SAS® GLOBAL FORUM 2021

Paper 1022-2021

Weight of Evidence, Dummy Variables, and Degrees Of Freedom

Bruce Lund, Statistical Consultant and Trainer, Novi Michigan

ABSTRACT

Predictive models with a binary target are often fitted by logistic regression. An important step in using logistic regression is transforming of predictors before the model fitting stage. In credit risk modeling and direct marketing modeling, predictors are often transformed by weight of evidence (WOE) coding. This approach applies to discrete predictors, whether nominal, ordinal, or numeric. It also applies to continuous numeric predictors after such a predictor has been reduced to discrete ranges ("fine classing"). A widely used alternative to WOE coding is dummy variable coding. Let C be a discrete predictor and C_woe be its WOE coding. A model where C_woe is the only predictor has the same probabilities as the model with C appearing in CLASS C as the only predictor. Hence, the degrees of freedom of C_woe are L-1 where C has L levels. But if there are additional predictors in the model, it is unclear how to assign degrees of freedom to C_woe when considering the entry of C_woe into the model. Does C_woe have 1 degree of freedom, L-1 degrees of freedom, or something in between? This ambiguity affects the usage of predictor selection methods based on p-value significance, AIC, or BIC. In this paper it is shown that a model with predictor C_woe and other predictors <X> can be thought of as "nested" within the model with CLASS C and predictors C and <X>. It is this nesting property which suggests a process to assign degrees of freedom to C_woe when entering a model. This process enables the use of forward selection to select predictors for entry to minimize AIC but where the degrees of freedom for WOE predictors are adjusted, not simply assigned 1 degree of freedom. A SAS® macro to implement this process is available from the author.

INTRODUCTION

Predictive models with a binary target are often fitted by logistic regression.[1] An important step in using logistic regression is the transforming of predictors before the model fitting stage. In credit risk modeling and direct marketing modeling, predictors are often transformed by weight of evidence (WOE) coding.

The books by Siddiqi (2019), Finlay (2010), and Thomas (2009) show the usage of weight of evidence coding for credit risk modeling.

This WOE approach applies to discrete predictors (i.e. having only a few levels), whether nominal, ordinal, or numeric. It also applies to continuous numeric predictors after the predictor has been reduced to discrete ranges ("fine classing").

A widely used alternative to WOE coding is dummy variable coding. Let C be a discrete predictor and C_woe be its WOE coding. A model where C_woe is the only predictor has the same probabilities as the model with C appearing in CLASS C as the only predictor. Hence, the degrees of freedom (d.f.) of C_woe are L-1 where C has L levels.

[1] In this paper it is assumed that a logistic model does not have complete or quasi-complete separation. After models with separation are excluded, the logistic model has a unique maximum likelihood estimate. For discussion of separation, see Allison (2012, ch. 3).

But if there are additional predictors in the model, it is unclear how to assign d.f. to C_woe when considering the entry of C_woe into the model. Does C_woe have 1 degree of freedom, L-1 degrees of freedom, or something in between? This ambiguity affects the usage of predictor selection methods based on p-value significance, AIC, or BIC.[2]

In this paper it is shown that a model with predictor C_woe and other predictors <X> can be thought of as "nested" within the model with CLASS C, predictor C, and the same other predictors <X>. It will be this nesting property which suggests a process to assign d.f. to C_woe when entering a model.

This process enables the use of forward selection to select predictors for entry to minimize AIC but where the d.f. for WOE predictors are adjusted, not simply assigned 1 d.f.

A SAS macro to implement this process is available from the author.

WEIGHT OF EVIDENCE AND DUMMY VARIABLES

In Table 1 the weight of evidence coding (or transformation) of predictor C is illustrated. Predictor C can be numeric or character, with or without an ordering of its levels. The right most column gives the value "WOE(c_j)" of the weight of evidence transformation of C = c_j.

C	Y = 0 "B_j"	Y = 1 "G_j"	Col % Y=0 "b_j"	Col % Y=1 "g_j"	WOE(c_j)= Log(g_j/b_j)
c1	2	1	B_1 / B = 0.250	G_1 / G = 0.125	-0.69315
c2	1	1	B_2 / B = 0.125	G_2 / G = 0.125	0.00000
c3	5	6	B_3 / B = 0.625	G_3 / G = 0.750	0.18232
SUM	B=8	G=8			

Table 1. Weight of Evidence Transformation of C

The weight of evidence transformation is given here as a formula:

$$\text{If } C = c_j \text{ then } C\_woe(c_j) = \log((G_j / G) / (B_j / B)) = \log(g_j / b_j)$$

where:

g_j is the column percentage in the j^{th} row of Y=1, b_j is the column percentage in the j^{th} row of Y=0, G is the total count of Y=1 and B is the total count of Y=0.

If either g_j or b_j is zero, then C_woe is not defined.

There is not a tidy formula for computing the values for C_woe during processing of observations in a DATA Step. This is because the count of all occurrences of Y=1 and of Y=0 must first be made for each level C = c_j as well as the count of totals of Y=1 and of Y=0. Instead, a PROC SUMMARY is needed:

```
PROC SUMMARY data = Table1;
CLASS C Y;
TYPES C*Y Y;
OUTPUT OUT = WORK;
```

Next, a DATA Step processes data set `WORK` to compute the weight of evidence coding. Details of this DATA Step processing are not included here.[3] [4]

[2] AIC = -2*Log(L) + 2*K where K = 1 + d.f. of predictors in model. BIC = -2*Log(L) + log(N)*K with sample size N.

[3] SAS macros for computing WOE, as a by-product of variable binning, are given in Lund (2017). Since the publication of this 2017 paper, the macros have been enhanced. Contact the author for latest version of the macros.

[4] PROC HPBIN can compute the weight of evidence for numeric variables X. See parameter NUMBIN to specify the number of levels of X and see Example 4.5 in the documentation for PROC HPBIN.

CLASS STATEMENT AND DUMMY VARIABLE CODING

Suppose C has L > 2 levels and C appears in a CLASS statement in a logistic model:

PROC LOGISTIC descending; CLASS C (PARAM=ref); MODEL Y = C <other predictors>;

The statement CLASS C with (PARAM=ref) has the effect of creating dummy variables for the lowest (in natural sort order) of the L-1 levels of C. The effect of `PARAM=ref` is to set to zero the implied coefficient of the dummy variable for $C = c_L$.

Let dummy variables D_j be created from C by $D_j = (C = c_j)$ for j = 1 to L-1.

The equivalent dummy variable MODEL can be expressed, in terms of log-odds as:

$$\text{Log } (P/(1-P)) = \alpha + \sum_{j=1}^{L-1} \beta_j * D_j + \text{<other predictors>} \ldots \text{“CLASS model”}$$

The SAS code, using the dummy variables, is:

PROC LOGISTIC descending; MODEL Y = D_1 - D_{L-1} <other predictors>;

WEIGHT OF EVIDENCE AS AN ALTERNATIVE

Weight of evidence recoding of C is an alternative to using dummy variable coding for entering the predictor C into a logistic model.

With C_woe, the SAS logistic model statement is:

PROC LOGISTIC descending; MODEL Y = C_woe <other predictors>;

The weight of evidence model can be expressed, in terms of log-odds as:

$$\text{Log } (P/(1-P)) = \alpha + \beta * C\_woe + \text{<other predictors>}; \ldots \text{“WOE model”}$$

“NESTING” OF THE WOE MODEL WITHIN THE CLASS MODEL

WHEN WOE AND CLASS ARE THE SAME MODELS

In the simple case where the “WOE” model and “CLASS” model have no <other predictors>, these models are actually the same model. That is, they give the same probabilities, as explained below. Let C have L > 2 levels: $c_1 \ldots c_L$.

The maximum likelihood estimators (MLE) for the WOE model are given here:

$$\alpha = \log(G/B) \text{ and } \beta = 1 \ldots \text{(see Appendix for discussion.)}$$

Using the MLE’s, the probability P for $C=c_j$ is seen to be (here, via log-odds):

$$\text{Log}(P/(1-P) \mid C=c_j) = \log(G_j / B_j)$$

The maximum likelihood estimators for the CLASS (param=ref) model are given below:

$$\alpha = \log(G_L / B_L) \text{ and } \beta_j = \log(G_j / B_j) - \log(G_L / B_L) \ldots \text{(see Appendix for discussion.)}$$

Using the MLE’s, the probability P for $C=c_j$ is, again, seen to be (here, via log-odds):

$$\text{Log}(P/(1-P) \mid C=c_j) = \log(G_j / B_j)$$

OTHER PREDICTORS AND “NESTING”

The model with WOE predictors as well as other predictors is “nested” (in a sense to be explained) within the model with the corresponding CLASS predictors and the same “other predictors”.

The term “nested” is used here in a non-standard manner. Here is the usage:

For the CLASS model there exist values for the coefficients that give the same probabilities as the probabilities from the MLE solution for the WOE model.

These coefficients for the CLASS model are not its MLE's. The MLE solution for the CLASS model has greater log likelihood (or at least equal). That is, $Log(L)_{WOE(MLE)} \leq Log(L)_{CLASS(MLE)}$.

Here is a proof for the special class of one classification predictor C and one numeric predictor X. A more general case is discussed in the Appendix.[5]

Consider binary target Y, classification predictor C with four levels c1, c2, c3, c4, and one numeric predictor X. Let C_woe give the weight of evidence coding of C.

Consider the WOE model:

PROC LOGISTIC descending; MODEL Y = C_woe X;

Let $\alpha 0$, $\beta 0$, $\lambda 0$ denote the MLE coefficients for intercept, coefficient of C_woe, and coefficient of X respectively. The WOE values for C will be abbreviated by w1= C_woe(c1), …, w4= C_woe(c4).

Let dummy variables Dc1, Dc2, Dc3 be created for C=c1, C=c2, C=c3 by:

Dc1 = (C= "c1"), etc.

Consider the CLASS model:

PROC LOGISTIC descending; MODEL Y = Dc1 Dc2 Dc3 X;

It will now be shown that coefficients $\alpha 1$, $\beta 11$, $\beta 12$, $\beta 13$, $\lambda 1$ for the intercept, Dc1, Dc2, Dc3, and X can be found for the CLASS model so that the CLASS model is the same model (same probabilities) as the WOE model where the WOE model is evaluated at its MLEs.

The sought-after CLASS model coefficients $\alpha 1$, $\beta 11$, $\beta 12$, $\beta 13$, $\lambda 1$ must satisfy equations shown below with respect to the WOE model at its MLE solution given by $\alpha 0$, $\beta 0$, $\lambda 0$.

	WOE at MLE		CLASS
When C=c4 and X=0:	$\alpha 0 + w4 * \beta 0$	=	$\alpha 1$
When C=c4 and X=1:	$\alpha 0 + w4 * \beta 0 + \lambda 0$	=	$\alpha 1 + \lambda 1$

This implies that $\lambda 1 = \lambda 0$ and $\alpha 1 = \alpha 0 + w4 * \beta 0$

These equations must also hold:

When C=c1 and X=0:	$\alpha 0 + w1 * \beta 0$	=	$\alpha 1 + \beta 11$
When C=c2 and X=0:	$\alpha 0 + w2 * \beta 0$	=	$\alpha 1 + \beta 12$
When C=c3 and X=0:	$\alpha 0 + w3 * \beta 0$	=	$\alpha 1 + \beta 13$

The equations above imply that

$\beta 11 = \alpha 0 + w1 * \beta 0 - \alpha 1 = \alpha 0 + w1 * \beta 0 - \alpha 0 - w4 * \beta 0 = w1 * \beta 0 - w4 * \beta 0$

with similar equations for $\beta 12$ and $\beta 13$

The coefficients for CLASS have been solved in terms of the given coefficients for the MLE solution for the WOE model. Therefore, the WOE model is "nested" in the CLASS model.

The SAS code below gives an example for this simple case:

First a data set is created with classification variable C and numeric variable X. Target is Y:

```
DATA test;
do i = 1 to 500;
   z = rannor(1);
   C = (-2 <= z <= 2) * z;
   C = floor(C);
   X = ranuni(1) - .5;
```

[5] In the generalization discussed in the Appendix it is assumed that WOE predictors do not appear in interactions with each other or with other predictors.

```
    Y_star = C - 0.2*X + 2*rannor(1);
    Y = (Y_star > 0);
    output;
    end;
run;
```

Using a PROC SUMMARY and a DATA Step (not shown), the weight of evidence coding for C is obtained and inserted into the DATA Step below:

```
DATA test2; set test;
if C in ( -2 ) then C_woe = -1.557080788 ;
if C in ( -1 ) then C_woe = -0.430782916 ;
if C in (  0 ) then C_woe = 0.320521773 ;
if C in (  1 ) then C_woe = 1.259833467 ;
Dc1 = (C=-2);
Dc2 = (C=-1);
Dc3 = (C= 0);
Dc4 = (C= 1);
run;
```

Then PROC LOGISTIC fits the WOE model. In Table 2 the option "itprint" displays the coefficients as the fitting algorithm converges to the MLE solution:

```
PROC LOGISTIC data= test2 desc;
model Y= C_woe X / itprint;
score data= test2 out= woe;
run;
```

Maximum Likelihood Iteration History					
Iter	**Ridge**	**-2 Log L**	**Intercept**	**C_woe**	**X**
0	0	674.600231	-0.388826	0	0
1	0	616.649727	-0.336143	0.868390	-0.004264
2	0	615.547512	-0.386416	0.993267	-0.019273
3	0	615.544706	-0.388778	1.000243	-0.020309

Table 2. Maximum Likelihood Iterations

The MLE solution is used in setting the initial parameter values for fitting the CLASS model. The initial coefficients for the intercept and X are used directly (-0.388778 and -0.020309). The coefficient of C_woe which will be denoted by "beta0" remains to be used to define the initial coefficients for the three dummy variables in the CLASS model (corresponding to the four levels of C).

The initial coefficients for the three dummy variables (called beta11, beta12, beta13) involve the beta0 coefficient and the weight of evidence values w1, w2, w3, and w4 for C_woe. Here is the formula for beta11 (with similar formulas for beta12 and beta13):

beta11 = beta0*(w1 - w4) = 1.000243 * (-1.557080788 - 1.259833467) = -2.81760.

These initial coefficient values are saved in DATA step called "initial". The variable names Dc1, Dc2, Dc3 must be used for the coefficients beta11, beta12, beta13:

```
DATA initial;
beta0= 1.000243;
correction1 = (1.259833467)*beta0;
Dc1 =  -1.557080788*beta0 - correction1;
Dc2 =  -0.430782916*beta0 - correction1;
Dc3 =   0.320521773*beta0 - correction1;
X = -0.020309;
intercept= -0.388778 + correction1;
```

```
output;
run;
PROC PRINT data=initial;
run;
```

Obs	beta0	correction1	Dc1	Dc2	Dc3	X	Intercept
1	1.00024	1.26014	-2.81760	-1.69103	-0.93954	-0.020309	0.87136

These initial estimates are used in the 0th iteration of PROC LOGISTIC, below, by the statement `INEST= initial.` No iterations are run for the CLASS model when `MAXITER= 0` and model-fit uses the initial coefficient estimates. The same -2*Log(L) of 615.544 is found as for the WOE model:

```
PROC LOGISTIC data = test2 desc INEST = initial;
model Y = Dc1 Dc2 Dc3 X / MAXITER = 0 itprint;
score data = test2 out = class;
run;
```

Maximum Likelihood Iteration History							
Iter	Ridge	-2 Log L	Intercept	Dc1	Dc2	Dc3	X
0	0	615.544706	0.87136	-2.81760	-1.69103	-0.93954	-0.020309

Comparing output data sets `woe` and `class` it is seen that data set `both` (below) is empty:

```
DATA both; merge woe(rename=(P_1 = woe_P_1))
                 class(rename=(P_1 = class_P_1)); by i;
if abs(woe_P_1 - class_P_1) > .0001 then output;
run;
```

Let the log likelihood for the class model be $\text{Log(L)}_{\text{CLASS}}$ and the log likelihood for weight of evidence be $\text{Log(L)}_{\text{WOE}}$. Then, $\text{Log(L)}_{\text{WOE}} \leq \text{Log(L)}_{\text{CLASS}}$.

With an abuse of terminology, the results above show the WOE model is "***nested***" in the CLASS model.

WHY IS WEIGHT OF EVIDENCE EVEN CONSIDERED?

Why is the usage of C_woe considered in view of the greater fit from using CLASS C? Several reasons are presented below.

MODEL LOG-ODDS ARE LINEAR V. EMPIRICAL LOG-ODDS OF PREDICTOR C

The WOE coding of a predictor C gives the modeler more control over how this predictor impacts the predictions of the logistic model as explained below. Consider C with L > 2 levels and other predictors denoted by Z. Assume C_woe is not part of an interaction with another predictor.

The model log-odds for values of C are given by the left side of this equation:

$$\text{Log(P/(1-P)} \mid \text{C=c}_j\text{)} = \text{xbeta} = \widehat{\beta_{\text{C\_woe}}} * \text{C\_woe}(c_j) + \alpha + \widehat{\beta_Z} * Z$$

where Z gives the terms in xbeta for the other predictors and α is the intercept.

Therefore, for fixed values of Z, the model log-odds are linearly related to weight of evidence values (i.e. values of $\text{C\_woe}(c_j)$).

Now, $\text{C\_woe}(c_j) = \text{Log}(g_j/b_j) = \text{Log}(G_j/B_j) - \text{Log}(G/B)$. This gives:

$$\text{Log(P/(1-P)} \mid \text{C=c}_j\text{)} = \text{xbeta} = \widehat{\beta_{\text{C\_woe}}} * \text{Log}(G_j/B_j) - \widehat{\beta_{\text{C\_woe}}} * \text{Log}\ (G/B) + \alpha + \widehat{\beta_Z} * Z$$

The empirical log-odds of success, based solely on C = c_j are:

$$\text{Log}(G_j/B_j) = \text{Log}\ (\ [G_j/T_j]\ /\ [B_j/T_j]\ ) \text{ where } T_j = G_j + B_j$$

Thus, for fixed values of Z, the model log-odds are linearly related to the empirical log-odds. (The terms: - $\widehat{\beta_{C\_woe}}$ * Log (G/B) + α + $\widehat{\beta_Z}$*Z are constant.)

IF C_WOE IS MONOTONIC V. C, THEN SO ARE THE MODEL LOG-ODDS

Suppose C is ordinal and C_woe is monotonic with respect to C. Then the formulas developed above show there is a monotonic relationship between C and Log(P/(1-P)), where P is computed for values of C with other predictors being held fixed.

In the case of dummy variable coding with L-1 fitted coefficients, the effect of C on Log(P/(1-P)) is given through $\sum_{j=i}^{L-1} \hat{c}_j * \text{Dcj}$ where Dcj = (C = c_j) while other predictors are held fixed. Here, $\hat{c}_j$ is the coefficient of dummy variable Dcj, with reference level coding applied to the largest level of C.

The relationship between C and $\sum_{j=i}^{L-1} \hat{c}_j * \text{Dcj}$ need not be monotonic despite there being a monotonic relationship between C and C_woe. In fact, data set `test2` (given earlier) can be used to create an example, if it is assumed that that levels of C give an ordering.

OTHER VIRTUES OF WEIGHT OF EVIDENCE CODING INCLUDE:

- Fewer parameters are added to the logistic model. If C has L levels, then dummy variable coding adds L-1 parameters versus only 1 for WOE. Reduction in the number of parameters could be considerable.
- C_woe is numeric and can be compared with other numeric predictors to assess collinearity.

WOE is widely used in credit risk modeling since there is a natural connection between WOE coding and the generation of a scorecard.[6]

BUT PROBLEMS WITH WOE FOR PREDICTOR SELECTION

Let C have L > 2 levels. A disadvantage of WOE coding is that the degrees of freedom for C_woe, when being entered into a logistic model, are unknown. Here is an explanation.

As shown earlier, the following two models produce the same probabilities:

(A) PROC LOGISTIC DESCENDING; CLASS C; MODEL Y=C;

(B) PROC LOGISTIC DESCENDING; MODEL Y=C_woe;

In Model (A) predictor C uses L-1 d.f. Therefore, in Model (B), C_woe must also use L-1 d.f.

Suppose numeric predictors W and Z are added to models (A) and (B) to form models (A2) and (B2). The models (A2) and (B2) are not the same and $\text{Log(L)}_{A2} \geq \text{Log(L)}_{B2}$.

(A2) PROC LOGISTIC DESCENDING; CLASS C; MODEL Y = C W Z;

(B2) PROC LOGISTIC DESCENDING; MODEL Y = C_woe W Z;

The degrees of freedom for Model (A2) equal L-1 + 3 (includes 1 d.f. each for W, Z, and Intercept). The degrees of freedom for Model (B2) are undetermined but lie between 4 and L-1 + 3. It is difficult to accept that the d.f. of Model (B2) would only be 4 after noting that C_woe, in Model (B), has L-1 d.f. where L could be much greater than 2. On the other hand, to fully load Model (B2) with L-1 + 3 d.f. seems wrong in cases where Log(L)_{A2} is much larger than Log(L)_{B2} and, therefore, (A2) and (B2) are not, at all, the same model.

[6] See: Siddiqi (2017)

Of course, the problem in assigning d.f. to Model (B2) is due to the problem in assigning d.f. to C_woe as it enters a model already having W and Z. Should C_woe have 1 d.f. or L-1 d.f. or something else?

This d.f. assignment is important when considering the use of p-values in predictor selection methods (e.g. stepwise p-value based) and also for predictor selection methods based on minimum BIC and AIC (as are provided by PROC HPLOGISTIC).

None of the SAS procedures allow for d.f. adjustment for WOE predictors. Of course, more fundamentally, it is unclear how to make such an adjustment.

I have not seen a discussion of how to assign degrees of freedom for WOE predictors in modeling applications. I assume that, in practice, C_woe is simply regarded as having 1 d.f. in conformance with its usage in PROC LOGISTIC and PROC HPLOGISTIC. In situations with many predictors, is the use of the 1 d.f. assignment a reasonable simplifying assumption? Research on the question is needed. Some insights are given in discussions which follow.

THE MODEL COMPARISON TEST - REVIEW

The model comparison test requires truly nested models where each model has degrees of freedom equal to its number of parameters. The test statistic T is the difference of the "-2*Log(L)" from the models:

$$T = -2*\text{Log}(L)_{\text{restricted}} - (-2*\text{Log}(L)_{\text{full}})$$

For large samples the distribution of T is a chi-square with degrees of freedom given by $\text{d.f.}_{\text{full}} - \text{d.f.}_{\text{restricted}}$. Let t be the value of T from a sample and specify α in $0 < \alpha < 1$ (e.g. $\alpha = 0.05$). If $P(T \geq t) > \alpha$, then the restricted and full model are deemed statistically equal.

D.F. FOR C_WOE IN SELECTION=FORWARD IN MODEL FITTING

Suppose a logistic model is fit by "forward selection" where the selected predictor is the one giving smallest AIC. Assume numeric Z has already been selected. Let the candidate predictors be numeric X and weight of evidence C_woe where C has L > 2 levels.

Logistic procedures, (PROC LOGISTIC or PROC HPLOGISTIC), regard C_woe as having 1 degree of freedom. Suppose AIC values associated with *entry* of X or C_woe (with 1 d.f.) are given below. Using minimum AIC criterion, the predictor C_woe is selected next.

Predictor	-2*Log(L)	AIC	d.f. with entry
C_woe	**100.0**	**106.0**	1 (Intercept) + 1 (Z) + 1 (C_woe) = 3
X	102.0	108.0	1 (Intercept) + 1 (Z) + 1 (X) = 3

Table 3. -2*Log(L) and AIC for Hypothetical Predictors C_woe and X

But C_woe involves a pre-modeling transformation of C to C_woe which packs a lot of information into C_woe. What might be a "more fair" assignment of d.f. to C_woe than a default of d.f. equal to 1?

For this purpose a new "model comparison test" is proposed. Admittedly, this new test does abuse the standard model comparison test. Here, the WOE and CLASS models will be regarded as nested and the test statistic T is treated as a chi-square:

$$T = -2*\text{Log}(L)_{\text{woe}} - (-2*\text{Log}(L)_{\text{class}})$$

with $\text{d.f.}_T = \text{d.f.}_{\text{class}} - \text{d.f.}_{\text{woe}}$.

While $\text{d.f.}_{\text{class}}$ are known, the value of d.f._T is not known, due to the uncertain d.f. for C_woe. If, by some means, d.f._T could be assigned, then d.f._{woe} are known.

The next definition assigns this value of d.f._T. First, let t be the sample value of T.

Definition: Given α (0 < α <1), then $d.f._T$ are declared to be the d.f. value so that

$$P(T > t \mid d.f._T) = \alpha, \text{ with fractional values of } d.f._T \text{ being allowed.}^7$$

These are the d.f. that just barely make WOE and CLASS models statistically equal (for α).

Now $d.f._{class} = (L-1) + 2$ with L-1 d.f. for C, 1 d.f. for Z and 1 d.f for Intercept.

Then solving for $d.f._{woe}$ gives:

$$d.f._{woe} = d.f._{class} - d.f._T = (L-1) + 2 - d.f._T$$

Canceling out 2 d.f. for Z and Intercept from WOE and CLASS models gives the final formula for the degrees of freedom for C_woe when entering the MODEL Y = Z;

$$C\_woe_{d.f.} = (L-1) - d.f._T$$

Now two conventions are imposed on this formula.

(1) If $d.f._T \geq L-2$, then $d.f._T$ are reset to L-2. It is in these cases where 1 d.f. is assigned to C_woe for model entry.

(2) If $d.f._T \leq 0.1$, then $d.f._T$ are reset to 0. It is in these cases where L-1 d.f. are assigned to C_woe for model entry. That is, Class C and C_woe add, essentially, equal information.

The operational approach to assigning the d.f. for entry of C_woe is given by the procedure below called **FSAA**.

FSAA: FORWARD SELECTION with ADJUSTED AIC

Referring back to Table 3, this FSAA process is applied to decide if C_woe or X will enter into the model. For this example, suppose C has L = 4 levels.

Step 1: Enter C as a CLASS predictor (with 3 d.f.) to the logistic model already having Z. Suppose $-2*Log(L)_{CLASS}$ for this model is 96.

Step 2: Compute t = $-2*Log(L)_{C\_woe} + 2*Log(L)_{CLASS}$. Here, $-2*Log(L)_{C\_woe}$ is the result of entering C_woe into a model already having Z. Then t = 100 - 96 = 4 … 100 from Table 3.

Now, t is regarded as a value from a chi-square statistic T with k d.f. where k > 0, allowing even fractional values. As a chi-square, P(T > t | k) increases with increasing k, for fixed t.

Step 3: Fix α. Let $d.f._T$ be the minimum k > 0 for which P(T > t | k) > α.

To find $d.f._T$, P(T > t | k) is computed for a sequence of $\{k_j\}_1^J$. This sequence begins with k_1= 0.1 and ends at k_J = L-2 with increments of 0.1. Two conventions are imposed. If P(T > t | 0.1) > α, then $d.f._T$ are reset to 0 and if P(T > t | L-2) ≤ α, then $d.f._T$ are reset to L-2.

Let α = 5%. Then $d.f._T$ = 1.1, as shown in the working table:

α = 5% and t = $-2*Log(L)_{C\_woe} + 2*Log(L)_{CLASS}$ = 100-96 = 4		
k	P(T > t | k)	Exceeds α ?
0.1 (*)	0.0026	No
Etc.		No
1.1	0.0525	Yes
Etc.		Not needed
2 (**)	0.1353	Not needed

(*) If k = 0.1, then assign $d.f._T$ = 0
(**) If "Exceeds α" is "No" in all rows, then $d.f._T$ = bottom row of k (here = 2)

[7] SAS function cdf('CHISQ', t, k) gives cum chi-square probability P(T < t) for k d.f. for any k > 0.

Step 4: The degrees of freedom assigned to C_woe (with Z already in the model) are determined by the formula:

$$C\_woe_{d.f.} = (L-1) - d.f._T = 3 - 1.1 = 1.9$$

Conclusion: The new AIC for C_woe is 100 + 2*(2 + 1.9) = 107.8. (Two d.f. are given by Intercept and Z). The AIC for C_woe is less than the AIC for X (108). So, C_woe is entered.

NOW CONSIDER A NEW CASE:

If $-2*Log(L)_{CLASS} = 99.7$, then the calculations in the table below assign $d.f._T = 0$.

$\alpha = 5\%$ and $t = -2*Log(L)_{C\_woe} + 2*Log(L)_{CLASS} = 100-99.7 = 0.3$		
k	P(T > t | k)	Exceeds α ?
0.1 (*)	0.0722	Yes
Etc.		Not needed
2 (**)	0.8607	Not needed

(*) If k = 0.1, then assign $d.f._T = 0$;
(**) If "Exceeds α" is "No" in all rows, then $d.f._T$ = bottom row value of k

The formula for d.f. to enter for C_woe is:

$$C\_woe_{d.f.} = (L-1) - d.f._T = 3 - 0 = 3$$

The new AIC for C_woe is 100 + 2*(2 + 3) = 110. (Two d.f. are given by Intercept and Z). The AIC for C_woe is more than the AIC for X (108). So, X is entered.

COMMENTS:

- The choice of α is important in the assignment process. In general, for fixed sample size N, if α is decreased toward 0, then the adjusted d.f. for C_woe will reach L-1. Otherwise, if α is increased toward 1, then the adjusted d.f. for C_woe will reach 1.
- The chi-square test is sensitive to sample size N. For larger samples, with the same set of predictors, the α significance level should be decreased.
- The user should regard α as a tuning parameter and not so much as the type I error probability for an hypothesis test.

AN EXAMPLE OF FSAA

SAS PROC HPLOGISTIC documentation includes an example data set called getStarted.[8] It has 100 observations with binary target Y, one character variable C with 10 levels, and numeric X1 - X10. This data set is used in this example, however, only predictors X2, X8, X10, and C will be considered.

First, predictor C will be "binned" to reduce the 10 levels of C to 7 bins. The binning algorithm maximizes information value (IV) at each step in the reduction from 10 levels to 7 bins. For details about binning, see Lund (2017).[9]

After binning, C_woe is created as weight of evidence coding of C, as shown below:

```
DATA getStarted; length C $5; SET getStarted;
if C in ( "A","F" ) then C_woe = -0.809318612 ;
if C in ( "B","I","H" ) then C_woe = 0.1069721196 ;
if C in ( "C" ) then C_woe = 0.6177977433 ;
if C in ( "D" ) then C_woe = -0.403853504 ;
```

[8] https://support.sas.com/documentation/cdl/en/stathpug/66410/HTML/default/viewer.htm#stathpug_hplogistic_gettingstarted01.htm

[9] The macro %NOD_BIN from Lund (2017) was used for binning. In truth, the binning process should continue past 7 to 4 or even 3 bins. But the 7 bin solution was selected for the purpose of creating a good example to illustrate FSAA.

```
if C in ( "E" ) then C_woe = -1.145790849 ;
if C in ( "G" ) then C_woe = -0.703958097 ;
if C in ( "J" ) then C_woe = 1.4932664807 ;
/* After assigning WOE, collapse levels of C for current observation */
if C in ( "A","F") then C = "A_F" ;
if C in ( "B","I","H" ) then C = "B_I_H" ;
run;
```

To reduce lengthy reports, SAS code is displayed and then summary results are given. The reader can run the SAS PROC LOGISTIC code to obtain the full reports. PROC HPLOGISTIC could also be used. For this example, set α = 5%.

Step 1:

```
PROC LOGISTIC DATA= getStarted desc; MODEL Y = X2;
PROC LOGISTIC DATA= getStarted desc; MODEL Y = X8;
PROC LOGISTIC DATA= getStarted desc; MODEL Y = X10;
PROC LOGISTIC DATA= getStarted desc; MODEL Y = C_woe;
PROC LOGISTIC DATA= getStarted desc; CLASS C; MODEL Y = C;
```

PROC LOGISTIC treats C_woe as having 1 d.f. The FSAA process needs to consider an adjustment. Using the bottom two PROC LOGISTIC's, the d.f. for C_woe can be determined. But with no other predictors, models (i) and (ii) are the same models, as discussed earlier.

(i) MODEL Y=C_woe;
(ii) CLASS C; MODEL Y=C;

Therefore, C_woe is given 6 d.f.

As shown in Table 4, X8 gives the minimum adjusted AIC and is entered in Step 1. Without the adjustment, C_woe would have been selected.

Step 1: Predictors in Model: Intercept				
Model d.f. =	1			
AIC =	125.820			
For entry	-2*Log(L)	Adj. d.f.	Model d.f.	Adj. AIC
X2	119.804	1	2	123.804
X8	119.462	1	2	123.462
X10	123.229	1	2	127.229
C_woe	111.531	6	7	125.531

Table 4. First Step in Forward Selection by Minimum Adjusted AIC

Step 2: Which is the next predictor to be entered? PROC LOGISTIC models need to be run:

```
PROC LOGISTIC DATA= getStarted desc; MODEL Y = X8 X2;
PROC LOGISTIC DATA= getStarted desc; MODEL Y = X8 X10;
PROC LOGISTIC DATA= getStarted desc; MODEL Y = X8 C_woe;
PROC LOGISTIC DATA= getStarted desc; CLASS C; MODEL Y = X8 C;
```

Using the bottom two PROC LOGISTIC's, the d.f. for C_woe to enter the model are computed to give L-1 - $d.f._T$ = 6 - 0 = 6. See the working table below.

α = 5% and t = $-2*\text{Log}(L)_{C\_woe} + 2*\text{Log}(L)_{CLASS}$ = 106.243 - 106.119 = 0.124		
k (*)	P(T > t | k)	Exceeds α ?
0.1	0.1087	Yes
Etc.		Not needed

(*) If k = 0.1, then assign $d.f._T$ = 0

As shown in Table 5 below, X2 gives the minimum adjusted AIC and this AIC is smaller than the AIC (123.462) at Step 1.

Therefore, X2 is entered in Step 2.

Step 2: Predictors in Model: Intercept, X8				
Model d.f. =	2			
AIC =	123.462			
For entry	-2*Log(L)	Adj. d.f.	Model d.f.	Adj. AIC
X2	114.396	1	3	120.396
X10	119.056	1	3	125.056
C_woe	106.243	6	8	122.243

Table 5.

Step 3: Which is the next predictor, if any, to be entered:

```
PROC LOGISTIC data = getStarted desc; MODEL Y = X8 X2 X10;
PROC LOGISTIC data = getStarted desc; MODEL Y = X8 X2 C_woe;
PROC LOGISTIC data = getStarted desc; CLASS C; MODEL Y = X8 X2 C;
```

The d.f. for C_woe to enter the model are computed to give L-1 - $d.f._T$ = 6 - 0 = 6.

α = 5% and t = -2*Log(L)$_{C\_woe}$ + 2*Log(L)$_{CLASS}$ = 100.577 - 100.294 = 0.283		
k (*)	P(T > t | k)	Exceeds α ?
0.1	0.0745	Yes
Etc.		Not needed

(*) If k = 0.1, then assign $d.f._T$ = 0

As shown in Table 6, C_woe gives the minimum adjusted AIC of 118.577 and this AIC is smaller than the AIC (120.396) at Step 2.

C_woe is entered in Step 3.

Step 3: Predictors in Model: Intercept, X8, X2				
Model d.f. =	3			
AIC =	120.396			
For entry	-2*Log(L)	Adj. d.f.	Model d.f.	Adj. AIC
X10	114.113	1	4	122.113
C_woe	100.577	6	9	118.577

Table 6.

Now the question is whether X10, the only remaining predictor, will enter the model. The adjusted AIC for X10 must be less than the AIC (118.577) of Step 3. The following PROC LOGISTIC must be run:

```
PROC LOGISTIC data = getStarted desc;
MODEL Y = X8 X2 C_woe X10;
run;
```

In Table 7 it is noted that C_woe has added 6 d.f. to the Model's d.f.

Step 4: Predictors in Model: Intercept, X8, X2, C_woe				
Model d.f. =	9			
AIC =	118.577			
For entry	-2*Log(L)	Adj. d.f.	Model d.f.	Adj. AIC
X10	99.175	1	10	119.175

Table 7.

Adjusted AIC after entry of X10 (119.175) is not less than the model AIC (118.577) from Step 3. Therefore, X10 is not entered.

COMMENTS:

- In the example, for each step, the degrees of freedom for C_woe were adjusted to 6 d.f. But generally, adjustments for WOE coding can go from 1 (no adjustment) to L-1, including all fractional values in between.[10]
- The FSAA process is likely to reduce or delay the entry of WOE predictors. This is simply because the penalty term of 2*(K+1) in AIC may be increased by FSAA. In this regard it is easy to give examples of a weak C_woe which would enter into a model if given 1 d.f. but would not enter the model, or would enter later, if the d.f. were adjusted by FSAA.
- Binning of discrete predictors (nominal, ordinal, numeric with few levels) is important to simplify a model. In the case of FSAA, binning will increase the chances of WOE predictors entering the model because predictive power is largely maintained while reduced d.f. make the adjustment penalty smaller.
- Forward selection was discussed here but the process could be reframed for Backward or Stepwise.

SAS MACRO AVAILABLE

The author has a SAS macro, with examples and documentation, for the FSAA process, available upon request.

CONCLUSION

This paper proposes a solution to the "degrees of freedom problem" for weight of evidence coded predictors when fitting a logistic model using forward selection where the selected predictor, at each step, minimizes AIC among the candidates for entry.

REFERENCES

Allison, P.D. (2012), *Logistic Regression Using SAS: Theory and Application 2nd Ed.,* Cary, NC : SAS Institute Inc.

Finlay, S. (2010). *Credit Scoring, Response Modelling and Insurance Rating*, London, UK : Palgrave Macmillan.

Hosmer D., Lemeshow S., Sturdivant R. (2013). *Applied Logistic Regression, 3rd Ed.,* Hoboken, NJ : John Wiley & Sons.

Lund, B. (2017). SAS® Macros for Binning Predictors with a Binary Target, *Proceedings of the SAS Global Forum 2017*.

Siddiqi, N. (2017). *Intelligent Credit Scoring*, 2nd edition, Hoboken, NJ : John Wiley & Sons.

Thomas, L. (2009), *Consumer Credit Models*, Oxford, UK : Oxford University Press.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Bruce Lund
blund_data@mi.rr.com or blund.data@gmail.com

[10] See Lund (2010), video presentation at SGF 2021, for a different example.

APPENDIX

MLE FOR: MODEL Y= C_woe; and CLASS C; MODEL Y= C;

For MODEL Y = C_woe

The likelihood equations for solving for MLE's are shown below.[11]

(i) For the intercept: $\sum_{i=1}^{n}[y_i - P\mathbf{x}_i] = 0$ … or … $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} P\mathbf{x}_i$

where there are n observations, y is a 0/1 target, **x** is a vector of predictors, and P**x** is the model probability at **x**

(ii) For a predictor z (from among the **x**): $\sum_{i=1}^{n} z_i[y_i - P\mathbf{x}_i] = 0$ … or … $\sum_{i=1}^{n} z_i y_i = \sum_{i=1}^{n} z_i P\mathbf{x}_i$

Given that an MLE solution exists for the WOE Model, it is the unique solution to the likelihood functions. The task is to show that α = log(G/B) and β = 1 satisfy the likelihood equations. (It is assumed that the model does not have separation.)

First, show that α = log(G/B) and β = 1 solves (i) the intercept equation.

Let G_j equal the number of Goods (y=1) at C = c_j. Let B_j equal the number of bad's. Let $N_j = G_j + B_j$. Let $G = G_1 + \ldots + G_L$ and $B = B_1 + \ldots + B_L$.

The LHS of the likelihood equation simplifies to $G_1 + \ldots + G_L$

Now consider P**x** evaluated by C = c_j with the estimates α = log(G/B) and β = 1.

Then $\exp(\alpha + \beta * C\_woe(c_j)) = (G/B) * (G_j/B_j) / (G/B) = G_j/B_j$

For the j^{th} level of C the RHS contributes the term $N_j * (G_j/B_j) / (1 + G_j/B_j) = G_j$

Collecting terms on the RHS gives $G_1 + \ldots + G_L$ which completes the solution.

By following similar calculations, it is shown that α = log(G/B) and β = 1 also solve the likelihood equation (ii) for the predictor C_woe.

For CLASS C; MODEL Y = C;

The arguments follow similar patterns to what has been given above.

WEIGHT OF EVIDENCE MODEL IS "NESTED" IN CLASS MODEL

This is an example of "nesting" for two WOE predictors and two numeric predictors. The WOE predictors do not appear in any interactions. First a data set TEST is created:

```
DATA TEST;
do i = 1 to 500;
   z = rannor(1);
   C1 = (-2 <= z <= 2) * z;
   C1 = floor(C1);
   C2 = floor(5*ranuni(1));
   X1 = ranpoi(1, 2);
   X2 = ranuni(1);
   Y_star = C1 + 0.2*C2 + X1 + 0.1*X2 + 0.3*rannor(1);
   Y = (Y_star > 1);
   output;
   end;
run;
/* compute WOE for C1 and C2 (off line) */
/* code C1_woe and C2_woe */
```

[11] Hosmer, et al. (2013, p. 37)

```
DATA test2; set test;
if C1 in (  0 ) then C1_woe = 0.7537248551 ;
if C1 in (  1 ) then C1_woe = 3.2496813411 ;
if C1 in ( -1 ) then C1_woe = -0.24808805 ;
if C1 in ( -2 ) then C1_woe = -1.740455431 ;
if C2 in ( 0 ) then C2_woe = -0.568030985 ;
if C2 in ( 1 ) then C2_woe = 0.0135637098 ;
if C2 in ( 2 ) then C2_woe = 0.2221084617 ;
if C2 in ( 3 ) then C2_woe = 0.2792668755 ;
if C2 in ( 4 ) then C2_woe = 0.1431347012 ;
C11D = (C1=0);
C12D = (C1=1);
C13D = (C1=-1);
C14D = (C1=-2);
C21D = (C2=0);
C22D = (C2=1);
C23D = (C2=2);
C24D = (C2=3);
C25D = (C2=4);
run;
/* Find MLE estimators for WOE model */
PROC LOGISTIC data = test2 desc;
model y = C1_woe C2_woe X1 X2 / itprint;
score data = test2 out = woe;
run;
/* Use MLE from WOE to initialize parameters for CLASS model */
DATA initial;
beta1=2.897408;
beta2=2.738215 ;
correction1 = (-1.740455431)*beta1;
correction2 = 0.1431347012*beta2;
correction = (-1.740455431)*beta1 + 0.1431347012*beta2;
C11D =  0.7537248551*beta1 - correction1;
C12D =  3.2496813411*beta1 - correction1;
C13D =  -0.24808805*beta1  - correction1;
C21D =  -0.568030985*beta2 - correction2;
C22D =  0.0135637098*beta2 - correction2;
C23D =  0.2221084617*beta2 - correction2;
C24D =  0.2792668755*beta2 - correction2;
X1 = 3.7041;
X2 = -0.1797;
intercept= -4.2096 + correction;
output;
run;
/* Initial parameters and no iterations to show CLASS = WOE */
PROC LOGISTIC data = test2 desc INEST= initial;
model y = C11D C12D C13D C21D C22D C23D C24D X1 X2 / maxiter= 0 itprint;
score data = test2 out = class;
run;
```

The reader may show that the logistic probabilities in data sets “woe” and “class” are essentially equal.