A SMART GUIDE TO DUMMY VARIABLES: FOUR APPLICATIONS AND A MACRO

Susan Garavaglia and Asha Sharma
Dun & Bradstreet
Murray Hill, New Jersey 07974

Abstract: Dummy variables are variables that take the values of only 0 or 1. They may be explanatory or outcome variables; however, the focus of this article is explanatory or independent variable construction and usage. Typically, dummy variables are used in the following applications: time series analysis with seasonality or regime switching; analysis of qualitative data, such as survey responses; categorical representation, and representation of value levels. Target domains may be economic forecasting, bio-medical research, credit scoring, response modeling, and other fields. Dummy variables may serve as inputs in traditional regression methods or new modeling paradigms, such as genetic algorithms, neural networks, or Boolean network models. Coding techniques include "1-of-N" and "thermometer" encoding. Statistical properties of dummy variables in each of the traditional usage and application contexts are discussed, and a more detailed introduction of a Boolean network model is presented. Because conversion of categorical data to dummy variables often requires time-consuming and tedious re-coding, a SAS macro is offered to facilitate the creation of dummy variables and improve productivity.

1. Introduction to Dummy Variables

Dummy variables are independent variables which take the value of either 0 or 1. Just as a "dummy" is a stand-in for a real person, in quantitative analysis, a dummy variable is a numeric stand-in for a qualitative fact or a logical proposition. For example, a model to estimate demand for electricity in a geographical area might include the average temperature, the average number of daylight hours, the total number of structure square feet, numbers of businesses, numbers of residences, and so forth. It might be more useful, however, if the model could produce appropriate results for each month or each season. Using the number of the month, such as 12 for December, would be silly, because that implies that the demand for electricity is going to be very different between December and January, which is month 1. It also implies that Winter occurs during the same months everywhere, which would preclude the use of the model for the opposite polar hemisphere. Thus, another way to represent qualitative concepts such as season, male or female, smoker or non-smoker, etc., is required for many models to make sense.

In a regression model, a dummy variable with a value of 0 will cause its coefficient to disappear from the equation. Conversely, the value of 1 causes the coefficient to function as a supplemental intercept, because of the identity property of multiplication by 1. This type of specification in a linear regression model is useful to define subsets of observations that have different intercepts and/or slopes without the creation of separate models. In logistic regression models, encoding all of the independent variables as dummy variables allows easy interpretation and calculation of the odds ratios, and increases the stability and significance of the coefficients. Examples of these results are in Section 3. In addition to the direct benefits to statistical analysis, representing information in the form of dummy variables is makes it easier to turn the model into a decision tool. Consider a risk manager who needs to assign credit limits to businesses. The age of the business is almost always significant in assessing risk. If the risk manager has to assign a different credit limit for each year in business, it becomes extremely complicated and difficult to use because some businesses are several hundred years old. Bi-variate analysis of the relationship between age of business and default usually yields a small number of groups that are far more statistically significant than each year evaluated separately.

Synonyms for dummy variables are *design variables* [Hosmer and Lemeshow, 1989] , *Boolean indicators*, and *proxies* [Kennedy, 1981]. Related concepts are *binning* [Tukey, 1977] or *ranking*, because belonging to a bin or rank could be formulated into a dummy variable. Bins or ranks can also function as *sets* and dummy variables can represent non-probabilistic set membership. Set theory is usually explained in texts on computer science or symbolic logic. See [Arbib, et. al., 1981] or [MacLane, 1986].

Dummy variables based on set membership can help when there are too few observations, and thus, degrees of freedom, to have a dummy variable for every category or some categories are too rare to be statistically significant. Dummy variables can represent mixed or combined categories using logical operations, such as:

a. a business in the wholesale ***or*** retail trades;
b. a business in the retail trades ***and*** that is less than 3 years old;
c. a business that has had a prior bankruptcy or payments placed for collection, but not both (***exclusive or***).

In a., two categories from the same variable, industry group are combined, using a logical ***or***. Two categories, industry and age of business are combined using the logical ***and*** operator. The ***exclusive or (XOR)*** operator is not part of many programming languages, including the SAS lanaguage. However, the discussion of Boolean networks in Section 4 includes a programmable definition of XOR.

The four applications of dummy variables discussed here are: 1) regime switching (or seasonality); 2) categorical representation; 3) interval level representation, and, 4) Boolean operators. The rest of this article is organized around answering the following questions: (Section 2) What is the information content of dummy variables and how is it measured?; (Section 3) How can dummy variables add predictive power and stability to traditional regression analysis?; (Section 4) How are dummy variables used in non-parametric analysis and dynamic systems?; and, (Section 5) How can developers use the SAS® language to make dummy variable coding easy? Section 6 is a summary.

2. An Information Theoretic Interpretation of the Statistical Properties of Dummy Variables

Any definition of any dummy variable implies a logical proposition with a value of *true* or *false*, a statement of fact, and the respective information value of that fact. Here are some typical examples of facts about businesses, followed by hypothetical variable names, that can be represented by dummy variables:

a. Business is at least 3 years old and less than 8 years old. (BUSAGE2);
b. Business has experienced a prior bankruptcy (BNKRPIND);
c. Business is in the top quartile in its industry with respect to its Quick Ratio (TOPQUICK);
d. Business is a retailer of Children's Apparel (CHILDAPP);
e. Business is located in the Northeast Region (NEREGN).

As dummy variables, these five variables would have the value of 1 if any statement is true, and 0 if it is false. The creation of each variable requires considerable pre-processing, with TOPQUICK requiring the most complicated processing, because, at some point, population norms for the quick ratio would have to be established to determine quartile breaks. Variable BUSAGE2 just needs the current year and the year the business started; BNKRPIND needs bankruptcy history on the case; CHILDAPP needs the SIC (Standard Industrial Classification) code; and NEREGN needs the state. The impact of these variables on further analysis depends on the application. For example, BUSAGE2 might be a derogatory indicator for credit risk but a positive indicator for mail-order response.

The information value of these variables depends on the overall proportion of observations having these dummy variables containing ones. The mean, μ_d, of a dummy variable is always in the interval [0,1], and represents the proportion, or percentage of cases that have a value of 1 for that variable. Therefore, *it is also the probability that a 1 will be observed*. It is possible to calculate the variance and standard deviation, $_d$, of a dummy variable, but these moments do not have the same meaning as those for continuous-valued data. This is because, if μ_d is known for a dummy variable, so is $_d$ because there are only two possible $(x - \mu_d)$ values. The distribution of any dummy variable can be classified as a Binomial distribution of *n* Bernoulli trials. Some helpful tables on distributions and their moments are in Appendix B of [Mood, et. al., 1977] . The long expression for calculating the Standard Deviation is $((\mu_d (1 - \mu_d)^2) + (1 - \mu_d)(0 - \mu_d)^2))^{½}$. Sometimes statistics texts refer to (1- p) as q, and the standard deviation reduces to $(pq)^{½}$.

What is the information content of a dummy variable? If $\mu_d = 1$ or $\mu_d = 0$, there is no uncertainty - an observer will know what to expect 100% of the time. Therefore, there is no benefit in further observation nor will this variable be significant in prediction, estimation, or detection of any other information. As μ_d moves up or down to 0.5, the information content increases, because there is less certainty about the value of the variable. This is discussed further with more examples in [Garavaglia 1994].

A set of dummy variables can also be thought of as a string of *bits* (a common name for *binary digits* in computer science). One of the roles of basic Information Theory [Shannon and Weaver, 1948] is to provide a methodology for determining how many bits are needed to represent specific information which will be transmitted over a channel where noise may interfere with the transmission. The term *entropy* is the measure of information in a message as a function of the amount of uncertainty

as to what is in the message. The formula for entropy is

$H = - \sum p_i \log p_i$, where p_i is the probability of a state i of a system, and H is the traditional symbol for entropy. An example of a system is a set of dummy variables. In the special case of one dummy variable:

$$H = - (p \log p + (1-p) \log (1-p)).$$

Figure 1 shows the relationship between the standard deviation and entropy for one dummy variable: they both peak at $\mu_d = 0.5$.

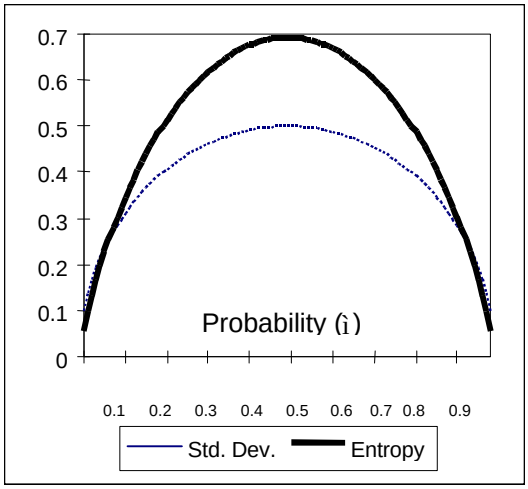

Figure 1 - Entropy and Standard Deviation

3. Impact on Regression Analysis: Two Examples - Linear and Logistic

In this section, the general use of dummy variables in linear and logistic regression are covered in the context of being part of the continuum from basic signal processing to non-parametric methods to dynamical systems. There are many additional considerations and the interested reader is advised to consult the references.

Suppose we are trying to determine the effects of research and development (RnD) and advertising (ADV) on a firm's profit (PFT). If data is available for a number of years, we can try linear regression and other techniques to determine if there is any functional form underlying the relationship between R&D and advertising. When observations span a number of time periods, varying outside factors may influence the results for some portion of the time span. In this example, during one period, the company's management was extremely enthusiastic about R&D and supported higher expenditures, and during another period a different management regime supported a higher advertising budget. At all other times, there were no unusual resource allocations. Understanding the true relationship requires modeling to differentiate these two regimes ("R&D Boosters" versus "Advertising Boosters") from the "control" or prevailing regime.

A sample of the data for this example is in Table 1. The data were artificially generated to facilitate the discussion. The underlying relationship for the "control" regime is:

$$PFT = -20 + 0.2\ RnD + 0.5\ ADV. \qquad (1)$$

Some random noise was added to the data to create some small error terms. During the years 1970-1975, the "R&D Booster" regime added **$1000** per year per observation, and, during the years 1990-1998, the "Advertising Booster" regime added **$2,500** per year per observation. Using SAS® PROC REG, the simple linear regression of Profits on R&D and Advertising yielded the following parameter estimates:

$$PFT = 399.324522 + 0.112988\ RnD + 0.309778\ ADV. \qquad (2)$$

The goodness of fit measures for all PROC REG examples are in Table 2. Adding a dummy variable for each non-control regime means that the R&D regime dummy (RDDMY) would have a value of 1 for the years 1970-1975 and 0, otherwise, and the Advertising regime dummy (ADDMY) would have a value of 1 for the years 1990-1998 and 0, otherwise. The new set of estimators is:

$$PFT = 280.125098 + 0.131938\ RnD - 277.404758\ RDDMY + 0.402229\ ADV - 945.677009\ ADDMY \qquad (3)$$

The effect of the dummy variables is to create two alternate intercepts representing the respective investment boosts during the two regimes. Note that, when the dummies were added, the goodness of fit statistics improved. Separate models were estimated for each of the three regimes, with the following results:

Control: $-19.735466 + 0.200125\ RnD + 0.499943\ ADV \qquad (4)$

R&D: $-19.930811 + 0.100087\ RnD + 0.499243\ ADV \qquad (5)$

Advertising: $-19.501744 + 0.199803\ RnD + 0.300014\ ADV. \qquad (6)$

Year	R_and_D	Advert.	Profits
1946	100	250	124
1947	149	379	198
1948	101	817	410
1949	280	987	530
1950	304	1288	686

Data from R&D Boosters Regime			
1970	1000	1352	755
1971	2789	2271	1393
1972	4825	3096	2009
1973	9298	1985	1901
1974	3915	1657	1199
1975	8799	1095	1408
Data from Advertising Boosters Regime			
1994	154	4863	1471
1995	266	4896	1503
1996	586	12201	3756
1997	1254	5366	1842
1998	1243	10514	3384

Table 1 - Sample of Regression Data

Note that (5) underestimates the R&D coefficient and (6) underestimates the Advertising coefficient to a greater degree than in (3). The dummy variables provide valuable information about the existence of alternate regimes. A more detailed example using a dummy variable for the World War II years is covered in [Judge, et. al., 1988], which also describes how to use dummy variables to create alternative slopes as well as intercepts.

Equ#	DF	CV	R^2	Max(Prob>|T|)
(2)	52	22.71	0.8374	0.0001
(3)	52	13.43	0.9454	0.0390
(4)	5	0.03952	1.0000	0.0002
(5)	8	0.0644	1.0000	0.0001
(6)	37	0.07318	1.0000	0.0001

Table 2 - Some PROC REG Goodness-of-Fit Meas.

The principles behind using dummy variables in logistic regression are similar, with regard to the design of the regime-switching. However, the exact interpretation of the coefficients now involves the calculation of the odds ratio. With a dummy variable's coefficient β_d , the odds ratio is simply $\exp(\beta_d)$. The odds ratio of a real-valued variable's coefficient β_c , is $\exp(c\beta_c)$, which makes it dependent on the real-valued variable itself and non-linear. This non-linearity makes the resulting model difficult to interpret. Creating a logistic regression model using exclusively dummy variables provides 3 distinct advantages:

1. The odds are easier to calculate and understand for each predictor
2. The binary logic of the predictors is more consistent with the binary outcome and decision making
3. The use of dummy variables to represent intervals or levels tends to increase the likelihood of events, resulting in a generally more powerful and stable model.

Advantages 1. and 2. allow the statement to be made: "if ***x*** is true the odds of ***y*** are ***z*** , all other factors held constant." The advantage of 3. is that the coefficients and their corresponding odds ratios produce more useable models, although there is some loss in the goodness-of-fit measures. However, with the loss of detail, ranking measures such as the concordance of predicted probabilities with observed responses may suffer.

Intuitively, increasing the overall probability of an observation, by grouping an interval of values into a single dummy variable should increase the value, significance and contribution of the variable. Good real world examples that demonstrate this from both a statistical goodness-of-fit standpoint and show usable results, however, are difficult to find, and credible simulated data is difficult to produce. As a compromise, a real-world business dataset from Dun & Bradstreet's database will show the result of representing the age of business as a dummy variable versus a continuous-valued variable with less dramatic but measurable contrasts. The following example is a regression of payment behavior (*prompt* or *not prompt*) on age of business and industry group dummies (see Table 4). In general, the older the business, the more likely it is to pay invoices promptly. Relevant data from the SAS® PROC LOGISTIC is in Table 3. The first regression uses the continuous variable CONYRS. Although the significance is high, the odds ratio is about even, and really doesn't convey any meaning. The second regression uses the dummy variable OLDER, having the value 1 if the company is at least 26 years old. The odds ratio shows that an older company has a much better odds of prompt payment behavior. An additional example of the difference between continuous and dummy independent variables in logistic regression can be found in Hosmer and Lemeshow (1984).

Another way to use independent dummy variables in linear or logistic regression is to represent the continuous variable in a set of levels. For age-of-business in the above example, a number

Variable	Parameter Estimate	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate	Odds Ratio
Regression #1 - Continuous-valued Age of Business					
INTERCPT	1.1690	1170.0722	0.0001	.	.
CONYRS	0.0151	84.8774	0.0001	0.121006	1.015
SICMANF	0.3304	6.0482	0.0139	0.030267	1.391
SICSVCS	0.2700	3.9437	0.0470	0.023887	1.310
SICWHOL	0.3683	30.8717	0.0001	0.067973	1.445
Regression #2 Single Dummy Variable for Age of Business >= 26					
INTERCPT	1.2399	1390.8279	0.0001	.	.
OLDER	0.2934	50.2675	0.0001	0.077388	1.341
SICMANF	0.3449	6.6145	0.0101	0.031598	1.412
SICSVCS	0.2763	4.1436	0.0418	0.024449	1.318
SICWHOL	0.3661	30.5663	0.0001	0.067575	1.442
Regression #3 Interval Level Dummy Variables for Age of Business					
INTERCPT	1.2390	1389.0309	0.0001	.	.
CONBKT4	0.1431	7.2738	0.0070	0.032935	1.154
CONBKT5	0.2413	19.6529	0.0001	0.055018	1.273
CONBKT6	0.5466	82.4710	0.0001	0.120895	1.727
SICMANF	0.3445	6.5847	0.0103	0.031558	1.411
SICSVCS	0.2809	4.2738	0.0387	0.024856	1.324
SICWHOL	0.3737	31.7602	0.0001	0.068977	1.453

Akaike Information Criterion

Regression	AIC: Intercept Only	Int. + Covariates	Reduction in AIC
Regression #1	15582.908	15455.222	127.686
Regression #2	15582.908	15500.191	82.717
Regression #3	15582.908	15464.222	118.686

Table 3 - Selected PROC LOGISTIC Output

of dummy variables were created to signify separate age ranges, namely:

CONBKT1 = 0 to 2 years
CONBKT2 = 3 to 7 years
CONBKT3 = 8 to 15 years
CONBKT4 = 16 to 20 years
CONBKT5 = 21 to 25 years
CONBKT6 = 26+ years.

Logically, a monotonic relationship is expected, i. e, the older the company, the lower the risk. Two caveats: 1) care must be taken not to overlap values, and 2) one dummy variable must be excluded from the regression if a constant or intercept is estimated to prevent less than full rank matrices (this example excluded CONBKT1).

The results of this regression in Table 3 show that three categories are significant, and the older the group, the stronger the coefficient and the better the odds of prompt payment. This technique is used as common practice in developing credit scoring models, because it provides more discrimination for rank ordering of risk and a useable odds ratio.

4. From Dummy Variables to Genetic Algorithms, and Neural Networks, and Boolean Networks

Data that is represented entirely with dummy variables opens up opportunities for new types of modeling and quantitative models of dynamical systems such as financial markets and also non-quantitative domains, such as social behavior. These modeling techniques are non-parametric and the models are usually developed using iterative methods. One common thread in genetic algorithms, neural networks, and Boolean networks is that they imitate, on a very simplistic scale, the biological processes of adaptation and evolution and the characteristics of distributed information and parallelism. Another interesting fact is that these three ideas are not at all new: genetic algorithms were introduced in the early 1960s by John Holland at University of Michigan; artificial neural networks can be traced back to the article by McCulloch and Pitts (1943); and, Boolean networks were introduced by Stuart A. Kauffman in the late 1960s. The benefits of these types of models are that the functional form need not be pre-defined, predictive/discrimination performance is superior in highly complex and non-linear applications, and they can be applied to solving a wide range of problems. In addition, the fundamentals of these models are extremely simple. Much of the theoretical research is involved with finding the fastest paths to the optimal state of these systems or special variants to

Variable Name	Industry Represented
SICAGRI	Agriculture, Mining, Fishing
SICCONS	Construction
SICMANF	Manufacturing
SICTRCM	Transportation, Communications Utilities
SICWHOL	Wholesalers
SICRETL	Retail
SICFNCL	Financial Services
SICSVCS	Business and Personal Services

Table 4 - Industry Group Dummy Variables

solve specific problems. A frequent criticism is that often the only measure of efficacy is performance and reliable goodness-of-fit measures are not available.

In Genetic Algorithms (GAs) sets of binary digits called *strings* undergo the "genetic" processes of reproduction, crossover, and mutation as if they were genes in chromosomes. Artificial chromosomes that are more "fit" have a higher probability of survival in the population.

The business data example from the previous section will be used to illustrate GAs. Suppose we know something about business "fitness" in that older companies and certain industries are more desirable credit risks. The goal is to discover and select companies with a mix of desirable characteristics. The available data is the set of 6 age categories (CONBKT1-CONBKT6) and the industry indicators (see Table 4). Instead of looking at the promptness performance indicator for the fitness measure, the fitness algorithm is:

```
fitness =
(1 * conbkt1)+(2 * conbkt2)+ (4 * conbkt3)+(8 * conbkt4)+(16 *
conbkt5)+
(32 * conbkt6) + 2(sicmanf + sicsvcs + sicwhol)
- (sicagri + siccons + sictrcm + sicfncl + sicretl);
```

This algorithm gives the older categories more points with a maximum of 32 points, the better industries group 2 points, and subtracts 1 point for the weaker industries. Thus, the minimum fitness score is 0 and the maximum fitness score is 34 (the best age category = 32 points plus 2 points for a favorable industry group).

The population is 11,551 cases, all fitness scored. Table 5 has the distribution of fitness scores, and each fitness group's relative contribution to the total fitness of the population, which is defined as the weighted sum of all the possible fitness scores. The iterative process first randomly selects candidates for reproduction according to the fitness contribution, e. g., cases in the score of 31 group have the highest likelihood of being chosen, initially.

Fitness	No. Obs.	Percent	Fitness%
0	268	2.3	0.00%
1	1237	10.7	0.55%
2	246	2.1	1.09%
3	1638	14.2	1.64%
4	249	2.2	2.19%
5	503	4.4	2.73%
7	1702	14.7	3.83%
8	336	2.9	4.37%
9	709	6.1	4.92%
15	1610	13.9	8.20%
16	257	2.2	8.74%
17	562	4.9	9.29%
31	1456	12.6	16.94%
32	247	2.1	17.49%
33	531	4.6	18.03%
183	**11551**	**99.9**	**100.00%**

Table 5 - Genetic Algorithm Fitness

Suppose a random draw process produces this "happy couple."

{0,0,0,0,0,0,0,1,0,0,0,0,0,1,0} = a business at least 26 years old in the retail industry (score = 31)
{0,0,0,0,0,0,1,0,0,0,0,0,1,0,0} = a business in the 20-25 year group in the wholesale industry (score = 18)

The process of producing offspring involves taking a substring of genes from each parent, and creating two new strings each with a portion of each parent.

The crossover point (see vertical bar) in most applications is selected randomly, but because two major characteristics are represented, the crossover point will be at the break between the age and industry groups. Thus the new members of the population are:

{0,0,0,0,0,0,0,1, **|** 0,0,0,0,1,0,0} = a business at least 26 years old in the wholesale industry (score = 34)
{0,0,0,0,0,0,1,0, **|** 0,0,0,0,0,1,0} = a business in the 20-25 year group in the retail industry (score = 15)

Although the average fitness of the two offspring is the same as for the parents, the chances of a best fitness case being selected for the next generation have now improved slightly. An additional elementary operation that could be performed in creating a new generation is mutation, which randomly selects an element in the string and reverses it. Continuing in this manner, it would take many generations to evolve into the optimal population. One technique for "cutting to the chase" and boosting the selection process is to select according to templates or "schemata." For

example, since it is known that the oldest business categories produce the highest fitness levels, a selection template of {0,0,0,*,*,*,*,*,*,*,*,*,*}, where the *s are *wild cards*, would greatly increase the likelihood of a higher fitness candidate, because only older businesses would filter through.

In some neural network applications, continuous valued variables are better choices, because the number of hidden units available can capture the information provided by the data. However, representing information as vectors of dummy variables expands the selection of neural network paradigms. Almost any neural network model, from Adaptive Resonance Theory (ART1) to the Self-Organizing Map can accept binary inputs, but only a subset of these can accept real-valued inputs. The precise impact of the data representation method in any given neural network paradigm depends on many factors, including the number of hidden nodes, the distribution of the values, and so forth. Readers who need general and introductory information on neural networks should consult the information on the Neural Networks FAQ Web Site at ftp://ftp.sas.com/pub/neural/FAQ.html. For those who are seeking theoretical foundations and precise methodologies, [White, et. al., 1992] and [Golden, 1996] are good choices.

In a feed-forward neural network with supervised learning, the neural network is trained using a desired outcome for each training case. Given an example with a large number of training cases, which allow enough degrees of freedom for a large number of hidden units, a continuous-valued variable might be expected to provide better performance because the hidden units could "model" different complex regions of the data space. As a test, the business data set was used to train two feed-forward networks, one using the groups CONBKT1-CONBKT6 and the other using the variable CONYRS. With 9 hidden units, the neural network with the 6 groups generated 136 connection weights, and the neural network with CONYRS generated only 91 hidden units (both networks include a bias input unit). Then, to "level the playing field," the network with CONYRS was given enough additional hidden units to generate about the same number of total connection weights (141). The results of these three networks from a validation data set were essentially, a "photo finish." All three had the identical misclassification rate (about 19.2%), and about the same Mean Squared Error (about 0.156). The flexibility and "un-messy" dummy variables did not cause the predictive power of the neural network to suffer.

In a clustering neural network, such as the Self-Organizing Map (SOM), dummy variable inputs make the resulting clustering nodes easier to interpret, and help to avoid unnecessary one-observation nodes. In business data especially, extreme values do not necessarily correlate with singular behavior, e. g., a company that is 250 years old often behaves like a company that is 50 years old. The weight vectors can be interpreted as the proportion of cases from each group in the cluster. Examples of using dummy variables in Self-Organizing Maps are in [Garavaglia and Sharma, 1996].

A Boolean network is a type of discrete dynamical system composed of binary-valued nodes which are interconnected, in that all nodes both send and receive information. The system changes according to logical operations on nodes that are input to other nodes in the system. At any point in time, the Boolean network represents a state of a dynamic system and a change from one state to the next occurs when all binary nodes are simultaneously updated. Here, in Figure 2, is an example of a simple Boolean network with five nodes, each of which is updated according to a logical operation on two connected inputs. These types of networks grow in complexity as they grow in size, but they also develop one or more *attractors,* which are states or sets of states that they achieve regularly without outside influences.

The Boolean network in Figure 2 updates its states as follows:

A = (B *and* D)
B = (A *or* C)
C = ((A *and* (*not* E)) *or* ((*not* A) *and* E)))
(i. e., ***exclusive OR***)
D = (C *or* E)
E = (B *or* D)

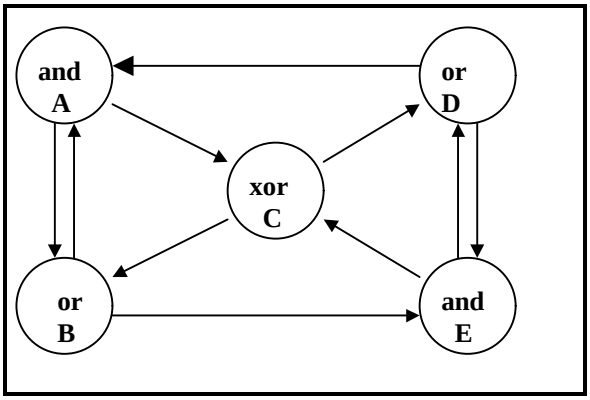

Figure 2 - A Simple Boolean Network

This simple dynamical system will develop attractors very quickly; the exact nature of the attractors depends on the initial state of the system. Figure 3 shows the first 16 states of the system after being initialized in state 1 to A=1, B = 1, C = 0, D = 1, E = 0. What can Boolean networks represent?

This simple 5 node network could be 5 commodity traders, 5 voters, 5 weather systems, or any other dynamic environment in which there is circular influence among parts of the system.

A last topic in the representation of data with dummy variables is *thermometer encoding.* The categorical level coding of the age of business variable CONYRS into six dummy variables is called 1-of-N encoding. Another way this information could have been encoded is Table 6 This type of encoding is used almost exclusively only in neural networks, and is best suited to modeling analog data, such as color intensity. See [Harnad, et. al. ,1991] for an example.

	CONBKT	CONBKT	CONBKT	CONBKT	CONBKT	CONBKT
Value	1	2	3	4	5	6
Age						
0 to 2 yrs	1	0	0	0	0	0
3 to 7 yrs	1	1	0	0	0	0
8 to 15 yrs	1	1	1	0	0	0
16 to 20 yrs	1	1	1	1	0	0
21 to 25 yrs	1	1	1	1	1	0
26+ yrs.	1	1	1	1	1	1

Table 6 - Thermometer Encoding

5. Using a SAS Macro to Create Dummy Variables from Raw Data

Recoding a categorical variable into individual dummy variables can get tedious quickly if there are more than a few categories. In addition, the process is error prone. Realistically, only a subset of the categories may be statistically significant, but all must be analyzed in the context of their final representation in the resulting model. The SAS® Language provides a meta-coding capability within its macro-language, providing the tools for "logic about logic," code generation, and conditional execution of statements. Another example of code generation is in [Liberatore, 1996].

For a real-valued variable, when the number of levels are not known prior to analysis, a "select clause shell" such as the coding example below, is handy. Copying lines of code and string substitutions can be used to change this code as necessary.

```
level1 = 0;
level2 = 0;
level3 = 0;
select;
when ( a <= var < b)  level1 = 1;
when ( b <= var < c)  level2 = 1;
when ( c <= var < d)  level3 = 1;
otherwise;
end;
```

The following macro, *dmycode*, will produce code to turn a categorical variable with *n* categories into *n* dummy variables with the category value as part of the variable name (the length may be truncated, if necessary). It is presented with the example of generating code to create dummy variables for each 2-digit SIC (Standard Industrial Classification) code. In this macro and the one that follows, a dummy variable name is constructed from the value of the category within the original variable, so that the dummy variable is easily recognizable.

```
option nosymbolgen mlogic mprint
obs=9999;
libname risk '/nesug98/research';
filename out
'/nesug98/research/testsic.out';
 data sicwork;set risk.sicwork;
sic2=int(sic4/100);
;
/*
 MACRO PARAMETERS :
 dsn = input dataset name ,
 var = variable to be categorized ,
 prefix = categorical variable prefix ,
 flat = flatfile name with code  (
referenced in file name statement)
 */

%macro dmycode   ( dsn   = ,
                   var  = ,
                   prefix = ,
                   flat =   );

proc summary data = &dsn nway ;
     class &var ;
     output  out = x(keep=&var )  ;
proc print ;
*;
data _null_ ;
 set x nobs=totx end=last;
 if last then   call symput ( 'tot',
trim(left(put( totx, best. ) ) ) ) ;
  call symput ( 'z' || trim ( left (
put ( _n_,     best. ) ) ),trim ( left
( &var ) ) ) ;
 data _null_;
```

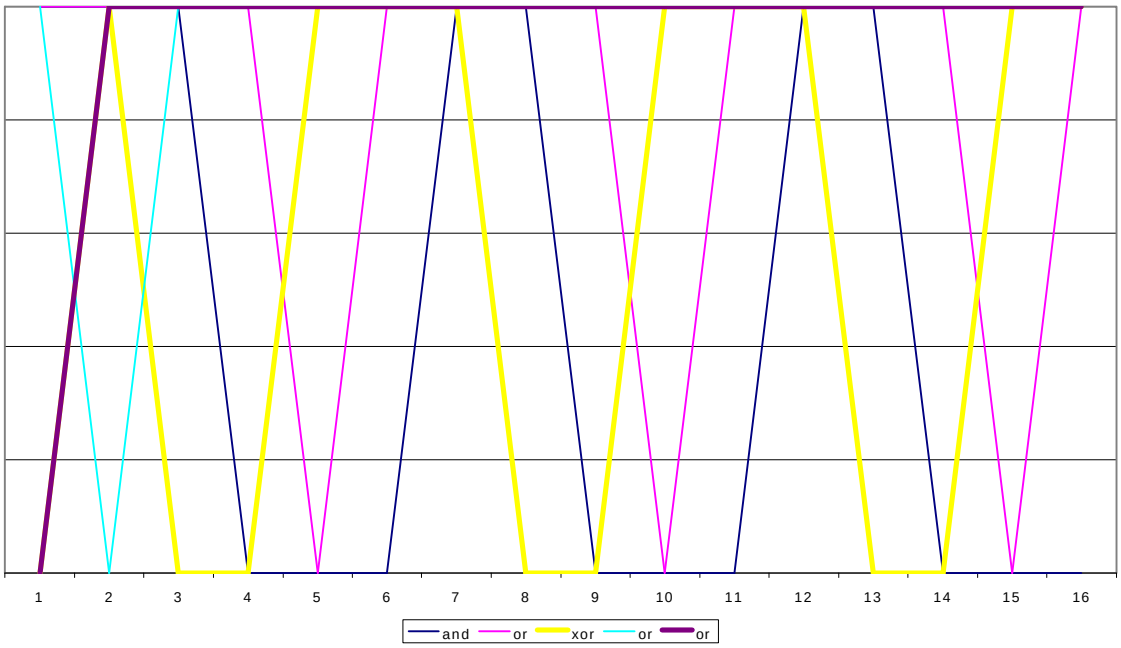

Figure 3 - First 16 States of 5-Node Boolean Network

```
file &flat;
  %do i=1 %to &tot ;
  put  " &prefix&&z&i =0;" ;
  %end;
   put "SELECT ;";
  %do i=1 %to &tot ;
   put "  when ( &var =  &&z&i )
&prefix&&z&i =1 ;";
  %end;
  put " otherwise sic_oth=1;";
  put " end;" ;
run ;
%mend dmycode ;
  %dmycode ( dsn = sicwork, var = sic2,
prefix = sic_ , flat =out ) ;
 run;quit;
```

This macro, *dummy*, will create dummy variables in a file for each category in a categorical data element, and also uses 2-digit SIC codes as an example.

```
option nosymbolgen mlogic obs=99999;
libname risk '/nesug98/research';
 libname dat '/nesug98/research';
 data sicwork;set risk.sicwork;
sic2=int(sic4/100);  format sic2 z2. ;
;
/*
 MACRO PARAMETERS :
 dsn = input dataset name ,
 var = variable to be categorized ,
 prefix = categorical variable prefix ,
 */

%macro DUMMY    ( dsn    = ,
                  var    = ,
                  prefix = ) ;

proc summary data = &dsn nway ;
     class &var ;
     output out = x ( keep = &var ) ;
proc print ;
*;
data _null_ ;
 set x nobs=last  ;
  if _n_ = 1 then  call symput ( 'num',
trim(left(put( last, best. ) ) ) ) ;
  call symput ( 'c' || trim ( left (
put ( _n_,     best. ) ) ),trim ( left
( &var ) ) ) ;
run ;

      data &dsn   ;
       set &dsn  nobs=last;
      array ct  ( &num ) %do k=1 %to
&num ;
                              &prefix&&c&k
                            %end ; ;
      %do i = 1 %to &num ;
       select;
        when (&var="&&c&i" ) ct(&i)=1;
        otherwise ct(&i)=0;
       end;
      %end;
run ;
%mend Dummy ;
   %Dummy ( dsn = sicwork , var = sic2,
prefix = sic_  ) ;

proc print ;
run;
quit;
```

6. Summary

Dummy variables play an important role in the analysis of data, whether they are real-valued variables, categorical data, or analog signals. The extreme case of representing all the variables (independent and dependent) as dummy variables provides a high degree of flexibility in selecting a modeling methodology. In addition to this benefit of flexibility, the elementary statistics (e. g., mean and standard deviation) for dummy variables have interpretations for probabilistic reasoning, information theory, set relations, and symbolic logic. Whether the analytical technique is traditional or experimental, highly complex information structures can be represented by dummy variables. Examples presented included multiple regimes, business behavior, and dynamical systems. There are no hard boundaries between the relationships of dummy variables in quantative analsyis, sets and logic, and the computer science concept of data representation in bits. The intelligent use of dummy variables usually makes the resulting application easier to implement, use, and interpret.

References

Arbib, Michael A., A. J. Kfoury and Robert N. Moll. 1981. A Basis for Theoretical Computer Science. Springer-Verlag. New York.

Garavaglia, Susan. 1994. An Information Theoretic Re-Interpretation of the Self-Organizing Map With Standard Scaled Dummy Variables. *World Congress on Neural Networks '94 Proceedings*. INNS Press. San Diego, CA.

Garavaglia, Susan and Asha Sharma. 1996. Statistical Analysis of Self-Organizing Maps. NESUG '96 Proceedings.

Goldberg, David E. 1989. Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley. Reading, MA.

Golden, Richard M. 1996. Mathematical Methods for Neural Network Analysis and Design. The MIT Press. Cambridge, MA.

Harnad, Stevan, S. J. Hanson, and J. Lubin. 1991. Categorical Perception and the Evolution of Supervised Learning in Neural Nets. Working Papers of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology. Current as of July 1, 1998 at http://www.cogsci.soton.ac.uk/~harnad/Papers/Harnad/harnad91.cpnets.html.

Holland, John H. 1992. Adaptation in Natural and Artificial Systems. The MIT Press. Cambridge, MA.

Judge, George G., R. Carter Hill, William E. Griffiths, Helmut Lutkepohl, and Tsoung-Chao Lee. 1988. Introduction to the Theory and Practice of Econometrics. John Wiley & Sons, Inc. New York.

Kauffman, Stuart A. 1993. The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press. New York.

Kennedy, Peter. 1989. A Guide to Econometrics. Second Edition. The MIT Press. Cambridge, MA.

Liberatore, Peter. 1996. Too Many Variables, Too Little Time: A Macro Solution. NESUG '96 Proceedings.

MacLane, Saunders. 1986. Mathematics Form and Function. Springer-Verlag. New York.

Maddala, G. S. 1983. Limited Dependent and Qualitative Variables in Econometrics. Cambridge U. Press. Cambridge

McCulloch Warren S. and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. reprinted in Neurocomputing: Foundations of Research. 1988. James A. Anderson and Edward Rosenfeld, eds. The MIT Press. Cambridge, MA.

Mood, Alexander M., Franklin A. Graybill, and Duane C. Boes. 1974. Introduction to the Theory of Statistics. Third Edition. McGraw-Hill, Inc. New York.

Shannon, Claude E. and Warren Weaver. 1948. The Mathematical Theory of Communication. U. of Illinois Press. Urbana, IL.

Tukey, John W. 1977. Exploratory Data Analysis. Addison-Wesley. Reading, MA.

White, Halbert, Jr, 1992. Artificial Neural Networks: Learning and Approximation Theory. Blackwell's. Oxford.