

Portland State University

PDXScholar

Systems Science Faculty Publications and
Presentations

Systems Science

2004

Directed Extended Dependency Analysis for Data Mining

Thaddeus T. Shannon
Portland State University

Martin Zwick
Portland State University, zwick@pdx.edu

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



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

Let us know how access to this document benefits you.

Citation Details

Shannon, T. and Zwick, M. 2004. "Directed Extended Dependency Analysis for Data Mining." *Kybernetes*, vol. 33, No. 5/6, pp. 973-983.

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.

Directed Extended Dependency Analysis for Data Mining

Thaddeus T. Shannon and Martin Zwick
Systems Science Program, Portland State University

Abstract: Extended Dependency Analysis (EDA) is a heuristic search technique for finding significant relationships between nominal variables in large datasets. The directed version of EDA searches for maximally predictive sets of independent variables with respect to a target dependent variable. The original implementation of EDA was an extension of reconstructability analysis. Our new implementation adds a variety of statistical significance tests at each decision point that allow the user to tailor the algorithm to a particular objective. It also utilizes data structures appropriate for the sparse datasets customary in contemporary data mining problems. Two examples that illustrate different approaches to assessing model quality tests are given.

I. Introduction

Extended Dependency Analysis (EDA) is a heuristic search strategy, proposed by Conant [1][2] that directs the information theoretic exploration of large sets of nominal variables. It is an extension of Reconstructability Analysis (RA) [5][6][8][13]. The number of computations involved in a complete RA scale doubly exponentially in the number of variables considered, while the more limited directed Dependency Analysis (DA) scales exponentially in the number of variables (RA allows an arbitrary number of relations in a model, while DA considers models with only one predicting relation). Conant devised EDA to scale polynomially with respect to the number of variables considered, while still being capable of discovering significant high order interactions between variables.

In this paper we illustrate some possible uses for EDA in the data mining context. The analysis in the examples we present was performed with a new software implementation developed for use on large data sets [12]. The examples we present are intended to illustrate some of the goals and issues that arise in data mining. The problem data sets themselves are still relatively small (26 and 195 independent variables) and so are not indicative of EDA's true potential, since in both cases the analysis can be performed in under a minute on a personal computer.

In the following section, we begin by reviewing Conant's original work in our own terms. The third section is a discussion of implementation issues we believe germane in a data mining context. We introduce several variations on Conant's original proposal to address these issues. In the next sections we present two examples of data mining tasks and the results we obtained using EDA: developing a forecasting model from a rainfall time series, and designing a pattern classifier for satellite imagery. The final section extends our earlier discussion of implementation options and suggests further applications for EDA.

II The Origins of EDA

Reconstructability Analysis and related methods are techniques that seek to explain some set of observed variables in terms of a set of one or more relations among the variables [5][6][8]. The variables are assumed to be nominal and the sense in which the relations explain the observations differs depending on the nature of the data. The nominal variables may be crisp possibilistic in which case the analysis is essentially set theoretic. Alternatively, the variables may be probabilistic, i.e., random variables with some joint distribution. In both the crisp possibilistic and probabilistic cases, RA seeks to find projections of the joint distribution that can reconstruct the original distribution when joined using the maximum uncertainty principle. EDA applies to probabilistic systems, and that is the focus of this paper.

The variables that necessarily participate in a projection together are dependent on each other and form a relation. In what Conant termed a "static" analysis, finding all significant dependencies in a set of variables is the goal of the analysis, this corresponds RA for a neutral (undirected) system. Alternatively, what Conant called a "dynamic" analysis seeks to explain one particular variable using all the other variables; this corresponds to RA for a directed system. In this case, one wants to find all the relations (projected distributions) in which the variable of interest participates. The set of variables in the identified relations together explain the dependent variable in the sense that they maximally reduce the uncertainty of the dependent variable. This is the view we take for our data mining context.

RA is a method for finding the significant relations among a set of variables. The set of all possible relations that could be searched through grows doubly exponentially with the number of variables considered [4]. Thus direct application of RA quickly becomes impractical as the number of variables grows beyond seven or eight. Conant's EDA escapes this curse of dimensionality by limiting the search to the subset of "saturated" (single

relation) models that grows only polynomially with the number of variables. EDA will not therefore generally find the best possible model, i.e. the full set of significant relations.

EDA has two distinct phases, a directed modeling phase and a undirected modeling phase. If EDA is applied to a directed system, only the directed analysis need be performed. For a undirected analysis, the directed phase is carried out repeatedly with each variable in turn considered as the dependent variable, and the results are then aggregated in the undirected phase. The directed analysis is composed of three operations, an initial search named *Initialize* (Conant's H2), followed by repeated applications of a routine named *Reduce* (based on DA) and a second search heuristic named *Expand* (Conant's H3). The general relationship of the search phases is illustrated in Figure 1.

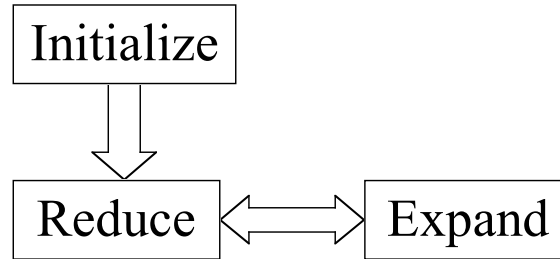


Figure 1 EDA Directed Analysis

The first heuristic, *Initialize*, begins by calculating the three-way transmissions (mutual information)

$$T(D;C_iC_j) = u(D|C_iC_j) - u(D|C_i) - u(D|C_j) + u(D) + u(C_iC_j) - u(C_i) - u(C_j) + u(C_iC_j),$$

where D is the dependent variable of interest, C_i is the i -th candidate variable [7], and u is information-theoretic uncertainty (Shannon entropy). All possible combinations of candidate variables are tried. These transmission values are the strengths of the relationships between the dependent variable and the pair of independent variables. If the dependent variable is completely determined when the values of the independent variables is known, this transmission value will equal the initial uncertainty of the dependent variable. If on the other hand the dependent variable is independent of the independent variables, this transmission will be zero. Given n independent variables there are $n(n-1)/2$ such pairs to check. For each independent variable C_i , the transmission value for all possible pairings with other independent variables is calculated, and the highest transmission value found is stored if that interaction passes a chi-square significance test. This results in a list of variables with transmission values of length n , which can be sorted based on the transmission values. The d variables with the highest transmission values then form an initial model $M_I = C_1C_2...C_d$, the search set of *candidate variables*. The size of the search set is the user selected parameter that controls both model complexity and run time for the analysis. In essence, this heuristic performs an exhaustive search through the triadic relationships involving the dependent variable.

Reduce tests the significance of the uncertainty reduction of the dependent variable provided by a candidate model M . The significance of the participation of each individual variable in the model is evaluated in the context of the high order relation posed by the model. A significance criteria, either statistical or information theoretic, must be provided for this evaluation. Some possible evaluation criteria and the problem contexts in which they could be appropriate are discussed in the following section on implementation issues. Variables that are found to be not significant are eliminated from the candidate model, resulting in a simplified model $M_S = M_I - C_{NS}$.

The second heuristic, *Expand*, takes the candidate set of variables passed from *Reduce* and attempts to expand it by checking for high order relations involving the dependent variable, all the current candidate variables, and every individual non-candidate variable. Specifically, the transmissions $T(D;M_S C_{NT})$ are calculated for each variable C_{NT} not tried already in the candidate set. This search allows variables into the explanatory set that are found to take part in higher order relationships to a significant degree (as determined by a chi-square test), even though they were not found to take part in significant relationships in *Initialize*. The search alternates between *Expand* in which the candidate model is expanded to contain d variables, and *Reduce* in which insignificant variables are discarded. The search ends when a set of d significant variables has been found, or when *Expand* can find no more candidate variables to add to the model. It is almost always the case that the bulk of the computation in any analysis takes place in *Reduce*.

III Implementation Issues for Data Mining

As the primary reason for adopting a set of search heuristics such as EDA is to avoid exponential scaling of the computational load with respect to the number of variables considered, there are a number of implementation issues that require attention. Consider for example, the analysis of 100 binary valued variables. In principle the complete contingency table representing this dataset contains 2^{100} entries. In practice, we will find this table sparsely populated, if for no other reason than that we have a limited amount of data (in this example one observation requires 12.5 bytes of storage, so 1 tera-byte of observations, i.e. 2^{40} observations requires 12.5 tera-bytes of storage, but would still leave at least 2^{60} empty entries in the table). Analysis of large numbers of variables necessarily implies comparative sparseness of data.

The first consequence of this observation is that sparse matrix methods should be used for data representation and manipulation. The size of the overall contingency table for a dataset scales exponentially in the number of variables; therefore, to maintain the desired polynomial scaling of EDA we must omit all the empty cells from our representation.

The second consequence of data paucity is the limitation of our ability to test large models or relations for statistical significance. As a rule of thumb, one needs five times as many observations as degrees of freedom in a model to justify a chi-square significance test of the model. The degrees of freedom for the models we consider are

$$\text{df}(D; C_1 \cdots C_n) = (\text{df}(D) - 1) \left[\left(\prod_{i=1}^n \text{df}(C_i) \right) - 1 \right].$$

Thus, the number of observations we need to justify fully analyzing a dataset (i.e. test all possible models) scales exponentially in the number of variables. As the above example illustrates, one needs an astronomical number of observations to justify completely analyzing even a medium size collection of variables.

Thus, the domain of EDA is a model search space that scales doubly exponentially in the number of variables over an observation state space that scales exponentially in the number of variables, wherein the required number of observations for a complete statistical analysis also scales exponentially. While the scaling of the observation space can be completely handled for any practical implementation by omitting all the empty cells from the contingency table, choosing which subsets of models to check and how to check them requires factoring together the goals of the analysis with the actual data constraint encountered.

Three alternative analyses can be performed, one in which models are compared to each other incrementally considering both complexity and uncertainty reduction, a second in which models are compared to a reference model (usually the independence model) to maximize uncertainty reduction subject to a significance constraint, and a third in which models are compared to a reference model to minimize complexity subject to an uncertainty reduction constraint. The first case uses an incremental significance test between models; the second a cumulative significance test with respect to the reference model; the third imposes no significance test. Different problem contexts suggest these different kinds of analyses. The first case might involve searching for significant relationships between variables that provide meaningful scientific explanations of observed phenomena. The second case might aim at developing a model that is maximally predictive of another variable. The third case might involve selection of features for use in compression or pattern recognition. These objectives suggest the use of different significance tests at the various decision points within EDA.

In the first case, the objective is to draw only the conclusions that are significantly supported by the data. If there is not enough data to justify a relation, then the relation is not considered in the analysis. In this case, one would use the usual chi-square significance tests with an appropriately chosen α at each decision point. Usually this test would be an aggregate test of the model H_1 versus the independence model H_0 ,

$$L^2(M) = n (1.3863) T(D;M) \sim \chi^2(\text{df}(D;M)),$$

in *Initialize* and an inter-model (incremental) test,

$$L^2(M_n \rightarrow M_{n-1}) = n (1.3863) [T(D;M_n) - T(D;M_{n-1})] \sim \chi^2(\text{df}(D;M_n) - \text{df}(D;M_{n-1})),$$

in *Reduce* and *Expand*. In addition, one sets the search set size to limit the number of models tested that contain too many degrees of freedom to justify testing, i.e. models that are *a priori* likely fail the significance test due to sample size. One would like to avoid testing such models and if the ordinality of all the independent variables is the same, this can be done exactly, i.e. no *a priori* likely fail models need be tested. Otherwise, the search set size must be large enough to include all the potentially justifiable models and some unjustified models will be included.

In the second case, the objective is to obtain the maximum uncertainty reduction possible. With comparatively small numbers of observations available, the probability of committing a type II error with an incremental significance test will tend to be high. A more appropriate significance test in *Reduce* would be the

cumulative significance of the proposed model versus the independence model. One can drop the significance test from *Expand* altogether at the cost of lengthening the analysis somewhat. Changing to a cumulative test in *Reduce* tends to abbreviate the search, since a depth first search will tend to stop at a larger model, which will in turn tend to limit the search depth. As *Reduce* is less computationally costly with the cumulative test, it can be applied to check a greater number of candidate variables.

In the third case, the objective is data reduction without serious information loss. There may be significant but minor information that can be discarded to obtain better compression, or there may be insufficient data to establish significance using standard statistical tests. In this later situation one is often faced with design challenges in which a large dataset is thought to contain sufficient information to achieve an objective. A wide variety of machine learning algorithms have been developed to extract significant rules from such datasets, e.g. artificial neural networks, tree classifiers, etc. Implementation of any such algorithm is greatly benefited by a reduction in the dimensionality of the dataset. Obviously, any dimensionality reduction strategy employed needs to minimize the inherent loss of information. EDA implemented in information only mode (with no statistical significance test) provides a method for identifying those variables that can be discarded without serious information loss. In this context, the significance of the final model can be treated explicitly using the tools of structural risk minimization [3].

IV Application: Mask Analysis of Time Series for Prediction

This example application uses EDA to perform mask analysis on a multivariate time series, with the goal of forecasting future states. The data consists of daily rainfall measurements collected at four sites over the period from 1982 to 1990. The original data series was quantitative (inches of rainfall), for our analysis we abstracted the data into two values: measurable precipitation and no measurable precipitation. This example is an extension and refinement of the analysis reported in Zwick et al. [14]. In this example, we use cumulative tests for model significance, i.e. we seek the most explanatory model that is significant compared to the independence model.

We form five variables out of the time series for each site by taking the current and the first through fourth lagged values of the time series as the variables current values. This variable assignment is summarized in Table 1.

day :	t - 4	t - 3	t - 2	t - 1	t
site: 1	Q	M	I	E	A
2	R	N	J	F	B
3	S	O	K	G	C
4	T	P	L	H	D

Table 1. Mask analysis framework

We also introduce a set of seasonal variables to include in the analysis. Since we do not know *a priori* how many seasons to include, nor when to define the start of each season, we include a range of possible season variables as defined in Table 2.

W1	12 seasons
W2	6 seasons (Jan/Feb, ...)
W3	4 seasons (Jan-Mar, Apr-Jun, ...)
W4	2 seasons (Oct-Mar, Apr-Sep)
W5	2 seasons (Nov-Apr, May-Oct)
W6	2 seasons (Dec-May, Jun-Nov)
W7	2 seasons (Jan-Jun, Jul-Dec)
W8	2 seasons (Feb-Jul, Aug-Jan)
W9	2 seasons (Mar-Aug, Sep-Feb)
W10	2 seasons (Nov-May, Jun-Oct)

Table 2. Season variables considered

In [14] it was noted that A, B, C, and D are not independent, i.e. if we form the aggregate variable $Z = ABCD$, then

$$u(Z) < u(A) + u(B) + u(C) + u(D) = u(A:B:C:D).$$

In this case $u(Z) = 3.25$ bits, which is significantly less than $u(A:B:C:D) = 3.79$ bits. The aggregate variable was then used as the dependent variable for the rest of the analysis. The analysis was further simplified by including only first and second lags of observed rainfall. Finally, a seasonal variable was selected in a preliminary analysis that compared the predictive power (uncertainty reduction divided by degrees of freedom) of each of the candidate variables in isolation. The preliminary analysis chose W_{10} as the most efficient predictor of Z . The analysis went on to find that

$$Y = FGHW_{10}$$

was the best statistically significant predictor of Z , with $u(Z|Y) = 2.52$ bits for an uncertainty reduction of 22.4%.

Using EDA we analyzed each dependent variable separately, including all 26 of the independent variables E, \dots, T , and W_1, \dots, W_{10} in each case. The models suggested by *Initialize* were:

$$\begin{array}{lll} Y_A = EFGHJLW_1W_9W_{10}, & u(A|Y_A) = 0.52 \text{ bits}, & \Delta u = 39.1\%, \\ Y_B = FGHJLPW_1W_9W_{10}, & u(B|Y_B) = 0.56 \text{ bits}, & \Delta u = 41.9\%, \\ Y_C = FGHJLW_1W_6W_9W_{10}, & u(C|Y_C) = 0.67 \text{ bits}, & \Delta u = 32.2\%, \\ Y_D = FGHJKLNPTW_6W_{10}, & u(D|Y_D) = 0.41 \text{ bits}, & \Delta u = 58.1\%. \end{array}$$

None of these intermediate models is statistically significant. Note that all these initial proposals include two or more seasonal variables. We would anticipate that these variables should be redundant and that the analysis will select the best from among them. The best models obtained at the end of the analysis using a cumulative significance test with an alpha of 0.05, and a search set allowing up to 9 independent variables, (11 for site 4) were:

$$\begin{array}{lll} Y_A = FGHJLSTW_1, & u(A|Y_A) = 0.42 \text{ bits}, & \Delta u = 50.5\%, \\ Y_B = FGHLNORW_1, & u(B|Y_B) = 0.46 \text{ bits}, & \Delta u = 52.3\%, \\ Y_C = FGHJOSTW_1, & u(C|Y_C) = 0.48 \text{ bits}, & \Delta u = 51.6\%, \\ Y_D = FGHKNPTW_1, & u(D|Y_D) = 0.46 \text{ bits}, & \Delta u = 53.5\%. \end{array}$$

These final models include only one seasonal variable each, in all cases W_1 , which differentiates all twelve months was chosen. In aggregate we then have

$$u(A|Y_A: B|Y_B: C|Y_C: D|Y_D) = 1.82 \text{ bits}, \quad \Delta u = 52.0\%,$$

which represents a reduction in uncertainty of 0.7 bits, approximately 28%, beyond that of the best joint predictor previously found. Finding the best predictors of A, B, C, and D separately is better than finding the best predictors of a single joint dependent variable, $Z = ABCD$. Further reduction in uncertainty may be possible by using the constraints among A, B, C, and D, by adding a model component specifying these constraints.

V Application: Classifier Design with Limited Observations

This example demonstrates the use of EDA in a case where there is too little data to justify significance tests of all but the simplest relations. For classifier design from a Bayesian viewpoint, the preferred procedure would be structural risk minimization. One would trade classifier simplicity against performance on the limited design examples. Greater simplicity tends to produce poorer performance on design samples but increases the likelihood that generalization performance will mirror performance on design examples.

Our proposal here is to use EDA to select a limited number of features out of all the observable variables, for use in a classifier. Furthermore, we can use the contingency table from EDA to form a prototype classifier that can be used as is or refined using other methods. In this context, performing statistical significance tests in EDA would amount to picking one particular solution to the structural risk minimization problem without considering the actual structural form of the final classifier. Worse yet, this solution would tend not to be a good starting point for further refinement since potentially useful information will already have been lost. Instead, we suggest using the information only version of EDA to select features with the structural risk minimization and refinement processes handled explicitly as separate steps.

Our example problem is that of designing a land-use classifier for satellite imagery. The images are 2-kilometer circular aperture photographs of the Phoenix metropolitan area, derived from plates taken during the Sky Lab II mission [10]. Each image is represented by wedge and annular ring samples of its 2-dimensional Fourier transform. One hundred 1.8° wedge samples and ninety-five ring samples were taken from each transform. This representation offers translation, size and rotational invariance for items in each image [9]. Five land use classes are represented in the sample: urban, residential, farm, mountain and water. While we have 195 observables per image, we have only 177 extant images to work with. We divide these into two sets, a 100-image set for classifier design, and a 77-image set for classifier evaluation. We suggest that this condition of limited data is actually representative of a major class of problems, e.g. genome studies.

Since the ring and wedge samples are quantitative variables, we begin by binning each variable. The fewer bins we use per variable, the smaller the contingency table will be. Nevertheless, our binning should preserve meaningful distinctions between values for each variable. In this case, we choose to use five bins per variable based on the surmise that five land use types could give rise to five distinct values. We break up the total observed interval of values for each variable into five subintervals of equal length and assign observations to bins based upon which subinterval they fall into.

Using EDA we searched through the entire set of wedge (W) and ring (R) variables. The intermediate models suggested by *Initialize* for a search set of size eight was:

$$Y = W_{37}W_{70}W_{83} W_{84}R_{47}R_{48}R_{49}R_{52}, \quad u(\text{type} | Y) = 0.0 \text{ bits}, \quad \Delta u = 100 \%,$$

This predictor is not statistically significant, however, it is the smallest set that *Initialize* can find that completely explains land use type. The best statistically significant predictor, however, using a cumulative significance test with an alpha of 0.05 and a search set of size eight was:

$$Y = W_{37}R_{48}R_{81}, \quad u(\text{type} | Y) = 0.15 \text{ bits}, \quad \Delta u = 93.7 \%,$$

We have several options in the data reduction context, we could keep all the variables selected by *Initialize* for use in our final classifier design, we could use only the three variables included in the statistically significant predictor, or we could employ *Reduce* and *Expand* with information content tests to see if there is a subset of the *Initialize* dependency set that still contains all the information necessary to determine land use type. Applying this last procedure, we find that there are four equally explanatory dependency sets of size four:

$$Y = W_{37}W_{83} W_{84}R_{47},$$

$$Y = W_{37}W_{83} W_{84}R_{48},$$

$$Y = W_{37}W_{83} W_{84}R_{49},$$

$$Y = W_{70}W_{83} W_{84}R_{47},$$

and one of size five,

$$Y = W_{70}W_{83} W_{84}R_{48}R_{52},$$

all with

$$u(\text{type} | Y) = 0.0 \text{ bits}, \quad \Delta u = 100 \%,$$

This additional analysis suggests that none of the eight variables found by *Initialize* is irrelevant, but there is redundancy when using all eight. In the final models, all of the variables appear in at least one set of size four or five. That Ring 52 does not participate in any of the sets of size four suggests that it could be a candidate for elimination. When the above binning scheme and analysis are used to directly construct a classifier, the generalization rates on the holdout dataset for the four variable predictors range from 89% to 94% [11].

VI Conclusion

In summary, directed EDA is a useful method for data reduction and forecasting problems involving large numbers of nominal variables. Our recent software implementation can apply EDA's polynomial time search heuristics on large numbers of variables using a variety of significance tests appropriate for data mining problems.

VII Acknowledgements

This work was partially supported by the National Science Foundation under grant ECS-9904378. We would like to thank Roy Koch for access to the rainfall data, George Lendaris for the use of the satellite imagery, and Roger Conant for sharing his original EDA routines.

VIII References

- [1] R.C. Conant, "Extended Dependency Analysis of Large Systems; Part I: Dynamic Analysis." *Int. J. General Systems*, Vol. 14, 1987, pp. 97-123.
- [2] R.C. Conant, "Extended Dependency Analysis of Large Systems; Part I: Static Analysis." *Int. J. General Systems*, Vol. 14, 1987, pp. 125-141.
- [3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 1999.
- [4] G.J. Klir, Special issue on Reconstructability Analysis, *International Journal of General Systems*, 7 (1), 1981.
- [5] G.J. Klir, *Architecture of Systems Problem Solving*, Plenum Publishing, New York, 1985.

- [6] G.J. Klir, "Reconstructability Analysis: An Offspring of Ashby's Constraint Theory." *International Journal of General Systems*, 3 (4), pp. 267-71, 1996.
- [7] G.J. Klir and M.J. Wierman, *Uncertainty-Based Information*. Physica-Verlag, Heidelberg, 1998.
- [8] K. Krippendorff, *Information Theoretic Structural Models for Qualitative Data*. Sage, 1986.
- [9] G.G. Lendaris and G.L. Stanley, "Diffraction pattern sampling for pattern recognition." *Proceedings of IEEE*, vol. 58, # 2, pp. 198-216, February, 1970.
- [10] G.G. Lendaris and S.B. Chism, *Land-Use Classification of Skylab S-190B Photography Using Optical Fourier Transform Data*. NASA LBJ Space Center, # LEC-5633, March, 1975.
- [11] G.G. Lendaris, T.T. Shannon and M. Zwick, "Prestructuring Neural Networks for Pattern Recognition Using Extended Dependency Analysis." *Proceedings of Applications and Science of Computational Intelligence II -AeroSense '99*, Orlando, FL, SPIE, April, 1999.
- [12] http://www.sysc.pdx.edu/res_struct.html
- [13] M. Zwick, "Wholes and Parts in General Systems Methodology." in G. Wagner ed., *The Character Concept in Evolutionary Biology*, Academic Press, 2001.
- [14] M. Zwick, H. Shu and R. Koch, "Information-Theoretic Mask Analysis of Rainfall Time-Series Data." *Advances in Systems Science and Applications*, Special Issue 1, pp. 154-159, 1995.