey is offered of RA applications, and comparisons with other methods are not undertaken. Section II provides details mostly on IRA reconstruction. Section III gives a few applications of RA to data analysis and mining. Section IV is a summary and discussion.

II. More Detailed Explanation

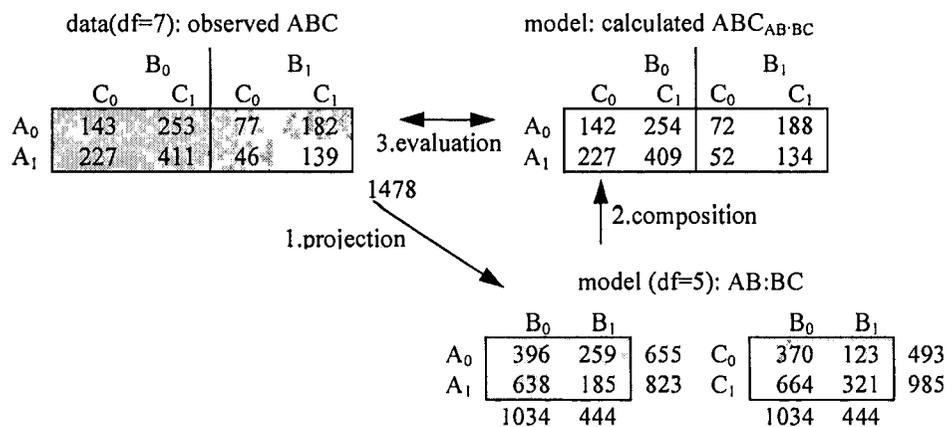
The "prototypical" RA analysis is IRA variable-based reconstruction, which will now be explained. Brief explanations will be provided for SRA, identification, and LVBM and SBM.

1. Information-theoretic variable-based reconstruction (& identification)

Basic reconstruction steps

Table 2 illustrates variable-based information-theoretic reconstruction. The data, ABC, is shown on the left. The best RA model, AB:BC, judged by its information and complexity, is shown on the lower right; its calculated frequencies are on the upper right. Reconstruction is done in 3 steps: (1) projection, (2) composition, and (3) evaluation.

Table 2. Example of IRA variable-based reconstruction of a neutral system (the number of shaded cells is the degrees of freedom of each model)



Projection

The ABC data is projected into the two contingency tables, AB and BC which define the model AB:BC (lower right). This step is straightforward, and is given by $f(A_0, B_1) = f(A_0, B_1, C_0) + f(A_0, B_1, C_1)$.

Composition

These two tables together yield the *calculated* ABC_{AB:BC} table (upper right), where frequencies are rounded to the nearest integer.

Shift from frequencies to probabilities (frequencies divided by the sample size). Let p and q denote observed and calculated probabilities, respectively. The IRA composition step is a "maximum entropy" procedure (Miller & Madow, 1954; Good, 1963), where entropy is Shannon entropy, referred to here as uncertainty. For model AB:BC, the *calculated* probability distribution, q_{AB:BC}(A,B,C), is the distribution which maximizes the uncertainty,

$$U(AB:BC) = -\sum \sum \sum q_{AB:BC}(A,B,C) \log q_{AB:BC}(A,B,C)$$

subject to the AB and BC projections of the data, i.e., to linear constraints q_{AB:BC}(A,B) = p(A,B) and q_{AB:BC}(B,C) = p(B,C). Since AB:BC has no cycles, the solution can be written algebraically:

$$q_{AB:BC}(A,B,C) = p(A,B) p(B,C) / p(B).$$

Calculated distributions for *cyclic* structures, like AB:BC:AC, need to be evaluated iteratively by the Iterative Proportional Fitting (IPF) algorithm. As shown in Table 3, calculation of q_{AB:BC:AC}(A,B,C) starts with a uniform distribution. IPF then imposes upon it iteratively the observed projections specified by the model. At iteration #1, first AB is imposed, then BC is imposed, and then AC is imposed. At iteration #2, AB is reimposed because agreement with AB was destroyed when the other projections were imposed at the previous iteration. Then BC is reimposed, then AC. And so on. IPF iterates until q_{AB:BC:AC}(A,B,C) converges.

Table 3. Iterative Proportional Fitting for model AB:BC:AC

The algorithm starts with

$q^{(0)}_{AB:BC:AC}(A,B,C) = 1 / (|A| |B| |C|)$, where || means cardinality and then loops over iterations from j=0 until convergence:

For all A,B	$q^{(j+1)}_{AB:BC:AC}(A,B,C) = q^{(j+0)}_{AB:BC:AC}(A,B,C)$	$p(A,B) / q^{(3j+0)}_{AB:BC:AC}(A,B)$
For all B,C	$q^{(j+2)}_{AB:BC:AC}(A,B,C) = q^{(j+1)}_{AB:BC:AC}(A,B,C)$	$p(B,C) / q^{(3j+1)}_{AB:BC:AC}(B,C)$
For all A,C	$q^{(j+3)}_{AB:BC:AC}(A,B,C) = q^{(j+2)}_{AB:BC:AC}(A,B,C)$	$p(A,C) / q^{(3j+2)}_{AB:BC:AC}(A,C)$

j=j+1

The IPF algorithm requires that the entire distribution be computed, not merely the set of observed states, because as each projection is imposed on the working q_{AB:BC:AC}(A,B,C), non-observed states will in general contribute to other calculated projections. Thus computer time and space requirements for cyclic models vary with the state space of the problem, and not with the sample size.

In log-linear modeling, calculations often go beyond the generation of q_{model}. The individual frequencies of specific states, e.g., f_{AB:BC}(A₀,B₁,C₀), can be decomposed into the contributions from all the separate "effects," i.e., A, B, C, AB, and BC. This can be useful if one is particularly interested in one or a few states. Such decomposition is not prominent (perhaps even absent) in the RA literature.

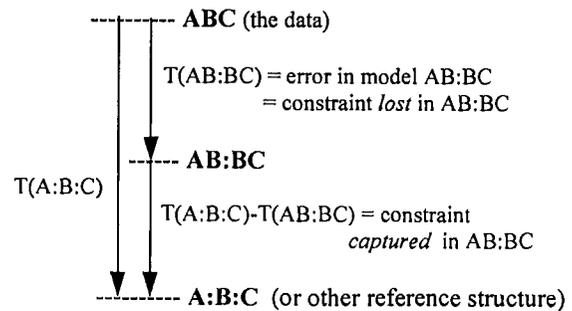
Evaluation

The calculated ABC_{AB:BC} is compared to the observed ABC. The calculated distribution approximates the data, which is always more constrained. The error is the constraint *lost* in the model (Figure 2), called transmission T(AB:BC), which is the difference between the uncertainty of the model and the uncertainty of the data.

$$T(AB:BC) = \sum \sum \sum p(A,B,C) \log [p(A,B,C) / q_{AB:BC}(A,B,C)]$$

$$= U(AB:BC) - U(ABC)$$

Figure 2. Constraint lost and retained in models



Information, i.e., constraint captured in the model, is normalized to [0,1] with respect to the independence model, A:B:C (or to AB:C, for directed systems where C is the output), as follows:

$$\text{Information} = 1 - [T(AB:BC) / T(A:B:C)]$$

In addition to model error, model complexity is also of interest. It is desirable to minimize both, but there is a tradeoff between the two. Decisions on model acceptance are made either by optimizing one subject to the other as a constraint, or by merging the two via Chi-square or other approaches.

Complexity of the model is defined as its degrees of freedom, *df*, the number of parameters needed to specify it. Reconstruction is compression; it reduces complexity. For the data and model, *df* is shown in Table 2 by the count of shaded cells. Knowing the sample size subtracts 1 from the number of cells (states) in a table; thus for data ABC (the "saturated model"), *df* = 7. In AB:BC, *df*(AB) = 3 but once AB is specified, only 2 more numbers are needed to specify BC, because the B margins of the tables must agree. Algebraically, *df*(AB:BC) = *df*(AB) + *df*(BC) - *df*(B) = 3+3-1 = 5. Normalizing *df* to the [0,1] ranges gives a normalized complexity,

$$\text{Complexity}(AB:BC) = \frac{[df(AB:BC) - df(A:B:C)]}{[df(ABC) - df(A:B:C)]}$$

In *confirmatory* analysis, AB:BC might be a hypothesized model. Its error would be assessed by calculating the Likelihood-ratio Chi-square, $L^2(AB:BC) = 1.3863 N T(AB:BC)$, where *N* is sample size. $L^2(AB:BC)$ and *df*(AB:BC) are then used to obtain α , the probability of making a Type-I error by rejecting the null hypothesis that the calculated $ABC_{AB:BC}$ is statistically the same as the observed ABC. There are also other ways of integrating model error (or information) and complexity to decide on model acceptability.

Exhaustive analysis

Table 2 illustrated *confirmatory* RA, where one model is assessed. The exhaustive evaluation of all models for this data is given in Table 4 which gives (information, α , *df*) for every model. This complete RA characterizes the data more fully than merely stating that the best model is AB:BC. ABC has 100% information. The probability of error in rejecting its agreement with the data is 1, since it is the data. A:B:C is the baseline for analysis, and thus has no

Table 4. IRA results for the data of Table 2. (Information, α , *df*) for ABC_{model} . The reference model for calculation of α is ABC. Shading shows models to be considered if the system were directed, with C being the dependent variable.

	ABC (1., 1., 7)	
	AB:AC:BC (.987, .382, 6)	
AB:AC (.827, .005, 5)	AB:BC (.978, .518, 5)	BC:AC (.153, .000, 5)
AB:C (.826, .014, 4)	AC:B (.000, .000, 4)	BC:A (.152, .000, 4)
	A:B:C (0., .000, 3)	

information. The probability of error in rejecting its agreement with the data in this case is 0. A good model has high information and low *df*. If models are compared to the data (as is done in Table 4), a good model has *high* α , because the probability of error in rejecting the equivalence of model and data *should* be high. The more familiar preference for low α holds if the model is compared not to ABC but to A:B:C.

If the Table 2 data were for a directed system, with output C and inputs A and B, only the *shaded* models in Table 4 need be considered. For directed systems it is useful to state results in terms of reductions in the output uncertainty, knowing the inputs, rather than in terms of information. If the uncertainty reduction, $\Delta U(C|B) \equiv U(C) - U(C|B) = T(B:C) = U(C) - U(B,C) + U(B)$ is positive and statistically significant, model AB:BC is acceptable, i.e., B is a predictor of C.

Because Shannon entropy involves a log term, even small ΔU may indicate high predictability. For example, if the odds of rain vs. no rain is 2:1 in winter and 1:2 in summer and 1:1 over the year (for equal seasons), then knowing the season makes a big change in the odds, but reduces $U(\text{rain vs. no-rain})$ by only 8%. For the Table 2 data, the results are shown in Table 5. For this data, A and B reduce the uncertainty of C by only very small amounts, less than 1%. The last column of the table also shows that quantitative RA measures (such as *U* or *T*) can be thought of equivalently as assessing *models*.

Table 5. Uncertainty reductions (output C; inputs A & B)

	% ΔU	Δdf	α	Associated model
$U(C A,B)$	0.56	3	.013	ABC
$U(C A)$	0.52	2	.007	AB:AC:BC
$U(C B)$	0.49	1	.002	AB:AC
$U(C)$	0.00	1	.866	AB:BC
$U(C)$	-	-	1.000	AB:C

Note that there are two ways that A and B can both predict C. In AB:AC:BC they do so separately, but in ABC there is an *interaction effect* (Zwick, 1996). Here α is computed relative to the independence model, not the saturated (top) model; it is small if results are statistically significant. The table shows that A is a better predictor of C than is B; and that $\Delta U(C)$ using both inputs, either in ABC or in AB:AC:BC, is very small but still statistically significant at the 0.05 level.

Identification

Table 2 can be used also to explain *identification*. If the data is not the ABC on the

left but the two distributions on the lower right, identification derives the *calculated* $ABC_{AB:BC}$ on the upper right. This is trivial when AB and BC are derived by projection from a single ABC distribution. Suppose, however, that AB and BC came from *different* sources, and that the BC table had a B margin of [1024, 454] while the AB table still had the B margin of [1034, 444]. AB and BC would be inconsistent in their common B margin; this could arise from sampling or other errors. When data sets are obtained from different sources inconsistency is the norm, and resolution of inconsistency is required before composition can be done (Klir, 1985).

Mariano (1984, 1987), Pittarelli (1990) and Anderson (1996) have shown how inconsistencies among distributed data sets may be resolved. Anderson also showed how "nuisance" variables can be exploited in the composition of disparate data sets. To find a relation between A and C from inconsistent AB and BC distributions, the inconsistencies are first resolved, after which the adjusted AB' and BC' distributions are composed into ABC and projected onto AC.

2. Information-theoretic state-based & latent variable-based reconstruction

Variable-based modeling (VBM) requires the *complete* specification of its components (e.g., the full AB and BC tables in Table 2). It is possible, alternatively, to define a model in terms of the probabilities of *any set* of non-redundant *states* selected from ABC and its projections (Jones, 1985a,b). Thus SBM encompasses VBM as a special case. SBM selects states with unexpectedly high or low probabilities (relative to some prior); these are the facts prominent in the data, and $q_{model}(A,B,C)$ has maximum uncertainty, constrained by these values. This approach is more powerful than VBM, and Chen (1997) has discussed its possible use for data mining. The penalty is that there are many more SB models, so SBM might be most useful for data mining if it were used as a follow-on to VBM. The essence of SBM is illustrated in Table 6.

Table 6. State-based decomposition

	B ₀	B ₁			
A ₀	.1	.1	.2	.04	.16
A ₁	.1	.7	.8	.16	.64
	.2	.8		.2	.8
model	AB		A:B		[A ₁ ,B ₁]
df	3		2		1
error, T	-		.087		0

The AB model has df=3, shown by the shading of 3 cells (arbitrarily chosen). The A:B model has df=2, i.e., it needs

two specified probability values, one (arbitrarily chosen and shaded) from each margin. A:B, with probabilities $p(A)p(B)$, is not identical to AB and exhibits error. A state-based model specifying the *single* probability value, $p(A_1,B_1)=.7$ (shown shaded) forces the remaining (A_0,B_0) , (A_0,B_1) , and (A_1,B_0) probabilities, by the maximum uncertainty principle, to be .1. These values are correct and this model has zero error even though it is simpler than A:B.

This example was constructed to show that a one-parameter SB model can be superior to a two-parameter VB model. SB analysis of the distribution of Table 2 reveals that a 4-parameter model, $p(A_1,B_0)$, $p(A_0,B_1)$, $p(B_0,C_1)$, and $p(B_1,C_0)$ captures virtually the same amount of information as the 5-parameter AB:BC model (note that the four states come from AB and BC). In more complex distributions the economy of SBM is more dramatic.

Latent variable modeling (LVBM) is not developed within RA, though it is in the log-linear literature. It is also widely used for set-theoretic mappings in the LDL literature. It will be discussed here only briefly.

To illustrate latent variable reconstruction, consider BA:BC, where B is an input and A and C are outputs. By attributing directionality to the relations in the structure, we model $A \leftarrow B \rightarrow C$. If variables were quantitative instead of nominal and relations were linear, this would be "factor analysis" (Kim & Mueller, 1978) and B would be a common factor, which explains (away) the relation between A and C. B may be a new construct, implicit in AC, which LVBM makes explicit.

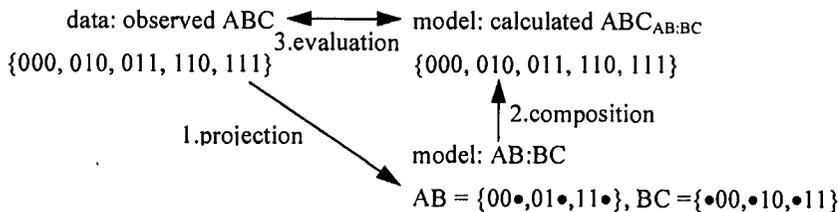
If B mediates between A and C, this is "path analysis" (Davis, 1985). The structure might be written AB:BC to suggest the causal sequence $A \rightarrow B \rightarrow C$. What factor and path analyses are for quantitative variables "latent class analysis" is for nominal variables (McCutcheon, 1987). In latent class analysis, a distribution AC might be explained by invoking a *latent variable* B and a calculated distribution ABC which is decomposable into BA:BC (equivalently, AB:BC). The generation of the calculated ABC is not as straightforward as it is in VBM, and a different algorithm is used in place of IPF (Hagenaars, 1993).

IRA or log-linear modeling with latent variables thus offers a nominal variable *generalization* of path analysis, factor analysis, and covariance structure modeling (Long, 1983), which are restricted in application only to linear relations among quantitative variables. Latent variables can also be used for compression. If given AC, a latent variable B yields a calculated ABC with structure BA:BC, and if $|A|=|C|=4$ and $|B|=2$, then $df(AC) = 15$, while $df(BA:BC) = 13$, so BA:BC is less complex than AC.

3. Set-theoretic reconstruction

In SRA, data is a set-theoretic relation -- a subset of a Cartesian product -- the set of states actually observed, without regard to frequency. Set-theoretic reconstruction is similar to information-theoretic reconstruction, as can be seen by comparing Table 2 to Table 7. In Table 7, the two values of A, B, and C are indicated as 0 and 1, but these are only arbitrary labels.

Table 7. SRA variable-based reconstruction (neutral system)



As in IRA, $ABC_{AB:BC}$ is the maximum uncertainty relation, given the model constraints, i.e., the model-specified projections. $U(ABC_{AB:BC})$ is now the Hartley (rather than the Shannon) entropy, $U(R) = \log_2 |R|$, where $| |$ is cardinality. This maximization is achieved in composition by (1) expanding the projections by Cartesian products with variables omitted in them (maximizing uncertainty), and (2) taking the intersection of the expanded relations (imposing the model constraints). Specifically, $ABC_{AB:BC} = (AB \otimes C) \cap (BC \otimes A)$, namely,

$$\begin{aligned}
 ABC_{AB:BC} &= [\{00\bullet, 01\bullet, 11\bullet\} \otimes \{\bullet\bullet 0, \bullet\bullet 1\}] \\
 &\quad \cap [\{\bullet 00, \bullet 10, \bullet 11\} \otimes \{0\bullet\bullet, 1\bullet\bullet\}] \\
 &= \{00\bullet, 01\bullet, 11\bullet\} \cap \{\bullet 00, \bullet 10, \bullet 11\} \\
 &= \{000, 010, 011, 110, 111\} = ABC
 \end{aligned}$$

(• is a place-holder for absent variable(s); * means “don’t care”). The second line of this equation shows that SRA resembles a product of sums of products. Generalizing the above equation, for model $P_1:P_2:\dots:P_n$, where P_j is a projection of relation R (the data), and where M_j is the Cartesian product of variables absent (projected on) in P_j , the calculated reconstructed R is given by

$$R_{P_1:P_2:\dots:P_n} = (P_1 \otimes M_1) \cap (P_2 \otimes M_2) \cap \dots \cap (P_n \otimes M_n)$$

Relations for cyclic models are calculated in the same way; no iterative procedure is needed. In the present case, $ABC_{AB:BC}$ has no error. Imperfect decomposition can, however, be allowed, and again transmission (error) is defined as $T(AB:BC) = U(ABC_{AB:BC}) - U(ABC)$. No model simpler than AB:BC agrees fully with the data, as shown below in Table 8.

Table 8. SRA results for the data of Table 7. in parentheses: (number of tuples in model)

	ABC (5)	
	AB:AC:BC (5)	
AB:AC (6)	AB:BC (5)	BC:AC (6)
AB:C (6)	AC:B (8)	BC:A (6)
	A:B:C (8)	

Note that SRA decomposition of ABC here uses *complete* projections and only *observed* variables. No latent (bound) variables are introduced as is commonly done in LDL techniques. The above relation could have been decomposed much more simply as $\{000, *1*\}$ or logically as $(A' \wedge B' \wedge C') \vee B$, where prime means negation. Such a

decomposition might be considered a “state based RA model,” in that specific states of subsets of variables are selected, were it not for the fact that in SRA SB models, states are always connected with an \wedge , as indicated in the intersect operations in the equation for $R_{P_1 P_2 \dots P_n}$.

The precise relationship between SRA and LDL methods has not yet been fully elucidated. Of particular interest would be a comparison of SRA to LDL multi-valued functional decomposition (Perkowski et al, 1997; Files & Perkowski, 1998). An initial examination of an *enhanced* version of SRA shows that it is superior for 3-variable binary functions to Ashenhurst-Curtis decomposition, a well-known and widely used LDL technique which utilizes latent variables (Al-Rabadi, Zwick, & Perkowski, 2003; Al-Rabadi & Zwick, 2003).

IRA problems can be approximated by SRA. In Table 2, if frequency is discretized into ranges (Chen, 1994; Grygiel, 2000; Grygiel, Zwick, & Perkowski, 2003), it becomes a nominal variable and the mapping $A \otimes B \otimes C \rightarrow F$ can be decomposed by SRA. In doing so, the different frequency bins become nominal states and their order cannot be exploited. This also precludes the assessment of statistical significance. Discretization of frequencies is qualitatively different from discretization of variables, and the error consequences of this approach, in comparison to IRA/LL statistical analyses, is under investigation.

Conversely, SRA problems can be treated by IRA by giving equal probability to observed tuples; IRA then generates results equivalent to SRA but in a different form. Just as a Fourier transform of a function may illuminate particular properties of the function previously obscure, even though the transform is fundamentally equivalent to the function, so

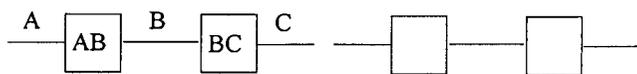
IRA computation may be informative in ways not directly achievable by set-theoretic analysis (Zwick & Shu, 2003).

4. Evaluating many models

In Figure 1, AB:AC, AB:BC, and AC:BC permute the three variables, as do AB:C, AC:B, and BC:A. There are 5 different *general* structures and 9 *specific* structures, where specific structures exemplifying the same general structure merely permute the variables. (Note: this nomenclature is not standard in RA.) For example, AB:BC and AC:CB are the same general structure. Structures are disjoint (AB:C), acyclic (AB:BC), or cyclic (AB:AC:BC).

A specific (acyclic) structure is exemplified in Figure 3, along with the general structure obtained from it by omitting the labels for specific variables (lines) and relations (boxes).

Figure 3. Specific structure AB:BC & general structure



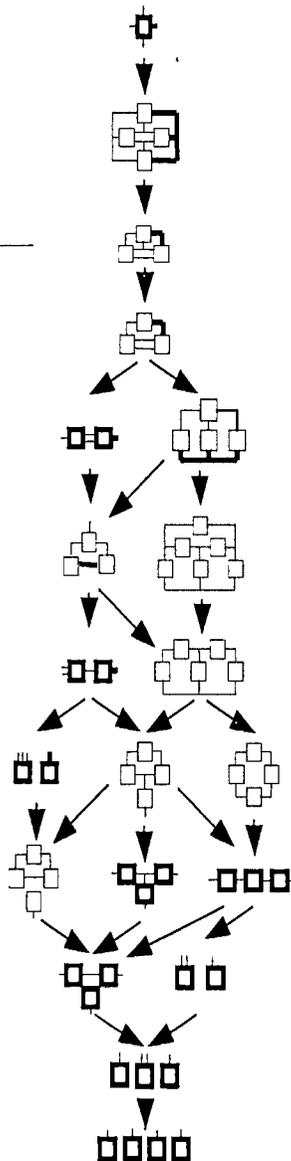
For four variables, there are 20 general structures and 114 specific structures. As the number of variables increases, the numbers of general and specific structures increase sharply (Table 9).

Table 9. Numbers of structures. The bottom line is for directed systems.

# variables	3	4	5	6
# general structures	5	20	180	16,143
# specific structures	9	114	6,894	7,785,062
# specific structures, 1 output	5	19	167	7,580

The table indicates that directed system lattices are simpler than neutral system lattices because we are not interested in the presence or absence of relations among the inputs. (Recall the shaded portions of Figure 1 and Table 4.) Table 9 also shows that exhaustive evaluation of all models ceases to be practical around 7 or 8 variables. Intelligent heuristics and sophisticated search techniques are required to sample the Lattice of Structures efficiently, and Klir (1985), Krippendorff (1986), Conant (1988), and others have made suggestions along these lines. The lattice can be pruned as a search procedure descends or ascends so that consideration is restricted only to promising candidates. Or, the search can be done roughly between groups of structures and then finely within these groups (Klir, 1985). The combinatorial explosion in models can also be mitigated by aggregating variables, but this poses difficulties of interpretation. It also does not reduce the number of possible states (the size of the contingency table) for the system, unless state aggregation,

Figure 4. Lattice of General Structures (4-variables). A box is a relation; a line, with branches, uninterrupted by a box, is a variable. Arrows indicate decomposition. The top structure is ABCD; the bottom is A:B:C:D. The 10 acyclic structures have relations in bold. For 1 output, D, and 3 inputs, D is in bold for the 9 structures in the simplified lattice.



another compression technique, also accompanies variable aggregation.

Figure 4 shows the lattice of general structures for a four-variable system. If variables are all dichotomous, the degrees of freedom of the structures range from 15 at the top (ABCD) to 4 at the bottom (A:B:C:D) and decrease by 1 at every level. The figure also shows the acyclic structures, indicated with boxes in bold (10 of the 20). The simplified lattice which obtains if D is an output and A,B,C are inputs has 9 structures whose bottom is ABC:D; these are indicated by structures where one line (representing D) is bold. In this simplified lattice, 4 of the structures are also acyclic. These correspond to single-predicting-component models in which the predicting component has 0, 1, 2, or 3 predictors of D.

Model complexity for IRA was defined above as the number of parameters needed to specify the model. The figure suggests another *type* of complexity, namely the number of components in the model. For three variables, this is at most 3, for four variables at most 6; in the medical sociology example discussed below it reaches 12. Multi-component models pose a challenge to interpretation. Beyond the mere number of components, there is also the complexity inherent in the connectedness of the components.

III. Examples of Variable-based Reconstruction

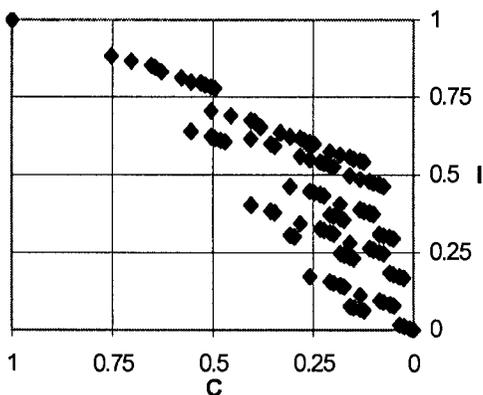
This section illustrates the exhaustive consideration of models in data analysis, and the heuristic use of RA in many variable data mining. To reiterate, “many” variables means that all possible models cannot be examined by brute force. The examples are restricted to information theoretic variable-based reconstruction.

1. IRA Example

Exhaustive Analysis (4 variables)

Figure shows a exhaustive analysis of a small subset of the OPUS data obtained from Dr. Clyde Pope of the Kaiser Permanente Center for Health Research in Portland, Oregon (Zwick & Pope, 2003). This data concerns health care utilization in the Kaiser member population. The figure summarizes the relationships between 4 variables in the data set, plotting the (Complexity, Information) for all 114 specific structures. The best models in this graph are its “northeastern frontier” where models are not dominated by (inferior in both complexity and information to) any other model. From this “solution set,” one might pick out the simplest model having information greater than some minimum, the most information-rich model having complexity less than some maximum, or some other single model.

Figure 5. Decomposition Spectrum. (Complexity, Information) for 114 four-variable structures; data from Kaiser Permanente Center for Health Research



2. Heuristic search (24 variables)

Basic results

From the OPUS data (subset) with sample size of 2100, twenty four possible predictors (inputs) of self-reported health status (output) were selected. The inputs include sex, age, family and work information, socioeconomic, behavioral, and psychological measures, health-related

attitudes and activities, and other variables. Quantitative input variables were discretized, and some nominal variables were also rebinned. The cardinalities of the variables as analyzed range from 2 to 6, and the product of the 25 cardinalities was 1.16×10^{13} . After analysis, the best model, involving 8 variables (7 inputs, BHLNORU, and 1 output, X), with a state space of 32,400 states, explains 65.4% of $U(X)$. This model is complex, having 11 predictive components, each of which represents a high ordinality interaction effect between four or five inputs and the output. The model, stated here without identifying the specific nature of the input variables, is:

BHLNORU:BHNRX:LORUX:LNOUX:HLOUX:HLNORX:LNRUX:HLRUX:NORUX:HORUX:HNOUX:HNRUX.

If this model had been proposed for confirmatory testing, its α , relative to a null hypothesis of no association between the 24 inputs and the 1 output, would be 0.03. (Rigorously speaking, one cannot simply say that the statistical significance of this model is 0.03 because the statistical significance of the results of an exploratory search depends upon how many models are looked at. However, reporting the significance which *would* have obtained had this model been subjected to a confirmatory test can motivate a follow-on study which explicitly tests the model. $U(X)$ may be capable of being even further reduced but if additional work is undertaken on this data set, simplifying the model to facilitate interpretation would be more critical.

Search strategy

The strategy used to obtain this model was a 2-step process:

ABCDEFGHIJKLMN OPQRSTUVWXYZ (the initial set of [24] inputs and the output [X] under consideration)

↓ Step #1: Input set reduction, using only one predicting-component models:

BHLNORUX (a reduced set of [7] inputs and the output)

↓ Step #2: Search through the lattice of multi-component-predicting models

BHLNORU:BHNRX:LORUX:LNOUX:HLOUX:HLNORX:LNRUX:HLRUX:NORUX:HORUX:HNOUX:HNRUX

In the first step, the 24 inputs were reduced to 12 and then to 7. This is “feature selection.” The subset of 7 inputs were selected by examining the uncertainty reductions achieved in single-predicting-component models. For 7 inputs, the best model is BHLNORUX. The uncertainty reduction of X

from the 7 inputs in this model is 81.7%, but this reduction is *not* statistically significant ($\alpha = 1.0$).

The 7 input model was chosen even though it was not significant because of the expectation that by decomposition the model could be simplified without major loss of predictive power. This was indeed possible. Step two yielded an 11-predicting-component model, reducing the uncertainty of the output by 65.4%, a reduction which is statistically significant ($\alpha = 0.03$). The model utilizes all 7 inputs, but in components each having 4 or 5 inputs. This decomposition preserved 80% (65.4/81.7) of the predictive information in the single-predicting-component model ($\alpha = 1.0$) obtained in the first step, BHLNORUX. This second step, unlike the first step, *does* require operations on the full state space, which here has cardinality 32,400.

Only 7 rather than a greater number of inputs were used because of the combinatorial limits of the program, which examined all single-predicting-component (SPC) models by brute force. (The program generated all models with one, two, three, etc. predicting inputs.) However, in more recent versions of the program, this brute force approach has been replaced by a breadth-first fixed width search, which allows the rapid examination of an arbitrary number of inputs. Also a related RA method exists, known as "Extended Dependency Analysis" (EDA), which allows heuristic search in single-predicting component models with 10's or 100's of variables (Conant, 1988; Lendaris, Shannon, & Zwick, 1999; Shannon & Zwick, 2003). EDA merges the inputs of several good components to produce a single-component model, without evaluating all models with this number of total inputs. Recently, in an RA study looking only at SPC models, 150 variables were reduced to 46 for subsequent use by neural net modeling (Chambless & Scarborough, 2001). To repeat a point made earlier: these calculations, involving SPC models, do not require operations on a complete state space and scale with the size of the data, which for the subset of OPUS data we used is quite modest, namely $N=2100$.

Model utilization

The 11-component model indicated above can be used in at least three ways. First, each predicting component, e.g., BHNRX or LORUX, represents a high ordinality interaction effect involving several inputs and the output (health status, X). If the inputs were originally quantitative variables, these effects are likely also to be nonlinear. Ideally, each of these components should be given a substantive interpretation of how the input variables combine to affect X. This requires extensive subject-specific knowledge, and has not yet been attempted for this study.

Second, it is possible to assess the multiple components relative to one another, to identify which are the most important. For the 7 inputs used, this 11-component model was maximally predictive (under the requirement that the model would have been statistically significant if it had been proposed for confirmatory test), but a simpler though less predictive model might be preferred. Further analysis of the model might allow such simplification.

Third, the eleven-component model yields a calculated BHLNORUX distribution, which can be converted to $p(X | B,H,L,N,O,R,U)$, a distribution on X conditioned on the input variables. Thus, given the values of all the 7 input variables, RA yields predicted probabilities for all the different possible output states. These can be compared to $p(X)$, the probabilities of the output without any knowledge of the inputs.

Finally, it should be reiterated that exploratory searches are precisely that: exploratory. Resulting models should ideally be tested in a confirmatory mode with new data.

The selected inputs and/or the specific multi-component model can be used in conjunction with methods which retain the quantitative character of variables which have been discretized. With SPC RA modeling, feature selection can be done from a large set of variables, as noted above, and the selected features fed into a neural net (NN). The NN part of such a RA-NN strategy may sometimes not be necessary. In the pattern-recognition study of Lendaris, Shannon, and Zwick (1999), RA on discretized variables essentially solved the problem by itself.

Multi-predicting-component RA modeling can also be used to prestructure neural nets, so that all-to-all connectivity is not required (Lendaris, Zwick, Mathia, 1993; Lendaris & Mathia, 1994; Lendaris, 1995; Lendaris, Rest, & Mislley, 1997); this speeds NN training times and improves generalization capacity. RA may also have promise for the prestructuring of genetic algorithms, more specifically for determining the optimal order of the variables on the GA genome (Zwick & Shervais, 2003).

2. SRA Example

SRA reconstruction is exemplified by a study which attempted to predict whether discrete dynamic systems -- specifically, elementary cellular automata (ECA) -- are chaotic or not.

An ECA is a one-dimensional array of cells, $s(1)...s(n)$, governed by a mapping, $s_t(i-1) \otimes s_t(i) \otimes s_t(i+1) \rightarrow s_{t+1}(i)$, which specifies how the state of each cell at time $t+1$ depends upon its state and the state of its two adjacent neighbors at time t . The three cells at time t and the center cell at time $t+1$ will be labelled A, B, C, and D. This is a deterministic directed system which illustrates the use of RA for time series analysis (see also Zwick, Shu, & Koch, 1996 for an IRA time series example).

An example of an ECA mapping or "rule" is given in Table 10. There are 256 mappings which preserve the identity of the 3 inputs, but in the ECA context these group into 88 equivalence classes, analyzed by SRA. Every mapping can also be converted in probability distribution by setting $p(A,B,C,D)$ to $1/8$ if an (A, B, C, D) tuple appears in the mapping and 0 if it does not. The resulting probability distributions were also analyzed by IRA.

Table 11 illustrates how a rule governs ECA dynamics. Eight cells are arranged in a toroid, so $s(9) = s(1)$. For every cell, the rule produces its next state, $s_{t+1}(i)$, given its present state, $s_t(i)$, and the present state of its left and right neighbors, $s_t(i-1)$ and $s_t(i+1)$. Eventually the system reaches either a fixed point or limit cycle attractor. Such discrete dynamics can be considered "chaotic" if the time to reach the attractor goes up rapidly with the number of cells. Assignments of chaoticity or non-chaoticity were taken from Li and Packard (1990).

Table 11. Three time steps for ECA #150. The enclosed four cells illustrate the mapping $A \otimes B \otimes C \rightarrow D$ for one time step. The array at time t is arbitrary.

t	0	1	0	1	1	0	1	0
t+1	1	1	0	0	0	0	1	1
t+2	1	0	1	0	0	1	0	1

Two "standard" parameters used to predict chaoticity were employed for comparison purposes: λ (Langton, 1992) and Z (Wuensche, 1992). The parameter λ was actually first

Table 10. An example of an ECA rule (#150).

The rule is indexed by considering the D column as a binary number, whose top-most value is its least significant bit.

t			t+1
s(i-1)	s(i)	s(i+1)	s(i)
A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

proposed by Walker and Ashby (1966), who called it "homogeneity."

SRA or IRA decomposition properties of the rules were also used (Zwick & Shu, 1997; Zwick & Shu, 2001) to predict chaoticity or non-chaoticity. It can be shown that the specific structures for this 3-input, 1-output problem can be reduced to the 12 specific structures shown in Table 12. These group into 6 levels of complexity, indexed by parameter, σ . Structures whose variables are permuted have the same complexity, but for ECAs not all permutations are equivalent, because the neighbors A and C are different from B, so there are 9 general structural types (shown in bold).

Table 12 indicates that while all single-predicting component models are themselves mappings, the $\sigma = 4$ and 5 models decompose the rule mapping into (stochastic) relations whose intersection yields the correct mapping. Such an approach to decomposition is quite different than what is encountered in LDL decomposition.

Table 12. Structures for ECA rules. The σ identifier for the 6 structural levels is given, and the 9 different specific structures are shown in bold.

σ	Structures	SRA
6	ABCD	mapping
5	ABC:ABD:ACD:BCD	3 relations (\rightarrow mapping)
4	ABC:ABD:ACD	2 relations (\rightarrow mapping)
3	ABC:ABD	mapping
2	ABC:AD	mapping
1	ABC:D	constant

By doing SRA on each rule, a σ is assigned to the rule, which indicates how decomposable without loss the rule is. Also, a vector parameter, τ , characterize each rule by the full set of decomposition losses (transmissions) for all 12 possible specific structures. IRA decomposition was also done on the rules, and two measures, f' and f'' , were calculated, which are closely related to the fluency measure of Walker and Ashby (1966). These measures involve the transmissions for the 2 of the 3 models (shown italicized in Table 12) at level $\sigma = 3$. The first measure, f' , is a vector measures which preserves information about the separate losses in the models; the second, f'' , is a scalar measure which sums the decomposition losses of these two models.

$$f' = \{T(ABC:BCD), T(ABC:ABD)\}$$

$$f'' = T(ABC:BCD) + T(ABC:ABD)$$

Table 13 indicates the predictability of chaoticity or non-chaoticity of ECA dynamics using RA measures (σ , f , f' , and τ) as compared to using standard ECA parameters (λ and Z). The table shows that RA measures predict better than standard parameters. Predictability is assessed information-theoretically as uncertainty reduction in attractor variable, a , which has two states, {chaotic, non-chaotic}. For rule parameter, r , which is either an RA measure or a standard parameter, the table lists ΔU_a and ΔU_r , which are large for good predictors, where

$$\Delta U_a = \text{fractional reduction of } U(a) \text{ knowing } r \\ = [U(a) - U(a | r)] / U(a)$$

$$\Delta U_r = \text{reduction of } U(a) \text{ per bit of predictor} \\ = [U(a) - U(a | r)] / U(r)$$

Table 13. Predicting cellular automata dynamics. Reduction of uncertainty of attractor & uncertainty reduction normalized by information of predictor, r . The best uncertainty reductions for criteria are shown in bold.

	r	$U(a r)$	ΔU_a	ΔU_r
	-	.679		
<i>Standard ECA parameters</i>				
Walker-Ashby, Langton	λ	.600	11.6	.044
Wuensche	Z	.458	32.6	.114
<i>RA measures</i>				
lossless complexity	σ	.553	18.6	.069
info.-theor. fluency	f	.355	47.7	.124
2nd fluency measure	f'	.447	34.2	.151
complete RA spectrum	τ	.263	61.3	.102

Not only do RA measures predict chaoticity or non-chaoticity better than the standard parameters of λ and Z , the RA framework actually subsumes these standard measures in τ , the complete vector of RA losses. Specifically, $U(\lambda | \tau) = U(Z | \tau) = 0$, i.e., τ specifies also λ and Z . In fact, λ turns out to be isomorphic with $U(D)$.

No SRA measure comparable to fluency was apparent, and this illustrates the point made earlier that IRA analysis can be useful even for set-theoretic functions and relations, because it presents the analytical results in a form different from SRA. Analysis using the complete loss vector τ is, however, equivalent in SRA and IRA.

IV. Software

Computations were done using a software package being developed at Portland State University named OCCAM (for the principle of parsimony and as an acronym for "Organizational Complexity Computation And Modeling"). OCCAM is intended eventually to include all data, problem,

and method types. Other RA software packages do exist, e.g., CONSTRUCT and SPECTRAL by Krippendorff (1981), SAPS by Uyttenhove (1984) and Cellier (1987), GSPS by Klir (1976), Elias (1988), and coworkers, EDA by Conant (1988), Jones' k-systems analysis (Jones, 1989) and a recent program by Dobransky and Wierman (1995). However, no package *fully* encompasses RA as shown in Table 1. Some programs are not easily used by researchers outside the systems field; others do not incorporate statistical tests. These existing packages are in limited use.

OCCAM is the result of a software development program under my direction in the Systems Science Ph.D. Program at PSU beginning in 1985. The first program, written by the author, did single-predicting component modeling. This was improved upon by Jamshid Hosseini (Hosseini, Harmon, & Zwick, 1986, 1991; Hosseini, 1987), and then by Doug Anderson who also wrote a program for multi-predicting component modeling, and another program for inconsistency resolution for IRA identification (Anderson, 1996). Hui Shu wrote SRA reconstruction and structure lattice programs. For convenience, the whole set of these earliest RA programs will be called OCCAM0. In this period, Klaus Krippendorff generously provided to us his programs mentioned above and these assisted our research and informed our development efforts. We also utilized GSPS obtained from Elias (1988). Marcus Daniels combined many functionalities of the Hosseini and Anderson reconstruction programs and Shu's lattice program by rewriting them and adding heuristic search in the multi-predicting component modeling, to produce OCCAM1. Stan Grygiel made innovations and improvements in the single-prediction-component calculations, in search heuristics, and general research usability; this produced OCCAM2. Calculations reported in this paper were done with OCCAM2 and occasionally with earlier separate programs mentioned above.

A new program (OCCAM3) has now been written by Ken Willett, which is a more effective research and applications platform (Willett & Zwick, 2003; Zwick, 2003a) and can be accessed over the web. Willett is also specifically exploring heuristic search and approximate computation approaches. Michael Johnson is programming SBM for future incorporation into the package, integrating it theoretically into the RA framework, and exploring its implications for decision analysis (Johnson & Zwick 2000; Zwick & Johnson 2003). Bjorn Chambless has written a stand-alone single-predicting-component information-theoretic program which includes binning and aggregation preprocessing capacities. Binning for OCCAM is being developed by Michael Johnson and Steve Shervais. Tad Shannon has programmed an updated version of Conant's EDA (Shannon & Zwick, 2003) and also a time-series preprocessing utility.

V. Discussion

In the example discussed above of IRA heuristic search, the first step can in principle be extended to much larger problems without difficulty, because single-predicting-component models do not require operations on the full state space, but depend rather on the size of the data. The second step, however, does require such operations, which as noted earlier limits the number of variables which can be analyzed to of order 20.

This limitation does not preclude the use of RA for data mining applications involving many more than 20 variables, in that the first step can always be used to select a smaller subset for lattice searches. Still, it would clearly be desirable if the second step could be implemented for a greater number of variables. The barrier here is the current requirement for cyclic models of an IPF operation which operates on the entire state space. If models could be generated with a procedure that operated only on observed states and scaled with the data, larger problems could be addressed, since usually data is sparse. At present, it is not apparent how to assess multi-predicting-component models in an alternative way, but two approaches which use approximate assessment and do not require the full state space will now be briefly mentioned.

The first involves the use of binary decision diagrams (Mishchenko, 2000). BDD make possible major economies of space and computing time by storing states not explicitly but implicitly in the paths of the diagram. So, while the size of the state space increases exponentially with the number of variables, the size of the graph does not, but the number of paths in the graph does, which allows the graph to represent the exponential dependence of the number of states on the number of variables. It is likely that to use BDD to analyze distributions the distribution frequencies will need to be binned, i.e., IRA problems have to be converted to SRA problems. If the resulting information loss is not severe, and if BDD can be applied to such SRA approximations, the size of the state space may become much less limiting. This approach is under investigation (Zwick & Mishchenko, 2003).

A second idea is to employ methods used in 3-dimensional image reconstruction (Zwick & Zeitler, 1973), and Fourier methods in particular. These methods allow the composition of multiple projections in a single step, regardless of cyclicity. They compute an approximation to IPF which may or may not suffice for practical purposes. Most critically, these methods scale with the data and not the state space. This approach is also under current investigation (Zwick, 2003b). The use of Fourier methods in

RA would bring RA into proximity to LDL methods which use wavelets, Walsh functions, Haar transforms, and similar global or local function-based decompositions.

In summary, RA methods are general, being applicable to set-theoretic relations as well as probability distributions. Both SRA and IRA may be of interest to the machine learning and logic design community. SRA offers another approach to decomposition of relations and mappings, and IRA can be used for these purposes as well. The LDL community might profitably examine the use of log-linear latent variable modeling and state-based RA techniques. It might also consider extension of its techniques to distributions, where statistical considerations are necessary. The RA community and the social science log-linear community, on the other hand, can gain from a deeper familiarity with the LDL literature. RA methodology is potentially a valuable new approach to data mining. Current techniques can be applied to 10's or 100's of variables, and heuristic and approximate methods may substantially expand the range of RA modeling.

Acknowledgements

I thank Pete Catching, Michael Johnson, Roberto Santiago, Tad Shannon, and Ken Willett for helpful comments on the manuscript. Special appreciation is due to Marek Perkowski for inviting me to present this paper at the 2000 LDL conference to introduce RA methodology to the LDL community, and for many stimulating conversations on decomposition techniques and other subjects.

References Cited

- Al-Rabadi, A., Zwick, M. and Perkowski, M. (2003). "A Comparison of Modified Reconstructability Analysis and Ashenurst-Curtis Decomposition of Boolean Functions." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.
- Al-Rabadi, A. and Zwick, M. (2003). "Modified Reconstructability Analysis For Many-Valued Functions and Relations." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.
- Anderson, D. R. (1996). *The Identification Problem of Reconstructability Analysis: A General Method for Estimation and Optimal Resolution of Local Inconsistency*. Systems Science Ph. D Dissertation, Portland State University. Portland, OR.
- Ashby, W. R. (1964). "Constraint Analysis of Many-Dimensional Relations." *General Systems Yearbook*, 9, 99-105.

- Bishop, Y., Feinberg, S., and Holland, P. (1978). *Discrete Multivariate Analysis*. MIT Press, Cambridge.
- Cellier, F. and Yandell, D. (1987). "SAPS-II: A New Implementation of the Systems Approach Problem Solver." *Int. J. of General Systems*, 13 (4), pp. 307-322.
- Cellier, F., Nebot, A., Mugica, F., and deAlbornoz, A. (1995). "Combined Qualitative -Quantitative Simulation Methods of Continuous-Time Process Using Fuzzy Inductive Reasoning Techniques." *Int. J. General Systems*, 24 (1&2), pp. 95-116.
- Chambless, B. and Scarborough, D. (2001). "Information-Theoretic Feature Selection for a Neural Behavioral Model." *Int. Joint Conference on Neural Nets*. Washington, DC, July 14-19.
- Chen, Z. (1994). "Qualitative Reasoning for System Reconstruction Using Lebesgue Discretization." *Int. J. Systems Sci.*, 25 (12), pp. 2329-2337.
- Chen, Z. (1997). "K-Systems Theory for Goal-driven Data Mining." *Advances in Systems and Applications* Special Issue 1, pp. 40-43.
- Conant, R. C. (1981). "Set-Theoretic Structure Modeling." *Int. J. General Systems*, 7, 93-107.
- Conant, R. C. (1988). "Extended Dependency Analysis of Large Systems. Part I: Dynamic Analysis." *Int. J. General Systems*, 14, 97-123.
- Conant, R. C. (1988). "Extended Dependency Analysis of Large Systems. Part II: Static Analysis." *Int. J. General Systems*, 14, 125-141.
- Davis, J. A. (1985). *The Logic of Causal Order* (Quantitative Applications in the Social Sciences #55). Sage, Beverly Hills.
- Dobransky, M. and Wierman, M (1995). "Genetic Algorithms: A Search Technique Applied to Behavior Analysis." *Int. J. of General Systems*, 24 (1&2), pp. 125-136.
- Elias, D. (1988). *The General Systems Problem Solver: A Framework for Integrating Systems Methodologies*. Ph.D. Dissertation, Dept. of Systems Science, SUNY-Binghamton.
- Feldman, D. and Crutchfield, J. (1998). "Discovering Noncritical Organization: Statistical Mechanical, Information Theoretic, and Computational Views of Patterns in One-Dimensional Spin Systems." *Santa Fe Institute Working Paper* 98-04-026.
- Files, C. and Perkowski, M. (1998). "Multi-Valued Functional Decomposition as a Machine Learning Method," *Proc. ISMVL'98*, May 1998.
- Good, I. (1963). "Maximum Entropy for Hypothesis Formulation Especially for Multidimensional Contingency Tables." *Annals of Mathematical Statistics*, 34, pp. 911-934.
- Grygiel, S. (2000). *Decomposition of Relations as a New Approach to Constructive Induction in Machine Learning & Data Mining*. Electrical Engineering Ph.D. Dissertation, PSU, Portland OR.
- Grygiel, S., Zwick, M., and Perkowski, M. (2003). "Multi-level Decomposition of Relations." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.
- Hagenaars, J. A. (1993). *Loglinear Models With Latent Variables*. (Quantitative Applications in the Social Sciences #94). Sage, Beverly Hills.
- Hosseini, J., Harmon, R. and Zwick, M. (1986). "Segment Congruence Analysis Via Information Theory." *Proceedings, International Society for General Systems Research*, Philadelphia, PA, May 1986, pp. G62 - G77.
- Hosseini, J (1987). "Segment Congruence Analysis: An Information Theoretic Approach." *Systems Science Ph.D. Dissertation*, Portland State University, Portland OR.
- Hosseini, J., Harmon, R. and Zwick, M. (1991). "An Information-Theoretic Framework for Exploratory Segmentation Research." *Decision Sciences* 22, 663-677.
- Johnson, M. and Zwick, M (2000). State-Based Reconstructability Modeling For Decision Analysis. In *Proceedings of The World Congress of the Systems Sciences and ISSS 2000*, Allen, J.K. and Wilby, J.M. eds., Toronto, Canada: International Society for the Systems Sciences. Forthcoming.
- Jones, B. (1985a). "Reconstructability Analysis for General Functions." *Int. J. General Systems*, 11, pp. 133-142.
- Jones, B. (1985b). "Reconstructability Considerations With Arbitrary Data." *Int. J. General Systems*, 11, pp. 143-151.
- Jones, B. (1986). "K-Systems Versus Classical Multivariate Systems." *Int. J. General Systems*, 12, pp. 1-6.

- Jones, B. (1989). "A Program for Reconstructability Analysis." *Int. J. General Systems*, 15, pp. 199-205.
- Kim, J. and Mueller, C. (1978). *Factor Analysis: Statistical Methods and Practical Issues*. (Quantitative Applications in the Social Sciences #14). Sage, Beverly Hills
- Klir, G. (1976). "Identification of Generative Structures in Empirical Data." *Int. J. General Systems*, 3 (2), pp. 89-104.
- Klir, G. (1985). *The Architecture of Systems Problem Solving*. Plenum Press, New York.
- Klir, G. (1986). "Reconstructability Analysis: An Offspring of Ashby's Constraint Theory." *Systems Research*, 3 (4), pp. 267-271.
- Klir, G., ed. (1996). *Int. J. General Systems* Special Issue on GSPS, 24 (1-2) (includes an RA bibliography).
- Klir, G. and Wierman, M. J. (1998). *Uncertainty-Based Information: Variables of Generalized Information Theory*. Physica-Verlag, New York.
- Knoke, D. and Burke, P. J. (1980). *Log-Linear Models*. (Quantitative Applications in the Social Sciences Monograph # 20). Sage, Beverly Hills.
- Krippendorff, K. (1981). "An Algorithm for Identifying Structural Models of Multivariate Data." *Int. J. of General Systems*, 7 (1), pp. 63-79.
- Krippendorff, K. (1986). *Information Theory: Structural Models for Qualitative Data*. (Quantitative Applications in the Social Sciences #62). Sage, Beverly Hills.
- Ku, H. and Kullback, S. (1968), "Interaction in Multidimensional Contingency Tables: An Information Theoretic Approach." *Journal of Research of the National Bureau of Standards - Mathematical Sciences*, 728 (3), pp. 159-199.
- Kullback, S. (1959). *Information Theory and Statistics*. John Wiley, New York.
- Lendaris, G., Zwick, M. and Mathia, K. (1993). "On Matching ANN Structure to Problem Domain Structure." *Proceedings of the World Congress on Neural Nets '93*, Portland.
- Lendaris, G. & Mathia, K. (1994). "Using A Priori Knowledge to Prestructure ANNs", *Australian Journal of Intelligent Information Systems*, 1 (1).
- Lendaris, G. (1995), "Prestructuring ANNs via A Priori Knowledge", *Proceedings of SPIE International Conference*, SPIE Press.
- Lendaris, G., Rest, A. & T. Mисley, T. (1997). "Improving ANN Generalization Using A Priori Knowledge to Pre-Structure ANNs", *Proceedings of International Conference on Neural Networks '97 (ICNN'97)*, IEEE Press, July 1997.
- Lendaris, G., Shannon, T. and Zwick, M. (1999). "Prestructuring Neural Networks for Pattern Recognition Using Extended Dependency Analysis" (invited paper). *Proceedings of Applications and Science of Computational Intelligence II AeroSense '99*, Orlando, FL, SPIE.
- Li, W. & Packard, N. (1990). "The Structure of the Elementary Cellular Automata Rule Space." *Complex Systems* 4, pp. 281-297.
- Long, J. S. (1983). *Covariance Structure Models: An Introduction to LISREL* (Quantitative Applications in the Social Sciences #34). Sage, Beverly Hills.
- Mariano, M. (1984). "Towards Resolving Inconsistency Among Behavior Systems." In *Proceedings - Society for General Systems Research International Conference 1984, Vol. I: Systems Methodologies and Isomorphies*, Smith, W., ed., Intersystems Publications, Lewiston, N.Y., pp. 225-229.
- Mariano, M. (1987). *Aspects of Inconsistency in Reconstruction Analysis*. Ph. D Dissertation, Department of Systems Science, SUNY-Binghamton.
- McCutcheon, Allan L. (1987). *Latent Class Analysis*. (Quantitative Applications in the Social Sciences #64). Sage, Beverly Hills.
- Miller, G. and Madow, W. (1954), *On the Maximum Likelihood Estimate of the Shannon-Wiener Measure of Information*. Air Force Cambridge Research Center, Washington 25, D.C.
- Mishchenko, A. (2000). "Implicit Representation of Discrete Objects."
- Perkowski, M., Marek-Sadowska, M., Jozwiak, L., Luba, T., Grygiel, S., Nowicka, M., Malvi, R., Wang, Z., and Zhang, J. "Decomposition of Many-Valued Relations," *Proc. ISMVL '97*, Halifax, Nova Scotia, May 1997, pp. 13-18.
- Pittarelli, (1990). "Reconstructability Analysis Using Probability Intervals." *Int. J. General Systems*, 16, pp. 215-233.

Shaffer, G. and Cahoon, P. (1987). "Extracting Information from Ecological Data Containing High Spatial And Temporal Variability: Benthic Microfloral Production." *Int. J. General Systems*, 13, pp. 107-123.

Shaffer, G. (1988). "K-Systems Analysis for Determining the Factors Influencing Benthic Microfloral Productivity in a Louisiana Estuary." *USA Mar. Ecol. Prog. Ser.* 43, pp. 43-54.

Shannon, T. and Zwick, M. (2003). "Directed Extended Dependency Analysis for Data Mining." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.

Uyttenhove, Hugo J.J. (1984), "SAPS - A Software System for Inductive Modelling." In: *Simulation and Model-Based Methodologies: An Integrative View*, edited by T.I Oren et al. Springer-Verlag, Berlin, Heidelberg. NATO ASI Series, Vol. F10, pp.427-449.

Vermunt, J. (1997). *Lem: A General Program for the Analysis of Categorical Data* (Program manual). Tilburg University.

Walker, C.C. and Ashby, W. R. (1966). "On Temporal Characteristics of Behavior in Certain Complex Systems." *Kybernetik*, 3 (2), pp. 100-108.

Willett, K. and Zwick, M. (2003). "A Software Architecture for Reconstructability Analysis." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.

Zadeh, L. A. (1965). "Fuzzy Sets." *Information and Control* 8 (3), 338-353.

Zwick, M. (1996). "Control Uniqueness in Reconstructability Analysis." *Int. J. General Systems*, 24 (1-2), 151-162.

Zwick, M. (2001). "Wholes and Parts in General Systems Methodology." *The Character Concept In Evolutionary Biology*. Günter Wagner, ed., Academic Press, New York.

Zwick, M. (2003a). "Discrete Multivariate Modeling." http://www.sysc.pdx.edu/res_struct.html

Zwick, M. (2003b). "Reconstructability Analysis With Fourier Transforms." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.

Zwick, M. and Johnson, M. (2003). "State-Based Reconstructability Analysis." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.

Zwick, M. and Shervais, S. (2003). "Reconstructability Analysis Detection of Optimal Gene Order in Genetic Algorithms." *Kybernetes*, vol.33, No. 5/6, pp. 877-905.

Zwick, M. and Shu, H. (1997). "Set-Theoretic Reconstructability of Elementary Cellular Automata." *Advances in Systems Science and Applications*. Special Issue 1, 31-36.

Zwick, M. and Shu, H. (2003). "Reconstructability and Dynamics of Elementary Cellular Automata." In preparation.

Zwick, M., Shu, H. and Koch, R. (1996). "Information-Theoretic Reconstructability of Rainfall Time-Series Data." *Advances in Systems Science and Applications*, Special Issue I, pp. 154-159.

Zwick, M. and Mishchenko, A. (2003). Binary Decision Diagrams and Reconstructability Analysis in Crisp Possibilistic Systems (In preparation).

Zwick, M. and Zeitler, E. (1973). "Image Reconstruction from Projections." *Optik*, 38, 550-565.

BIOGRAPHY

Martin Zwick

Martin Zwick is a Professor of Systems Science at Portland State University. Prior to taking his current position at PSU, he was a faculty member in the Department of Biophysics and Theoretical Biology at the University of Chicago, where he worked in macromolecular structure and mathematical crystallography. In the 1970's his interests shifted to systems theory and methodology. Since 1976 he has been on the faculty of the PSU Systems Science Ph.D. Program and during the years 1984-1989 he was the program director. His current research interests are in discrete multivariate modeling (reconstructability analysis), "artificial life" and theoretical/computational biology, and systems philosophy.