

Binning of Predictors for the Cumulative Logit Model

Bruce Lund, Magnify Analytic Solutions, Detroit MI, Wilmington DE, Charlotte NC

ABSTRACT

Binning of a predictor is widely used when preparing a predictor for use in binary logistic regression. In this paper the practice of binning is extended to the cumulative logit model. Here the target has more than 2 levels. A binning algorithm and SAS® code is presented. The user may choose to bin a predictor X so that the ordering of X is maintained. In this case, for each k for $2 \leq k \leq L$ (where L = number of levels of X), the algorithm can find the k-bin solution which is optimal with respect to log-likelihood or to a generalized information value (IV). The optimal solution is found by complete enumeration of all solutions. Additionally, the algorithm can find if there is an optimal k-bin solution which is monotonic versus the empirical cumulative logits of the target. Alternatively, the user may choose to bin X without regard to the ordering of X. In this case the algorithm maximizes log-likelihood or generalized IV at each step in the binning process but an optimal k-bin solution is not guaranteed. A generalization of weight of evidence (WOE) is defined and SAS code for WOE coding is generated. Examples are presented.

INTRODUCTION

Binary logistic regression models are widely used in CRM (customer relationship management) and credit risk modeling. In these models it is common to consider nominal, ordinal, or discrete¹ (NOD) predictors for use in the model. These predictors may appear in a CLASS statement so that the levels of the predictor appear as dummy variables in the model. Alternatively, these NOD predictors may be re-coded by weight of evidence (WOE) coding.

Ordinal logistic regression refers to logistic models where the target has more than 2 values and these values have an ordering. For example, ordinal logistic regression is applied when fitting a model to a target which is a satisfaction rating (e.g. good, fair, poor). Here, the scale is inherently non-interval. In other cases, the target could be a count or a truncated count (e.g. children in household: 0, 1, 2, 3+). The cumulative logit model is one formulation of the ordinal logistic model.² The focus of this paper is the cumulative logit model.

In either case, binary logistic or cumulative logit regression, it is important that NOD predictors be “binned”. Binning is the process of reducing the number of levels of a NOD predictor to achieve parsimony while preserving, as much as possible, the predictive power of the predictor. SAS macros for “optimal” binning of NOD predictors are discussed in the paper.

Also in this paper the idea of WOE coding of a NOD predictor is extended to the cumulative logit model. The comparative benefits of WOE coding and dummy-variable coding are discussed.

See Lund (2017a) for an alternative coverage of these topics.³

TRANSFORMING A PREDICTOR BY WOE FOR BINARY LOGISTIC REGRESSION

A NOD predictor C (character or numeric) with L levels can be entered into a binary logistic regression model with a CLASS statement or as a collection of dummy variables.⁴ Typically, L is 15 or less.

¹ A discrete predictor is a numeric predictor with only “few values”. Often these values are counts. The designation of “few” is subjective. It is used here to distinguish discrete from continuous (interval) predictors with “many values”.

² Allison (2012, Chapter 6) gives an introduction to cumulative logit model. See also Agresti (2010) and Hosmer, Lemeshow, Sturdivant (2013). These references do not discuss in any detail a generalization of cumulative logit called partial proportional odds (PPO). The PPO model will appear later in this paper.

³ This paper covers many of the same topics as in Lund (2017a) but the emphasis in this paper is on binning algorithms for the cumulative logit model.

⁴ “CLASS C;” creates a coefficient in the model for each of L-1 of the L levels. The modeler’s choice of “reference level coding” determines how the Lth level enters into the calculation of the model scores. See SAS/STAT(R) 14.1 User’s Guide (2015), LOGISTIC procedure, CLASS statement.

These two models produce exactly the same probabilities.

PROC LOGISTIC; **CLASS** C; **MODEL** Y = C <and other predictors>;

or

PROC LOGISTIC; **MODEL** Y = C_dum_k <and other predictors>; where k = 1 to L-1

An alternative to CLASS / DUMMY coding of C is the weight of evidence (WOE) transformation of C. It is notationally convenient to use G_k to refer to counts of Y = 1 and B_k to refer to counts of Y = 0 when C = "C_k". Let $G = \sum_k G_k$. Then g_k is defined as $g_k = G_k / G$. Similarly, for $B = \sum_k B_k$ and $b_k = B_k / B$.

For the predictor C and target Y of Table 1 the weight of evidence transformation of C is given by the right-most column in the table.

C	Y = 0 "B _k "	Y = 1 "G _k "	Col % Y=0 "b _k "	Col % Y=1 "g _k "	WOE= Log(g _k /b _k)
C1	2	1	0.250	0.125	-0.69315
C2	1	1	0.125	0.125	0.00000
C3	5	6	0.625	0.750	0.18232

Table 1. Weight of Evidence Transformation for Binary Logistic Regression

The formula for the transformation is: If C = "C_k" then $C_woe = \log(g_k / b_k)$ for k = 1 to L where $g_k, b_k > 0$. WOE coding is preceded by binning of the levels of predictor C, a topic to be discussed in a later section.

A PROPERTY OF A LOGISTIC MODEL WITH ONE WEIGHT OF EVIDENCE PREDICTOR

When a single weight of evidence variable C_woe appears in the logistic model, as shown,

PROC LOGISTIC DATA = <> DESCENDING; **MODEL** Y = C_woe;

then the slope coefficient equals 1 and the intercept is $\log(G/B)$.⁵

This property of a WOE predictor is verified by substituting the solution $\alpha = \log(G/B)$ and $\beta = 1$ into the maximum likelihood equations to show that a solution has been found. This solution is the global maximum since the log-likelihood function has a unique extreme point and this point is a maximum (ignoring the degenerate cases given by data sets having quasi-complete and complete separation). See Albert and Anderson (1984, Theorem 3).

INFORMATION VALUE OF PREDICTOR C FOR TARGET Y

An often-used measure of the predictive power of predictor C is Information Value (IV). It measures predictive power without regard to an ordering of a predictor. The right-most column of Table 2 gives the terms that are summed to obtain the IV. The range of IV is the non-negative numbers.

C	Y = 0 "B _k "	Y = 1 "G _k "	Col % Y=0 "b _k "	Col % Y=1 "g _k "	Log(g _k /b _k)	g _k - b _k	IV Terms (g _k - b _k) * Log(g _k /b _k)
C1	2	1	0.250	0.125	-0.69315	-0.125	0.08664
C2	1	1	0.125	0.125	0.00000	0	0.00000
C3	5	6	0.625	0.750	0.18232	0.125	0.02279
SUM	8	8				IV =	0.10943

Table 2. Information Value Example for Binary Logistic Regression

IV can be computed for any predictor provided none of the g_k or b_k is zero. As a formula, IV is given by:

$$IV = \sum_{k=1}^L (g_k - b_k) * \log(g_k / b_k)$$

where $L \geq 2$ and where g_k and $b_k > 0$ for all $k = 1, \dots, L$

⁵ Alternatively, to obtain the same result, the roles of 0 and 1 are reversed in the calculation of C_woe and DESCENDING is omitted in PROC LOGISTIC. This alternative is the approach taken in this paper when modeling the Cumulative Logit Model in a later section.

Note: If two levels of C are collapsed (binned together), the new value of IV is less than or equal to the old value. The new IV value is equal to the old IV value if and only if the ratios g_r / b_r and g_s / b_s are equal for levels C_r and C_s that were collapsed together.⁶

PREDICTIVE POWER OF IV FOR BINARY LOGISTIC REGRESSION

Guidelines for interpretation of values of the IV of a predictor in an applied setting are given below. These guidelines come from Siddiqi (2006, p.81). In logistic modeling applications it is unusual to see $IV \geq 0.5$.

IV Range	Interpretation
$IV < 0.02$	"Not Predictive"
$IV \text{ in } [0.02 \text{ to } 0.1)$	"Weak"
$IV \text{ in } [0.1 \text{ to } 0.3)$	"Medium"
$IV \geq 0.3$	"Strong"

Table 3. Practical Guide for Interpreting IV

THE DEFINING CHARACTERISTICS OF WEIGHT OF EVIDENCE CODING

The next step is to explore the extension of weight of evidence coding and information value to the case of the cumulative logit model.

There are two defining characteristics of the weight of evidence coding, X_{woe} , of a predictor X when the target is binary and X_{woe} is the single predictor in a logistic model. These are:

- Equality of Model (I) and Model (II):
 - PROC LOGISTIC DATA = <> DESCENDING; CLASS X; MODEL Y = X;**
 - PROC LOGISTIC DATA = <> DESCENDING; MODEL Y = X_{woe} ;**
- The values of the coefficients for Model (II): Intercept = $\text{Log}(G / B)$ and Slope = 1

GOAL: Find a definition of WOE to extend to the cumulative logit model so that the appropriate generalizations of (1) and (2) are true.

CUMULATIVE LOGIT MODEL

If the target variable in PROC LOGISTIC has more than 2 levels, PROC LOGISTIC regards the appropriate model as being the cumulative logit model with the proportional odds (PO) property. An explanation of the cumulative logit model and of the proportional odds property is given in this section.

A SIMPLIFICATION FOR THIS PAPER

In this paper all discussion of the cumulative logit model will assume the target has 3 levels. However, the concepts presented in the paper extend to targets with more than 3 levels.

DEFINITION OF CUMULATIVE LOGIT MODEL WITH PROPORTIONAL ODDS (PO)

To define the cumulative logit model with PO, the following example is given: Assume the 3 levels for the ordered target Y are A, B, C (in alphabetic order) and suppose there are 2 numeric predictors X1 and X2.⁷

Let $p_{k,j}$ = probability that the k^{th} observation has the target value $j = A, B \text{ or } C$. Let $X_{k,1}$ be the value of X1 for the k^{th} observation. Similarly, for $X_{k,2}$.

Then this cumulative logit model has 4 parameters $\alpha_A \alpha_B \beta_{X1} \beta_{X2}$ and is given via 2 response equations:

$$\begin{aligned} \text{Log}(p_{k,A} / (p_{k,B} + p_{k,C})) &= \alpha_A + \beta_{X1} * X_{k,1} + \beta_{X2} * X_{k,2} & \dots \text{ response equation } j = A \\ \text{Log}((p_{k,A} + p_{k,B}) / p_{k,C}) &= \alpha_B + \beta_{X1} * X_{k,1} + \beta_{X2} * X_{k,2} & \dots \text{ response equation } j = B \end{aligned}$$

⁶ See Lund and Brotherton (2013, p. 17) for a proof.

⁷ If a predictor X is not numeric, then the dummy variables from the coding of the levels of X appear in the right-hand-side of the response equations for $j = A$ and $j = B$.

It is noted that the coefficient β_{X1} of predictor X1 is the same in both response equations. Similarly, for predictor X2.

The “cumulative logits” are the log of the ratio of the “cumulative probability up to j” (in the ordering of the target) in the numerator to “one minus the cumulative probability up to j” in the denominator. When the target has levels A, B, C, then j goes from A to B. These cumulative logits are:

$$\begin{aligned} &\text{Log } (p_{k,A} / (p_{k,B} + p_{k,C})) \text{ for the first response equation} \\ &\text{Log } ((p_{k,A} + p_{k,B}) / p_{k,C}) \text{ for the second response equation} \end{aligned}$$

Formulas for the probabilities $p_{k,A}$, $p_{k,B}$, $p_{k,C}$ can be derived from the two response equations. To simplify the formulas, let T_k and U_k , for the k^{th} observation be defined by the equations below:

$$\begin{aligned} \text{Let } T_k &= \exp (\alpha_A + \beta_{X1} * X_{k,1} + \beta_{X2} * X_{k,2}) \\ \text{Let } U_k &= \exp (\alpha_B + \beta_{X1} * X_{k,1} + \beta_{X2} * X_{k,2}) \end{aligned}$$

Then, after algebraic manipulation, the probability equations in Table 4 are derived:

Response	Probability Formula
A	$p_{k,A} = 1 - 1/(1+T_k)$
B	$p_{k,B} = 1/(1+T_k) - 1/(1+U_k)$
C	$p_{k,C} = 1/(1+U_k)$

Table 4. Cumulative Logit Model - Equations for Probabilities

The parameters for the cumulative logit model are estimated by maximizing the log-likelihood equation in a manner similar to the binary case (Agresti 2010, p 58).

This cumulative logit model satisfies the following conditions for X1 (and the analogous conditions for X2):

Let “r” and “s” be two values of X1 and fix the value of X2. Using the probability formulas from Table 4:

$$\begin{aligned} \text{Log } \left[\frac{p_{r,A}/(p_{r,B} + p_{r,C})}{p_{s,A}/(p_{s,B} + p_{s,C})} \right] &= \text{Log } (p_{r,A} / (p_{r,B} + p_{r,C})) - \text{Log } (p_{s,A} / (p_{s,B} + p_{s,C})) = (r - s) * \beta_{X1} \dots \text{proportional odds} \\ \text{Log } \left[\frac{(p_{r,A} + p_{r,B})/p_{r,C}}{(p_{s,A} + p_{s,B})/p_{s,C}} \right] &= \text{Log } ((p_{r,A} + p_{r,B}) / p_{r,C}) - \text{Log } ((p_{s,A} + p_{s,B}) / p_{s,C}) = (r - s) * \beta_{X1} \dots \text{proportional odds} \end{aligned}$$

These equations display the “proportional odds” property. Specifically, the difference of cumulative logits at r and s is proportional to the difference (r - s). The proportional odds property is a by-product of the equality of the coefficients of predictors X1 and X2 across the cumulative logit response equations.

WOE AND INFORMATION VALUE FOR THE CUMULATIVE LOGIT MODEL

If Y has J levels, then J-1 WOE transformations are needed. WOE can be defined for the cumulative logit model without assuming proportional odds (i.e. without assuming equal predictor coefficients across response equations).

Consider an ordinal target Y with levels A, B, C and predictor X with levels 1, 2, 3, 4. Here, Y has 3 levels and, therefore, 2 weight of evidence transformations are formed. Here is the process:

Empirical cumulative logits are defined by:

$$\begin{aligned} \text{Empirical logit at } X=i \text{ for A vs } \{B, C\} &= (A_i / n_A) / ((B_i + C_i) / (n_B + n_C)) \\ \text{Empirical logit at } X=i \text{ for } \{A, B\} \text{ vs } C &= ((A_i + B_i) / (n_A + n_B)) / (C_i / n_C) \end{aligned}$$

where A_i = count of A's for $X = i$, similarly for B_i and C_i , and $n_A = \sum A_i$, $n_B = \sum B_i$, $n_C = \sum C_i$

The weight of evidence variables, X_woe1 and X_woe2 , are the log of the empirical logits:

$$\begin{aligned} X_woe1 (X=i) &= \text{LOG } [(A_i / n_A) / ((B_i + C_i) / (n_B + n_C))] \\ X_woe2 (X=i) &= \text{LOG } [((A_i + B_i) / (n_A + n_B)) / (C_i / n_C)] \end{aligned}$$

The (X, Y) values from Table 5 may be substituted into the formulas for X_woe1 and X_woe2 to verify the values in the right two columns of Table 5.

X=i	Y=			X_woe1	X_woe2
	Ai	Bi	Ci		
1	2	1	2	0.03	-0.18
2	4	3	1	0.44	1.36
3	4	1	2	0.72	0.33
4	1	2	5	-1.51	-1.10
Total	11	7	10		

Table 5. Data Set with Target Y and Predictor X and Weight of Evidence Coding of X

Although X in this example is numeric, any NOD predictor may take the role of X.

THERE IS NO MYSTERY ASSOCIATED WITH X_woe1 AND X_woe2

In the case of X_woe1 this is simply the binary weight of evidence transform for binary target {A} vs {B and C}. Similarly, X_woe2 is the binary weight of evidence transform for binary target {A and B} vs {C}.

ORDERED X AND WOE

If the NOD predictor X is ordered, then the business expectation may be that one or more of the empirical logits are monotonic with respect to X. Since weight of evidence coding is the log transform of the empirical logits, it is equivalent to see if the WOE is monotone vs. X.

Returning to Table 5, neither X_woe1 nor X_woe2 is monotonic vs X. (Here, X is assumed to be ordered.)

INFORMATION VALUE FOR THE CUMULATIVE LOGIT MODEL

There is also a natural extension of Information Value to the cumulative logit model.

Consider an ordinal target Y with levels A, B, C and predictor X with levels 1, 2, 3, 4. Here, Y has 3 levels and, therefore, 2 weight of evidence transforms are formed. Information Value is computed for the binary target A vs {B and C}. This is called IV1. Similarly, for {A and B} vs C there is IV2. The formulas are:

$$IV1 = \sum_{i=1}^4 [A_i / n_A - (B_i + C_i) / (n_B + n_C)] * \text{LOG} [(A_i / n_A) / ((B_i + C_i) / (n_B + n_C))]$$

$$IV2 = \sum_{i=1}^4 [(A_i + B_i) / (n_A + n_B) - C_i / n_C] * \text{LOG} [((A_i + B_i) / (n_A + n_B)) / (C_i / n_C)]$$

One approach to defining a single information value for predictor X is to sum IV1 and IV2 and call this TOTAL_IV. Guidelines, as given in Table 3 for binary targets, are needed in order to interpret the predictive power of values of TOTAL_IV.

The reader may check for the data of Table 5 that IV1 = 0.6760 and IV2 = 0.7902.

PO DOES NOT SUPPORT A GENERALIZATION OF WOE

Data set EXAMPLE1 is created from the counts of Table 5. The use of EXAMPLE1 will show that the cumulative logit PO model does not support the required two characteristics for a WOE predictor. To show the failure of the WOE definitions in the cumulative logit PO case, the Models (I) and (II) are considered:

(I) **PROC LOGISTIC** DATA = EXAMPLE1; **CLASS** X; **MODEL** Y = X;

(II) **PROC LOGISTIC** DATA = EXAMPLE1; **MODEL** Y = X_woe1 X_woe2;

The coefficients of Model (II) do not have the required values. This is displayed in Table 6.

Maximum Likelihood Estimates			Not Equal to:
Parameter		Estimate	
Intercept	A	-0.4870	≠ -0.4353 ... Log(n _A /(n _B +n _C))
Intercept	B	0.7067	≠ 0.5878 ... Log((n _A +n _B)/n _C)
X_Woe1		0.6368	≠ 1
X_Woe2		0.2869	≠ 1

Table 6. Results of MODEL (II) for the PO Model

The reader may verify the Models (I) and (II) do not produce the same probabilities.

PARTIAL PROPORTIONAL ODDS (PPO)

This section defines the partial proportional odds (PPO) cumulative logit model and how weight of evidence naturally extends to this setting. To describe PPO, the following example is given: Assume there are 3 levels for ordered target Y: A, B, C and there are 3 numeric predictors R, S and Z.

Let $p_{k,j}$ = probability that k^{th} observation has the target value $j = A, B$ or C

In this example the PPO Model will have 6 parameters $\alpha_A \alpha_B \beta_R \beta_S \beta_{Z,A} \beta_{Z,B}$ given in 2 equations:

$$\text{Log} (p_{k,A} / (p_{k,B} + p_{k,C})) = \alpha_A + \beta_R * R_k + \beta_S * S_k + \beta_{Z,A} * Z_k \quad \dots j = A$$

$$\text{Log} ((p_{k,A} + p_{k,B}) / p_{k,C}) = \alpha_B + \beta_R * R_k + \beta_S * S_k + \beta_{Z,B} * Z_k \quad \dots j = B$$

Here, Z has different coefficients for the 2 response equations. In general, for PPO some predictors may have coefficients with values that vary across response equations.

Formulas for probabilities $p_{k,A}$, $p_{k,B}$, $p_{k,C}$ continue to be given by Table 4 after modifications to definitions of T and U to reflect the PPO model. In unusual cases it is possible for a PPO probability to be negative.⁸

In the development of binning and weight of evidence of a single predictor for the PPO cumulative model, to be discussed below, negative probabilities will not arise as an issue. See Table 8 to follow.

WEIGHT OF EVIDENCE IN THE SETTING OF PPO CUMULATIVE LOGIT MODEL

Models (I) and (II) are modified to allow the coefficients of the predictors to depend on the cumulative logit response function. This is accomplished by adding the UNEQUALSLOPES statement.

(I) **PROC LOGISTIC DATA** = EXAMPLE1; **CLASS** X;
MODEL Y = X / **UNEQUALSLOPES** = (X);

(II) **PROC LOGISTIC DATA** = EXAMPLE1;
MODEL Y = X_woe1 X_woe2 / **UNEQUALSLOPES** = (X_woe1 X_woe2);

For data set EXAMPLE1, Models (I) and (II) are the same model (produce the same probabilities). Model (II) produces coefficients which generalize WOE coefficients from the binary case. Formulas for these coefficients are shown below:

$$\begin{aligned} \alpha_A &= \log (n_A / (n_B + n_C)) & \alpha_B &= \log ((n_A + n_B) / n_C) \\ \beta_{X_woe1,A} &= 1, \beta_{X_woe1,B} = 0; & & \dots \text{(Equation A)} \\ \beta_{X_woe2,A} &= 0, \beta_{X_woe2,B} = 1; \end{aligned}$$

where n_A is count of $Y = A$, n_B is count of $Y = B$, n_C is count of $Y = C$

The regression results from running Model (II) are given in Table 7.

Maximum Likelihood Estimates			Equal to:
Parameter		Estimate	
Intercept	A	-0.4353	$-0.4353 \dots \text{Log}(n_A/(n_B+n_C))$
Intercept	B	0.5878	$0.5878 \dots \text{Log}((n_A+n_B)/n_C)$
X_Woe1	A	1.0000	1
X_Woe1	B	-1.27E-12	0
X_Woe2	A	3.2E-10	0
X_Woe2	B	1.0000	1

Table 7. Results of MODEL (II) for the PPO Model with WOE Predictors

⁸ http://support.sas.com/documentation/cdl/en/statug/68162/HTML/default/viewer.htm#statug_logistic_examples22.htm
See note at bottom of webpage for discussion. Also see slide 47 of presentation slides by Richard Williams (2008) for a discussion and references. <https://www.stata.com/meeting/germany08/GSUG2008.pdf>

CONCLUSION REGARDING THE USAGE OF WEIGHT OF EVIDENCE PREDICTORS

In order to reproduce the two defining characteristics of the weight of evidence predictor from the binary case, the weight of evidence predictors should enter a cumulative logit model with unequalslopes.

Comments

There are degenerate $\{X, Y\}$ data sets where a cumulative logit model has no solution.⁹ Setting these cases aside, I do not have a solid mathematical proof that coefficients, as given by Equation A, always produce the maximum likelihood solution for Model (II) or that Model (I) and Model (II) are always equivalent. I am relying on verification by examples.

Using parameter values in Equation A for Model II, the probabilities when $X = r$ for target levels A, B, and C are obtained by substitution into the equations of Table 4 and the results are given in Table 8.

$$p_{r,A} = A_r / (A_r + B_r + C_r)$$

$$p_{r,B} = B_r / (A_r + B_r + C_r)$$

$$p_{r,C} = C_r / (A_r + B_r + C_r)$$

where A_r is the count of $Y = A$ when $X = r$, etc.

Table 8. Probabilities for X for Model II (and Model I)

WHEN TO USE UNEQUALSLOPES FOR A PREDICTOR

The use of unequalslopes for a predictor will unnecessarily increase the number of parameters in a cumulative logit model when the proportional odds assumption is actually valid.

The determination of whether to use unequalslopes applies to numeric predictors as well as to classification predictors. Three statistical tests for investigating the use of unequalslopes are given by Derr (2013). One of these tests, the “One-Up” test, is illustrated in the Appendix. The other tests are called “Down” and “Up”. All three are discussed in Derr (2013, p. 15). These tests, like any that use p-values, are influenced by sample size. A large sample may lead to a significant but meaningless rejection of the null.

ENTERING A CLASSIFICATION VARIABLE INTO A CUMULATIVE LOGIT MODEL

1. If a test for unequalslopes for a classification variable X fails, then there are two choices for how X can enter the logistic model:

```
PROC LOGISTIC; CLASS X <class>; MODEL Y = X <all>;
```

or

```
PROC LOGISTIC; CLASS <class>; MODEL Y = X_woe1 X_woe2 <all>;
```

Only one of X_woe1 and X_woe2 would be entered if these predictors are highly correlated.

2. When unequalslopes is needed for a classification variable X, there are also two choices:

```
PROC LOGISTIC; CLASS X <class>; MODEL Y = X <all> / UNEQUALSLOPES=(X <some>;
```

or

```
PROC LOGISTIC; CLASS <class>; MODEL Y = X_woe1 X_woe2 <all> /  
UNEQUALSLOPES=(X_woe1 X_woe2 <some>;
```

There are pros and cons regarding the use of CLASS statements versus WOE coding. These are summarized at the end of the paper. As in the case of a binary logistic model, using CLASS X will produce a model with greater log-likelihood.

⁹ Agresti (2010 p. 64)

BINNING PREDICTORS FOR CUMULATIVE LOGIT MODELS

Binning is the process of reducing the number of levels of a NOD predictor to achieve parsimony while preserving, as much as possible, the predictive power of the predictor. In addition to parsimony, the purpose of binning is to establish a relationship between the bins and empirical cumulative logits that satisfies business expectations and common sense.

BINNING ALGORITHMS FOR A BINARY TARGET

Nominal: If a predictor X is nominal (no ordering), then there are two approaches to binning. They are (1) collapsing of levels where each level is initially in a separate bin, and (2) splitting where all levels are initially in one bin. A macro for collapsing, called %NOD_BIN, is given in Lund (2017b).¹⁰ A splitting algorithm is usually based on a decision tree. SAS Enterprise Miner implements splitting in the Interactive Grouping Node. Splitting algorithms will not be discussed in this paper.

In %NOD_BIN the collapsing is performed step-by-step. At each step the two bins are collapsed which optimize a specified criterion. Two choices for the collapsing criterion are provided by %NOD_BIN: (a) Information Value, and (b) Log-Likelihood. (Another alternative, not offered in %NOD_BIN, is to optimize the p-value from the chi-square measure of independence of X and the target.)

Ordered: If the predictor X is ordered, then a collapsing process is required where only adjacent levels of X can be collapsed. Adjacent level collapsing is performed by a macro %ORDINAL_BIN, given in Lund (2017b). The best k-bin solution based on either Information Value or Log-Likelihood is *guaranteed* to be found. In addition, if there exists a k-bin monotonic solution (i.e. WOE is monotonic in the ordering of X), then the best k-bin monotonic solution will be found.

%NOD_BIN also provides the option of collapsing only adjacent levels of X but %NOD_BIN cannot guarantee the finding of the optimal solution.

BINNING ALGORITHMS FOR THE CUMULATIVE LOGIT MODEL

The collapsing criteria from the binary algorithms can be generalized for binning decisions for the cumulative logit model.¹¹

Information Value: For the cumulative logit model, the use of Information Value for binning is complicated because each empirical cumulative logit has its own associated IV. One approach for binning decisions is to compute TOTAL_IV by simply summing the individual IV's.

Log-Likelihood: Log-likelihood may also be generalized for the cumulative logit model.¹² The natural generalization, consistent with the generalization of weight of evidence to the cumulative logit model, is the log-likelihood of the model below.

```
PROC LOGISTIC; CLASS X; MODEL Y = X / UNEQUALSLOPES = (X);
```

There is a simple formula for calculating this log-likelihood during DATA step processing.

%CUMLOGIT_BIN FOR THE CUMULATIVE LOGIT MODEL

A macro %CUMLOGIT_BIN is discussed. It is the cumulative logit analogue of %NOD_BIN. For %CUMLOGIT_BIN the predictor X may be integer (with values 0 to 99) or be character. The target variable is character or numeric with at least 2 levels. Observations with missing values are ignored.

Two required parameters for %CUMLOGIT_BIN are:

- MODE: Defines the pairs of levels of predictor X that are eligible for collapsing together. Choices are:
 - “A” for any pairs are eligible.
 - “J” for only pairs of levels that are adjacent in the ordering of X.

¹⁰ Clustering is an alternative. SAS code for clustering, due to M. J. Greenacre, is given in Manahan (2006).

¹¹ For “Splitting” it is not clear how splitting by a decision tree would be used for binning for the cumulative logit.

¹² In the binary case the log-likelihood is computed for the model: `PROC LOGISTIC; CLASS X; MODEL Y = X;`

- **METHOD:** Defines the rule for selecting an eligible pair for collapsing. Choices are TOTAL_IV and -2*LOG(L) (which is -2 times log-likelihood).
 - “IV” for TOTAL_IV. Here, the two levels of the predictor which give the greatest TOTAL_IV after collapsing (versus all other choices) are the levels which are collapsed at that step.
 - “LL” for -2*LOG(L). Here, the two levels of the predictor which give the smallest -2*LOG(L) after collapsing (versus all other choices) are the levels which are collapsed at that step.

Once selected, MODE and METHOD are applied at each step in the binning process.

The parameters for %CUMLOGIT_BIN are defined below:

DATASET: Data set to be processed

X: Predictor variable.

If numeric, then X must have integer values from 0 to 99. There is no restriction on character values. If the character values represent an ordered predictor, then care must be taken to assign values that give the intended ordering. X must have at least 2 levels. Missing values are ignored.

TARGET:

Target variable with numeric or character values. The ordering of TARGET variable determines the ordering in CUMLOGIT_BIN processing. TARGET must have at least 2 levels. Missing values are ignored.

W: A frequency variable if present in DATASET. Otherwise enter 1. Space is not permitted as entry.

METHOD: IV or LL

MODE: A or J

ONE_ITER: YES | <other>.

YES restricts reporting to only the statistics for bins before any collapsing. Priority over MIN_BIN

MIN_BIN:

INTEGER > 1 | space. Integer value restricts the processing to bin solutions where the number of BINs is greater or equal to the INTEGER. If <space>, then all bin solutions are processed.

VERBOSE:

YES | <other>. The value YES significantly increases the volume of printed output. The use of YES is not recommended for the initial run of the macro. Once the number of bins in the binning solution is determined by the model, then VERBOSE=YES is run to obtain the SAS coding for weight of evidence and classification level coding. The initial run, without VERBOSE, will create only Summary Reports. By viewing the Summary Reports the modeler can usually decide on the number of bins for the final binning solution.

AN EXAMPLE OF %CUMLOGIT_BIN

THE DATA

Consider data set called BACKACHE in Table 9. It gives the age of women who were pregnant and the severity of backache they experienced.¹³ Severity has three levels: 1, 2, and 3 with 1 being least severe.

The observations in Table 9 were grouped (preliminary binning) into age-ranges due to small sample sizes for some age levels. A grouping of observations is needed to avoid “zero cells” where the count of severity is zero for a level of age. Otherwise, a binning algorithm based on Total IV or Log-Likelihood would not be possible due to undefined values of the logarithm.¹⁴

¹³ AGE_GROUP of pregnant women who had one of 3 levels of SEVERITY (1 = none or very little pain, 2 = troublesome pain, and 3 = severe pain) of backache from the “BACKACHE IN PREGNANCY” data set in Chatfield (1995, Exercise D.2).

¹⁴ An alternative is to assign a small value, such as 0.1 to a zero-cell.

SAS code for reading the raw data into the age-grouped BACKACHE data set is in the Appendix.

Age Group	Severity		
	1	2	3
15to19	10	5	2
20to22	20	12	2
23to24	19	8	3
25to26	15	13	4
27to28	8	7	2
29to30	6	7	3
31to32	4	5	3
33to35	5	1	4
36andUP	6	2	4
Total	93	60	27

Table 9. Data Set BACKACHE in Pregnancy

THE MACRO CALL

The data set Backache with Age_group and Severity is binned by the macro:

```
%CUMLOGIT_BIN (Backache, Age_group, Severity, _freq_, IV, A, NO, , NO);
```

Since VERBOSE=NO, only summary reports are printed. One summary report is the “Summary Report of All Steps for Best BIN Solution”. In Table 10 five columns from this report are displayed:

- Column “Collapsed” shows which two levels collapsed at each step. A plus sign “+” shows the two levels that were collapsed at that step.
- Minus2_LL gives -2 times the log-likelihood for the logistic model:

```
CLASS Age_group; MODEL Severity = Age_group / UNEQUALSLOPES = (Age_group);
```

- The “Information Value” columns are also computed for this same model.
- “Corr_woe_1_2” is the correlation between the two WOE variables for Age_group.¹⁵

```
CUMLOGIT_BIN, DATASET= Backache
X= Age_group, TARGET= Severity, FREQ= _freq_, METHOD= IV, MODE= A
Summary Report of All Steps for Best BIN Solution
```

step	Collapsed	minus2_LL	Total_IV	IV_1	IV_2	Corr_woe_1_2
9		339.5	0.614	0.138	0.476	0.581
8	25to26+27to28	339.5	0.613	0.138	0.476	0.581
7	15to19+23to24	339.6	0.609	0.135	0.474	0.578
6	33to35+36UP	339.9	0.605	0.135	0.470	0.579
5	29to30+31to32	340.0	0.598	0.134	0.464	0.578
4	15to19_23to24+20to22	341.0	0.561	0.133	0.429	0.638
3	25to26_27to28+29to30_31to32	342.4	0.493	0.111	0.381	0.607
2	25to26_27to28_29to30_31to32+33to35_36UP	350.4	0.324	0.103	0.221	1

Table 10. Step by Step Results from %CUMLOGIT_BIN and MODE=A

¹⁵ If the number of levels of TARGET is greater than 3, there are separate columns Corr_woe_r_s where correlations for each combination of target levels $1 \leq r < s$ and $r < s \leq J$ will appear

A good solution appears to be the 5 bin solution because there is a large drop-off in Total_IV when going to the 4 bin solution (0.598 down to 0.561).¹⁶ The 5-bin model uses 10 degrees of freedom since the predictor is a class variable with 5 levels and with unequalslopes. Intercepts add 2 d.f.

As shown in Table 10 the value of $-2 \times \text{Log}(L)$ for the 5 bin solution is 340.0. This compares with $-2 \times \text{Log}(L)$ of 348.4 for the logistic model with (unbinned) age as the single predictor but with unequalslopes. Based on log-likelihood, the 5-bin solution has better fit but at the cost of 6 degrees of freedom versus the model with Age and unequalslopes. The code for this logistic model is shown in the Appendix.

The use of unequalslopes for Age_group is supported by the OneUp Test with p-value of 0.049. This significance value rejects the hypothesis “no difference” from adding unequalslopes.

The correlation between WOE predictors is fairly low at 0.578. Both WOE’s can be entered in the model.

A SECOND MACRO CALL

Now the modeler would rerun %CUMLOGIT_BIN with MIN_BIN=5 and VERBOSE=YES to obtain SAS coding for WOE and CLASS re-coding.

The SAS code for 5 bin solution is shown below. First for WOE

if Age_group in ("15to19","23to24") then Age_group_woe1 = 0.4102326976 ;
if Age_group in ("15to19","23to24") then Age_group_woe2 = 0.3936306505 ;
if Age_group in ("20to22") then Age_group_woe1 = 0.2899835694 ;
if Age_group in ("20to22") then Age_group_woe2 = 1.0379876669 ;
if Age_group in ("25to26","27to28") then Age_group_woe1 = -0.189293697 ;
if Age_group in ("25to26","27to28") then Age_group_woe2 = 0.2348395911 ;
if Age_group in ("29to30","31to32") then Age_group_woe1 = -0.654478039 ;
if Age_group in ("29to30","31to32") then Age_group_woe2 = -0.435318071 ;
if Age_group in ("33to35","36UP") then Age_group_woe1 = -0.066691374 ;
if Age_group in ("33to35","36UP") then Age_group_woe2 = -1.174985267 ;

Table 11. WOE coding for 5-bin Solution

if Age_group in ("15to19","23to24") then Age_group_bin = 1;
if Age_group in ("20to22") then Age_group_bin = 2;
if Age_group in ("25to26","27to28") then Age_group_bin = 3;
if Age_group in ("29to30","31to32") then Age_group_bin = 4;
if Age_group in ("33to35","36UP") then Age_group_bin = 5;

Table 12. CLASS coding for 5-bin Solution

BINNING ORDERED PREDICTORS FOR CUMULATIVE LOGIT MODELS

Since MODE was set to A, it is possible there are bins that contain levels that are not adjacent in the ordering of Age_group. In fact, this does occur as seen in Table 12. The 5-bin solution has the bin {15to19, 23to24} with two levels that are not adjacent.

For the sake of illustration, we will now *assume* that the expectation of medical researchers is that the relationship between Age_group and Severity should follow the natural ordering of Age so that adjacent-only collapsing is required.

Further, the medical researchers may have expectations regarding whether Age_group should be monotonic versus one or more of the empirical cumulative logits (or WOE’s). There are three cases:

¹⁶ Guidelines (like those in Table 3 for binary targets) are needed for how to interpret TOTAL_IV. One approach may be to apply the binary guidelines to the individual IVs from the cumulative logit model.

Case 1: Severity must be monotonic versus Age_group_woe1

Case 2: Severity must be monotonic versus Age_group_woe2

Case 3: Severity must be monotonic versus both Age_group_woe1 and Age_group_woe2

For Cases 1, 2, and 3 and for a given k, there may be no solution for a given Case. Of course, for k=2 every solution is ordered and satisfies Cases 1, 2, and 3. For k=3 this is no longer true.

The use of MODE=J in %CUMLOGIT_BIN does force all bins to have only members that are adjacent but %CUMLOGIT_BIN does not include logic to handle the additional requirements of Cases 1, 2, or 3.

Instead of %CUMLOGIT_BIN, a new approach is provided by using %ORDINAL_BIN, the macro for the binary target with ordered predictor. This new approach will find the best solution for each k (with no requirement for monotonicity) and also the best solution for Cases 1, 2, and 3, if such a solution exists. “Best” is either maximum Total IV or maximum log-likelihood.

In the section that follows we also *assume* that the medical researchers insist on having a binning that satisfies Case 1.

EXAMPLE: OPTIMAL BINNING FOR AGE_GROUP FOR CASE 1 AND $2 \leq k \leq 5$

The following discussion below assumes the reader has reviewed %ORDINAL_BIN in Lund (2017b). Otherwise, the reader may want to skip to the **RESULTS** section below.

First some data preparation: Two copies of data set BACKACHE are created, each with a binary target.

Data

```
Backache1_23 (keep=_freq_ severity1_23 age1_23)
Backache12_3 (keep=_freq_ severity12_3 age12_3 Severity)
;
Set Backache;
  if Severity = 1 then severity1_23 = 0;
  else severity1_23 = 1;
  age1_23 = Age_group;
output Backache1_23;
  if Severity = 3 then severity12_3 = 1;
  else severity12_3 = 0;
  age12_3 = Age_group;
output backache12_3;
run;
```

Discussion of %ORDINAL_BIN. [The current version of %ORDINAL_BIN is v13d.]:

The first %ORDINAL_BIN is applied to binary target {1} vs {2 3}. Solutions are obtained for k=2 to 5 (see MIN_BIN and MAX_BIN) and, within k, these solutions are ranked by IV. The best, if it exists, monotonic solution is found for k=2 to 5.

The second %ORDINAL_BIN is applied to binary target {1 2} vs {3}. Solutions are obtained for k=2 to 5 (MIN_BIN and MAX_BIN) and, within k, these solutions are ranked by IV. The best (up to) 100 solutions is specified for k=2 to 5. The purpose of specifying 100 is to ensure that if a monotonic solution is found for k by the first %ORDINAL_BIN, then it will match to a solution from the second %ORDINAL_BIN.¹⁷

```
%Ordinal_Bin (
  DATASET = backache1_23,
  X = age1_23 ,
  Y = severity1_23,
  W = _freq_,
  RANKING = IV, /* IV | LL */
  ORDER = A, /* D | A */
```

¹⁷ The minimum choice for N_BEST, to ensure all solutions are produced, is $\binom{L-1}{k-1}$ where L is the number of levels of X and k is the number of bins. For L=9 and k=5, this value is 70. So, 100 is higher than needed, which is OK.

```

MISS = , /* MISS | <other> */
N_BEST = , /* space = 1 or positive integer */
N_MONO = 1, /* space = 1 or positive integer */
MIN_PCT = , /* space = 0 | integer from 0 to 99 */
MIN_NUM = , /* space = 0 | negative integer */
MIN_BIN = 2, /* space or integer >= 2 (see rules) */
MAX_BIN = 5, /* space or integer >= 2 (see rules) */
NOPRINT_WOE = YES, /* YES | other */
PRINT1_WOE = , /* space or integer >= 2 (see rules) */
PRINT2_WOE = , /* space or integer >= 2 (see rules) */
RUN_TITLE = Severity 1 vs. 2 and 3, /* Title for run, NO commas */
DELETE_PRIOR = /* DELETE_YES or <other> */
);
%Ordinal_Bin (
DATASET = backache12_3,
X = age12_3 ,
Y = severity12_3,
W = _freq_,
RANKING = IV, /* IV | LL */
ORDER = A, /* D | A */
MISS = , /* MISS | <other> */
N_BEST = 100, /* space = 1 or positive integer */
N_MONO = , /* space = 1 or positive integer */
MIN_PCT = , /* space = 0 | integer from 0 to 99 */
MIN_NUM = , /* space = 0 | negative integer */
MIN_BIN = 2, /* space or integer >= 2 (see rules) */
MAX_BIN = 5, /* space or integer >= 2 (see rules) */
NOPRINT_WOE = YES, /* YES | other */
PRINT1_WOE = , /* space or integer >= 2 (see rules) */
PRINT2_WOE = , /* space or integer >= 2 (see rules) */
RUN_TITLE = Severity 1 and 2 vs. 3, /* Title for run, NO commas */
DELETE_PRIOR = /* DELETE_YES or <other> */
);

```

Discussion of post processing of %ORDINAL_BIN results:

The values of $k=2$ to 5 are specified as macro variables.

```

%Let min_bin_x = 2;
%Let max_bin_x = 5;

```

The first %ORDINAL_BIN saves the best monotone binning solution (if any) for $2 \leq k \leq 5$ in the data sets `__ORD_Age1_23_Best_<k>`. Macro variable MONO1 contains the code for extracting this best monotonic solution from these data sets.

```

%Let MONO1 = where = (best_mono = '*');

```

The second %ORDINAL_BIN saves the best 100 binning solutions as ranked by IV in the data sets `__ORD_Age12_3_Best_<k>` for $2 \leq k \leq 5$. MONO2 places no restrictions on these data sets.

```

%Let MONO2 = ; *where = (best_mono = '*');

```

%MONO1 and %MONO2 appear in the DATA MERGE below.

The DATA MERGE merges the binning solutions from {1} vs {2 3} and {1 2} vs {3} by the contents of the bins. This content is comprised of the values of the variables: `__x_char1 - __x_char&x`.

```

%macro Solutions;
%DO x = &min_bin_x %to &max_bin_x;
proc sort data = __ORD_Age1_23_Best_&x out = __ORD_Age1_23_Best_&x._sort;
  by __x_char1 - __x_char&x;

```

```

proc sort data = __ORD_Age12_3_Best_&x out = __ORD_Age12_3_Best_&x._sort;
  by __x_char1 - __x_char&x;
data Ordinal_Age_&x; MERGE
  __ORD_Age1_23_Best_&x._sort(in = in1 &MONO1
  rename = (solution_num = num1_23 IV = IV1_23))
  __ORD_Age12_3_Best_&x._sort(in = in2 &MONO2
  rename = (solution_num = num12_3 IV = IV12_3))
;
by __x_char1 - __x_char&x;
%do ii = 1 %to &x;
  rename __x_char&ii = bin&ii;
%end;
keep __x_char1 - __x_char&x num1_23 num12_3 TOTAL_IV IV1_23 IV12_3;
if in1 and in2;
TOTAL_IV = IV1_23 + IV12_3;
run;
proc sort data = Ordinal_Age_&x; by descending TOTAL_IV;
run;

proc print data = Ordinal_Age_&x;
var num1_23 num12_3 IV1_23 IV12_3 TOTAL_IV bin1 - bin&x
;
TITLE "Best MONO with &x levels";
TITLE2 "ORDINAL_BIN: 1 vs 23 &MONO1";
TITLE3 "ORDINAL_BIN: 12 vs 3 &MONO2";
run;
%END;
%MEND;
%Solutions;

```

RESULTS

The best ordered solution for $k=5$ which is monotonic for $\{1\}$ vs $\{2\ 3\}$ is shown below. In fact, this is the only such solution for $k=5$. The solution for $\{1\}$ vs $\{2\ 3\}$ was the 62th in the ranking by IV. For $\{1\ 2\}$ vs $\{3\}$ the matching solution was the 40th in the ranking by IV. There was a large loss of TOTAL_IV in order to find a 5 bin solution which is monotonic for $\{1\}$ vs $\{2\ 3\}$.

Num1_2 3	Num12_ 3	IV1_2 3	IV12_ 3	TOTAL L _IV	Bin1	Bin2	Bin3	Bin4	Bin5
62	40	0.0742	0.4102	0.4844	15to1 9	20to2 2	23to24+25to2 6	27to2 8	29to30+31to32+ 33to35+36andU P

Table 13. IV for 5 bin Solution with monotonic WOE for $\{1\}$ vs $\{2\ 3\}$

The weight of evidence values for $\{1\}$ vs $\{2\ 3\}$ and $\{1\ 2\}$ vs $\{3\}$ can be generated by the code below:

```

DATA BIN_5; length Target $8; Set
  __ORD_AGE1_23_BEST_5(in=in1_23 where = (solution_num = 62))
  __ORD_AGE12_3_BEST_5(in=in12_3 where = (solution_num = 40))
;
drop minus2LL B_sum: G_sum: ;
if in1_23 then Target = "1 vs 2 3";
if in12_3 then Target = "1 2 vs 3";
run;
PROC PRINT DATA = BIN_5;
run;

```

Target	Solu tion _num	Bin1	Bin2	Bin3	Bin4	Bin5	Woe1	Woe2	Woe3	Woe4	Woe5
1 vs 2 3	62	15to19	20to22	23to24+ 25to26	27to28	29to30+ 31to32+ 33to35+ 36andUP	0.290	0.290	0.127	-0.184	-0.389
1 2 vs 3	40						0.280	1.038	0.327	0.280	-0.790

Table 14. Results for 5 bins with monotonic WOE for {1} vs {2 3}

The WOE's for {1} vs {2 3} are monotonic decreasing. The odds of no or very little pain vs. troublesome or severe pain decrease with age. That is, more pain with age. For Case 1, there is no requirement regarding the pattern of WOE's for {1 2} vs {3}.

The following is true (but not demonstrated here) regarding Case 1 and k: There are no solutions for k=6, exactly 1 for k=5, 6 solutions for k=4, 14 for k=3, and 8 (all the solutions) for k=2.

Comments Regarding the Use of %Ordinal_Bin

- For targets with 3 levels the use of %ORDINAL_BIN is a workable approach for finding solutions for Cases 1, 2, and 3. As the number of levels of the target increases, so does the complexity of this approach and it becomes unwieldy.
- If MODE is changed to J in %CUMLOGIT_BIN, then an ordered 5-bin solution will be obtained. If IV is selected as the collapsing criterion, then the macro call is:

```
%CUMLOGIT_BIN (Backache, Age_group, Severity, F, IV, J, NO, 5, YES);
```

The 5-bin solution is not monotonic for {1} vs {2 3}.

if Age_group in ("15to19") then Age_group_woe1 = 0.2899835694 ;
if Age_group in ("20to22","23to24") then Age_group_woe1 = 0.3779944468 ;
if Age_group in ("25to26","27to28") then Age_group_woe1 = -0.189293697 ;
if Age_group in ("29to30","31to32") then Age_group_woe1 = -0.654478039 ;
if Age_group in ("33to35","36andUP") then Age_group_woe1 = -0.066691374 ;

Table 15. WOE for 5 bin Solution using %CUMLOGIT_BIN and MODE=J

WOE CODING VERSUS CLASS / DUMMY VARIABLE CODING

For a NOD predictor X, it is clearly easier to put X in a CLASS statement than it is to code WOE variables when fitting a cumulative logit model. A macro program or tedious hand coding is needed to form the WOE transformation. Also, the use of CLASS X in the presence of other predictors will give a somewhat better fit (measured by log-likelihood) than using WOE predictors.

But a drawback to using CLASS statements is the proliferation of coefficients. If the target has 3 levels and X has 10 levels, then 18 coefficients are introduced by CLASS X with UNEQUALSLOPES while only 4 are added by WOE coding. The use of CLASS and UNEQUALSLOPES becomes awkward if not impractical for a model with 10 or 20 NOD predictors.

Finally, the modeler might find a pattern in X_woe1 and X_woe2 that is consistent with expectations. For example, it might be that both X_woe1 and X_woe2 increase as X increases. After fitting the model, the product of the coefficient and a WOE variable will still be monotonic versus X. But when CLASS X (with unequalslopes) is used, the coefficients for the response equations may not be monotonic versus X.¹⁸

But, as in the case of binary logistic models, either WOE or CLASS variable may be used successfully in model fitting.

¹⁸ See the Appendix for a simple example in the binary case.

SAS MACRO DISCUSSED IN THIS PAPER

Contact the author for an experimental beta version of %CUMLOGIT_BIN and for SAS code that applies %ORDINAL_BIN to binning of ordered predictors for the cumulative logit model.

SESUG 2017, CARY, NC v08_fx2

REFERENCES

- Agresti, A. (2010). *Analysis of Ordinal Categorical Data*, 2nd Ed., Hoboken, NJ, John Wiley & Sons.
- Albert, A and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models, *Biometrika*, 71, 1, pp. 1-10.
- Allison, P.D. (2012), *Logistic Regression Using SAS: Theory and Application* 2nd Ed., Cary, NC, SAS Institute Inc.
- Chatfield, C. (1995). *Problem Solving: A Statistician's Guide*, 2nd Ed., Boca Raton, FL: Chapman & Hall/CRC.
- Derr, B. (2013). Ordinal Response Modeling with the LOGISTIC Procedure, *Proceedings of the SAS Global Forum 2013 Conference*, Cary, NC, SAS Institute Inc.
- Hosmer D., Lemeshow S., and Sturdivant R. (2013). *Applied Logistic Regression*, 3rd Ed., New York, John Wiley & Sons.
- Lund B (2017a). Weight of Evidence Coding for the Cumulative Logit Model, *Proceedings of the SAS Global Forum 2017 Conference*. Paper 785-2017.
- Lund B (2017b). SAS® Macros for Binning Predictors with a Binary Target, *Proceedings of the SAS Global Forum 2017 Conference*. Paper 969-2017.
- Lund B. and Brotherton D. (2013). Information Value Statistic, *MWSUG 2013, Proceedings*, Midwest SAS Users Group, Inc., paper AA-14.
- Manahan, C. (2006). "Comparison of Data Preparation Methods for Use in Model Development with SAS Enterprise Miner", *Proceedings of the 31th Annual SAS Users Group International Conference*, Paper 079-31.
- Siddiqi, N. (2006). *Credit Risk Scorecards*, Hoboken, NJ, John Wiley & Sons, Inc.

ACKNOWLEDGMENTS

All SAS programming for this paper was done by using SAS University Edition.

Dave Brotherton of Magnify Analytic Solutions of Detroit provided helpful insights and suggestions.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at: blund_data@mi.rr.com or blund.data@gmail.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

APPENDIX: AGE and SEVERITY from original BACKACHE Data Set

See Chatfield (1995, Exercise D.2). A change to raw data was made: if SEVERITY=0 then SEVERITY=1

```

Data Summary;
input Obs Severity Age _freq_ @@;
datalines;
1 1 16 1 17 1 33 2 32 2 26 10 47 3 24 1
2 1 18 4 18 1 34 2 33 2 27 3 48 3 25 2
3 1 19 5 19 1 35 1 34 2 28 4 49 3 26 2
4 1 20 3 20 1 37 1 35 2 29 3 50 3 27 1
5 1 21 12 21 1 39 1 36 2 30 4 51 3 28 1
6 1 22 5 22 1 42 4 37 2 31 2 52 3 29 1
7 1 23 7 23 2 17 1 38 2 32 3 53 3 30 2
8 1 24 12 24 2 18 3 39 2 35 1 54 3 31 1
9 1 25 7 25 2 19 1 40 2 36 1 55 3 32 2
10 1 26 8 26 2 20 3 41 2 37 1 56 3 33 1
11 1 27 4 27 2 21 3 42 3 15 1 57 3 34 1
12 1 28 4 28 2 22 6 43 3 19 1 58 3 35 2
13 1 29 3 29 2 23 3 44 3 20 1 59 3 36 1
14 1 30 3 30 2 24 5 45 3 21 1 60 3 38 1
15 1 31 1 31 2 25 3 46 3 23 2 61 3 39 2
16 1 32 3
;
run;
DATA Backache; Set Summary;
length Age_group $8;
if age <= 19 then Age_group="15to19";
else if age <= 22 then Age_group="20to22";
else if age <= 24 then Age_group="23to24";
else if age <= 26 then Age_group="25to26";
else if age <= 28 then Age_group="27to28";
else if age <= 30 then Age_group="29to30";
else if age <= 32 then Age_group="31to32";
else if age <= 35 then Age_group="33to35";
else Age_group="36andUP";
Age_groupB = Age_group;
run;
PROC FREQ DATA = Backache; TABLE Age_group * Severity;
weight _freq_;
run;

```

APPENDIX: LOGISTIC MODEL WITH PREDICTOR AGE

```

PROC LOGISTIC DATA = Backache;
MODEL Severity(descending)=Age / unequalslopes=Age;
Freq _freq_;
run;

```

This model has -2*Log(L) of 348.4.

APPENDIX: ONEUP TEST FOR AGE_GROUP

In PROC LOGISTIC the predictors Age_group and Age_groupB are entered by FORWARD with Age_group being forced in by the INCLUDE = 1 and Age_groupB being entered in "step 1" (by use of SLE = .999). The Log-Likelihood and Score chi-squares for Step 0 (from Include=1) and for Step 1 (from FORWARD) are captured in the ODS OUTPUT GLOBALTESTS.

```
ods output globaltests = globaltests;
PROC LOGISTIC DATA = Backache;
CLASS Age_group Age_groupB;
MODEL Severity(descending)=Age_group Age_groupB
/ unequalslopes=Age_groupB
Selection = forward sle = .999 include = 1 stop = 2;
Freq _freq_;
run;
```

The difference of these chi-squares gives a chi-square test statistic for the significance of a model with unequalslopes for Age_group versus the model with equal slopes. The test statistic and its p-value are computed in the final DATA step. The p-value can be based on the likelihood ratio chi-square or the score chi-square. For Age_group the p-values are surprisingly different at 0.2798 for LR and 0.0369 for Score.

```
DATA OneUp_LRT; SET globaltests end = eof;
retain ChiSq_LR_0 ChiSq_LR_1 ChiSq_Score_0 ChiSq_Score_1 DF_0 DF_1;
if Step = 0 & Test = "Likelihood Ratio" then do;
    ChiSq_LR_0 = ChiSq;
    DF_0 = DF;
end;
if Step = 0 & Test = "Score" then do;
    ChiSq_Score_0 = ChiSq;
end;
if Step = 1 & Test = "Likelihood Ratio" then do;
    ChiSq_LR_1 = ChiSq;
    DF_1 = DF;
end;
if Step = 1 & Test = "Score" then do;
    ChiSq_Score_1 = ChiSq;
end;
if eof then do;
    OneUp_LR = 1 - PROBCHI(ChiSq_LR_1 - ChiSq_LR_0, DF_1 - DF_0);
    OneUp_Score = 1 - PROBCHI(ChiSq_Score_1 - ChiSq_Score_0, DF_1 - DF_0);
    output;
end;
run;
PROC PRINT DATA = OneUp_LRT; Var OneUp_LR OneUp_Score;
Title "OneUp LRT with Likelihood Ratio and Score Chi-Squares";
run;
```

PREDICTORS WITH EQUAL SLOPES

In this Appendix there is a small simulation of the OneUp Test for WOE predictors in order to decide if a classification predictor X has equal slopes.

The DATA step creates data for a cumulative logit model where target has 3 levels, variable X has 8 levels, and X has nearly equal slopes in the two response equations. The OneUp test is applied to X where X is a CLASS variable. The coefficients of X are given by macros variables T and U.

```
%LET T = 0.09;
%LET U = 0.11;
%LET Seed = 1;
DATA SLOPES;
do i = 1 to 800;
    X = mod(i,8) + 1;
    T = exp(0 + &T*X + 0.01*rannor(&Seed));
    U = exp(1 + &U*X + 0.01*rannor(&Seed));
    PA = 1 - 1/(1 + T);
    PB = 1/(1 + T) - 1/(1 + U);
```

```

PC = 1 - (PA + PB);
R = ranuni(5);
if R < PA then Y = "A";
else if R < (PA + PB) then Y = "B";
else Y = "C";
freq1 = 1;
freq2 = 2;
freq3 = 3;
XB = X; /* for OneUp Test */
output;
end;
run;

```

Table 16 below shows that the OneUp Test is influenced by sample size. The results of the first simulation are given in lines 1 to 3 of Table 16.

When there is one copy of the sample (800 observations), there is a rejection of unequalslopes (line 1). With 2 and 3 copies of the sample, the unequalslopes is accepted.

In lines 4 to 6 of the report the coefficients of X are set to be the same at 0.10 (proportional odds). But even in this case, for the large sample (3 copies of 800) there is a significant OneUp Test.

Coefficient in equation T	Coefficient in equation U	Seed	Freq (copies of sample)	One-up LR p-value	Memo: Correlation of WOE's
0.09	0.11	1	1	0.2851	0.72598
0.09	0.11	1	2	0.0165	0.72598
0.09	0.11	1	3	0.000568	0.72598
0.10	0.10	2	1	0.5650	0.88984
0.10	0.10	2	2	0.1156	0.88984
0.10	0.10	2	3	0.0152	0.88984

Table 16. Sample Size and Effect on One-Up Significant Level

APPENDIX: PROBABILITIES – MONOTONIC FOR X_WOE BUT NOT FOR CLASS X

In data set TEST there is a binary target Y, predictor X, and frequency variable F. In data set TEST2 a new predictor XX is created and X_woe is also created. In Test2 the WOE values (or Odds) of Y are monotonic with respect to X.

In the Model with X_woe and XX as predictors, the probabilities for X (p_woe) are monotonic versus X for fixed XX. See Table 17.

In the Model with CLASS X and predictors X and XX the probabilities for X (p_class) are not monotonic versus X for fixed XX. See Table 17.

The “flexibility” of CLASS X causes the potentially important monotonic property of X versus Y to be lost.

```

DATA test;
input X Y F;
datalines;
1 0 2
1 1 1
2 0 1
2 1 1
3 0 5
3 1 6
4 0 2
4 1 12
;
run;

```

```

DATA test2; Set test;
Drop F i;
do i = 1 to F;
  XX = X + ranuni(1);
  if X = 1 then X_woe = log(1/2) - log(20/10);
  else if X = 2 then X_woe = log(1/1) - log(20/10);
  else if X = 3 then X_woe = log(6/5) - log(20/10);
  else if X = 4 then X_woe = log(12/2) - log(20/10);
  output;
end;
run;
data probs_for_X;
input X XX;
if X = 1 then X_woe = log(1/2) - log(20/10);
else if X = 2 then X_woe = log(1/1) - log(20/10);
else if X = 3 then X_woe = log(6/5) - log(20/10);
else if X = 4 then X_woe = log(12/2) - log(20/10);
datalines;
1 0
2 0
3 0
4 0
;
run;
PROC LOGISTIC DATA = test2;
Class X(PARAM=REF);
Model Y = X XX;
Score Data = probs_for_X out = class(rename=(p_1 = p_class));
PROC LOGISTIC DATA = test2;
Model Y = X_woe XX;
Score Data = probs_for_X out = x_woe(rename=(p_1 = p_woe));
DATA both; Merge class x_woe; by x;
PROC PRINT DATA = both;
var X p_class p_woe;
run;

```

Obs	X	p_class	p_woe
1	1	0.047178	0.11941
2	2	0.020212	0.15734
3	3	0.006212	0.16882
4	4	0.006817	0.29914

Table 17. Probabilities p_class and p_woe for X = 1 to 4 when XX=0