Screening, Transforming, and Fitting Predictors for Cumulative Logit Model

Bruce Lund
Independent Consultant, Novi, MI

This is a presentation of a paper from SAS Global Forum 2019

https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3067-2019.pdf

Cum Logit has Target Variable with Ordinal Levels

Examples of Ordinal (Ordered) Target

- Long-term bond credit ratings assigned by S&P:
  AAA, AA, A, BBB, BB, B, CCC, CC, C, D
- Medical diagnosis stage: stage 1, stage 2, stage 3, stage 4
- Satisfaction survey: Poor, Fair, Good

Screening, Transforming, and Fitting Predictors for Cumulative Logit Model

Topics for Today

A. Describe the Cumulative Logit Model

B. Screening (keep or drop) “NOD” predictors for cum logit model
   - NOD = Nominal, Ordinal, Discrete … few levels but more than 2

C. Binning and transforming of NOD predictors

D. Transforming Continuous predictors (“many” numeric levels)

E. Selecting predictors for fitting cum logit models:
   - PROC LOGISTIC, PROC HPLOGISTIC, PROC HPGENSELECT

Contact me for SAS macros discussed today.

Cumulative Logit Model

In my examples, Target Y has J=3 *ordered* levels, but can be any J ≥ 3

Let Target Y have 3 levels A < B < C and Predictors X1 and X2

Then *one* form of the Cumulative Logit Model is given by:

- $\text{Log}\,(p_A / (p_B + p_C)) = \alpha_A + \beta * X1 + \lambda * X2$ … response eq. for A
- $\text{Log}\,((p_A + p_B) / p_C) = \alpha_B + \beta * X1 + \lambda * X2$ … response eq. for B

Predictors have "Equalslopes" (same coefficient in both equations)

This is Proportional Odds (PO) cum logit model.

Coefficients found by maximum likelihood estimation - like binary case

-- No error term assumed nor required in this formulation of Cum Logit

-- The Latent Variable derivation of Cum Logit has an error term (not discussed)

Comments

- Proportional Odds (PO) assumption *may be wrong*.
- PROC LOGISTIC has Test for “Proportional Odds” (see later slide)
- If test fails, then consider *Partial* PO (PPO) Model (next slide)

- CUM LOGIT is the usual binary logistic if Target has 2 levels

  Some of Today’s topics can be used for binary logistic models

Partial PO Cumulative Logit Model (PPO)

Target has 3 levels (A, B, C) and Predictors X1 and X2
Then an example of PPO Cum Logit Model is:

- $\text{Log}\,(p_A / (p_B + p_C)) = \alpha_A + \beta_A * X1 + \lambda * X2$ … response equation for A
- $\text{Log}\,((p_A + p_B) / p_C) = \alpha_B + \beta_B * X1 + \lambda * X2$ … response equation for B

Here, β_A β_B are unequal. Not so for λ.

PPO *allows designated predictors to have unequal coefficients*

PPO is implemented in PROC LOGISTIC by "UNEQUALSLOPES" Statement (see later slide)

PPO has issues

PPO model can produce negative probabilities

- Can occur during model fitting
- When scoring external data (especially if there are extreme values of X's)

This topic is discussed by Richard Williams (2008) -- see my paper for reference

Williams makes three points (which I will summarize):

- "Probabilities might go negative in unlikely or impossible X ranges, e.g. when years of education is negative" … Unrealistic Data
- "Seems most problematic with small samples, complicated models, analyses where the data are being spread very thin" … Ill-Structured Modeling Situations
- "Multiple tests with 10s of thousands of cases typically resulted in only 0 to 3 negative predicted probabilities." … Doesn't happen much in practice

Example: Cumulative Logit PO Model (target has 3 levels)

```
DATA Test;
X1=1; X2=3; Y="A"; output;
X1=1; X2=3; Y="B"; output;
X1=1; X2=3; Y="C"; output;
X1=1; X2=3; Y="A"; output;
X1=2; X2=2; Y="A"; output;
X1=2; X2=3; Y="C"; output;
X1=2; X2=3; Y="C"; output;
X1=2; X2=2; Y="C"; output;
X1=2; X2=3; Y="B"; output;
X1=3; X2=3; Y="C"; output;
X1=3; X2=3; Y="A"; output;
X1=3; X2=3; Y="A"; output;
X1=3; X2=4; Y="C"; output;
X1=3; X2=4; Y="B"; output;
run;
PROC LOGISTIC
DATA=Test;
MODEL Y = X1 X2;
run;
```

Analysis of Maximum Likelihood Estimates						
Parameter	Y	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	A	1	0.9310	2.9117	0.1022	0.7492
Intercept	B	1	1.8225	2.9422	0.3837	0.5356
X1		1	-0.1074	0.6618	0.0264	0.8710
X2		1	-0.4273	1.0043	0.1810	0.6705

Intercept A for 1st equation.

Intercept B for 2nd equation.

Score Test for the Proportional Odds Assumption		
Chi-Square	DF	Pr > ChiSq
4.7855	2	0.0914 (borderline reject)

← X1 or X2 has unequalslopes ?

Example: PPO Cumulative Logit Model

```
DATA Test;
X1=1; X2=3; Y="A"; output;
X1=1; X2=3; Y="B"; output;
X1=1; X2=3; Y="C"; output;
X1=1; X2=3; Y="A"; output;
X1=2; X2=2; Y="A"; output;
X1=2; X2=3; Y="C"; output;
X1=2; X2=3; Y="C"; output;
X1=2; X2=2; Y="C"; output;
X1=2; X2=3; Y="B"; output;
X1=3; X2=3; Y="C"; output;
X1=3; X2=3; Y="A"; output;
X1=3; X2=3; Y="A"; output;
X1=3; X2=4; Y="C"; output;
X1=3; X2=4; Y="B"; output;
run;
```

Y has 3 levels ➔ 2 response equations ★

```
PROC LOGISTIC DATA = Test;
MODEL Y = X1 X2 / UNEQUALSLOPES = (X1);
run;
```

★

Analysis of Maximum Likelihood Estimates						
Parameter	Y	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	A	1	0.8248	3.0117	0.0750	0.7842
Intercept	B	1	1.8812	2.9737	0.4002	0.5270
★ X1	A	1	-0.0733	0.7244	0.0102	0.9194
X1	B	1	-0.1535	0.7601	0.0408	0.8399
X2		1	-0.4145	1.0251	0.1635	0.6860

Brief discussion of NOD predictors

If X is numeric with many levels or is binary (2 levels), then X can go directly into a cum logit model: PROC LOGISTIC; MODEL Y = X;

Now consider X with “few”, but more than 2 levels ...

This is what I call a “**NOD**” predictor

Nominal {blue, green, brown}, Ordinal: {poor, fair, good}, Discrete {0, 1, 2, 3}

Why not put a discrete X in MODEL Y = X? … Can do.

But often “0” means something special. E.g. X= number of arrests. X=0 is much different then X=1, 2, 3. Maybe X=0 should have a “Dummy”. Maybe also “1” … soon, may as well consider CLASS X.

Topic: Screening NOD

- Efficient screening of NOD predictors for cum logit model
  - Maybe dozens, hundreds of NOD predictors
- In Paper two macros are discussed:
  - %CUM_LOGIT_SCREEN_1 (fully discussed)
  - %CUM_LOGIT_SCREEN_2 (briefly discussed)

Model "c" for Cum Logit in PROC LOGISTIC

Measures model performance … 0.5 to 1.0. Higher is better.

Let target have levels $k = 1, 2, 3$

For each observation:

- Let Probabilities be p_k for $k = 1, 2, 3$
- Compute "mean score" as Mscore = $\sum_{k=1}^{3} p_k * (k - 1)$

  e.g. If $p_2 = 0.4$ and $p_3 = 0.1$, then Mscore = 0.4 + 2*0.1 = 0.6
- NOW: Same Idea as Binary Case
  - IP = "Informative Pairs" of obs (r, s) where Targets $Y_r \neq Y_s$
  - If $Y_r > Y_s$ and $Mscore_r > Mscore_s$, then CONCORDANT
  - If $Y_r > Y_s$ and $Mscore_r < Mscore_s$, then DISCORDANT
  - Else TIE …. And **Model c** = {CONCORDANT + 0.5*TIE} / IP

Model_c = Max(1 - Model_c, Model_c)

I cannot find a paper on Model c for Cum Logit

Is “model c” useful as a measure of predictive accuracy?

PROC LOGISTIC; MODEL Y = X;

Almost_perfect

X\Y	A	B	C
1	90	7	3
2	5	90	5
3	3	7	90

LRCS = 392.4
Model c = 0.931

Very_good

X\Y	A	B	C
1	60	30	10
2	20	60	20
3	20	20	60

LRCS = 97.9
Model c = 0.742

Not_so_good

X\Y	A	B	C
1	40	30	30
2	30	40	30
3	30	30	40

LRCS = 3.08
Model c = 0.545

Independent

X\Y	A	B	C
1	34	33	33
2	33	34	33
3	33	33	34

LRCS = 0.03
Model c = 0.504

From various simulations: Model c measures predictive accuracy

Saturated PPO Cum Logit Model with 1 NOD Predictor

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

Why Look at Saturated Model with X?

- If X and Y independent, then X adds no information about Y versus Model with an intercept-only

- Saturated model gives ways to measure deviation from independence
  - Likelihood ratio chi-square (LRCS) … the right tail probability
  - Model c
- These measures allow predictors to be screened.
  - If Saturated X is weak on both LRCS / Model c, then eliminate X
- If X passes screening, use of X in Model may not be "saturated":
  - Unequalslopes may not be needed, X might transformed or binned

Saturated PPO Cum Logit Model with 1 NOD Predictor

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

But do not need PROC LOGISTIC to compute output from Logistic

Compute in Data Step!

Model c =	0.635

	Chi-Square	DF	Pr > ChiSq
Like. Ratio	1.3368	4	0.8551

Then, with "some" programming, the same Data Step can be used to efficiently compute Model c and LRCS for many X's … leads to my SAS macro

%CUM_LOGIT_SCREEN_1 (Dataset, Target, Input, Sort);

For Target with ≥ 2 levels this macro computes:

★ • Likelihood Ratio Chi-Sq for saturated model
• Model "c" for the saturated model

★ All this is done in a *DATA Step* for many X's

Useful for Binary Logistic

★ %CUM_LOGIT_SCREEN_1 (Test2, Y, X1 X2, Model_c);

Y target (3 levels)

X1 numeric (3 levels), X2 character (3 levels)

Sort results by descending Model c

```
DATA Test2;
X1=1; X2='3'; Y="A"; output;
X1=1; X2='3'; Y="B"; output;
X1=1; X2='3'; Y="C"; output;
X1=1; X2='3'; Y="A"; output;
X1=2; X2='2'; Y="A"; output;
X1=2; X2='3'; Y="C"; output;
X1=2; X2='3'; Y="C"; output;
X1=2; X2='2'; Y="C"; output;
X1=2; X2='3'; Y="B"; output;
X1=3; X2='3'; Y="C"; output;
X1=3; X2='3'; Y="A"; output;
X1=3; X2='3'; Y="A"; output;
X1=3; X2='4'; Y="C"; output;
X1=3; X2='4'; Y="B"; output;
run;
```

%CUM_LOGIT_SCREEN_1 (Test2, Y, X1 X2, Model_c);

Var_name	Levels	LRCS	Pr > ChiSq	Model c
X1	3	1.337	0.855	0.6349
X2	3	3.063	0.547	0.5556

- **Pr>ChiSq of LRCS**: Lower is better
- **Model c:** Higher is better … what is "poor" Model c ?
- **RANK** the X's by **Pr>ChiSq and Model c**
  - Similar not identical RANKINGS
  - Look for X with high rank on BOTH
- Should use with %CUM_LOGIT_SCREEN_2 (mentioned later)

See PAPER for extensive example of SCREEN 1 & 2

Topic: Binning and Transforming of NOD Predictors

- Binning and Transforming of NOD predictors

But First, review for Binary Target:

- Weight-of-evidence (WOE)
- Information Value (IV)

Information Value and WOE for BINARY

X	Y = 0 "B_k"	Y = 1 "G_k"	Col % Y=0 "b_k"	Col % Y=1 "g_k"	$Log(g_k/b_k)$ = X_woe	$g_k - b_k$	IV Terms $(g_k - b_k) * Log(g_k/b_k)$
X1	2	1	25.0%	12.5%	-0.69315	-0.125	0.08664
X2	1	1	12.5%	12.5%	0.00000	0	0.00000
X3	5	6	62.5%	75.0%	0.18232	0.125	0.02279
SUM	8	8	100%	100%		IV =	0.10943

Weight of Evidence of X is X_woe

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

Siddiqi, N. (2017). *Intelligent Credit Scoring*

WOE's, IV's for Cum Logit

Two Binary "splits" of Target (levels A, B, C) … (A vs. BC, AB vs. C)

WOE's, IV's for each "split"

	Y=		
X	A	B	C
1	4	1	1
2	3	1	3
3	1	2	1

Binary: A vs. BC WOE1_X =	Binary: AB vs. C WOE2_X =	IV_1	IV_2
0.811	0.734	0.441	0.327
-0.170	-0.588		
-0.981	0.223		

★ CUM_LOGIT_SCREEN_2:
Reports IV of X for each SPLIT

Binning

Binning of NOD predictors

Reduces number of levels of X

Reasons:

- Achieve Parsimony (fewer degrees of freedom)
- Find Logical relationships (e.g. monotonicity)
- Combine a low frequency level with another level

But BIN without giving up (much) predictive power

%CUMLOGIT_BIN: Binning for Cum Logit Model

	Severity		
Age_Group	A	B	C
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

BINNING

The DATA: BACKACHE (*)

- Gives age of pregnant women and Severity of backache experienced
- Severity has three levels: A, B, and C with "A" being least severe.
- 9 Levels for Age_group

	Severity		
Age_Group	A	B	C
BIN 1	n_{1A}	n_{1B}	n_{1C}
BIN 2	n_{2A}	n_{2B}	n_{2C}
Etc.			
BIN k	n_{kA}	n_{kB}	n_{kC}
Total	93	60	27

(*) "BACKACHE IN PREGNANCY" data set in Chatfield (1995, Exercise D.2)

%CUMLOGIT_BIN

DATASET: Data set to be processed
TARGET: Target with numeric or character levels
X: Predictor (numeric or character)
W: A frequency variable

MODE: A or J:
Defines the eligible pairs of levels of predictor X for collapsing. A = any pair; J = pairs with adjacent levels

METHOD: IV or LL:
Defines rules for selecting an eligible pair to collapse.
IV = Sum of binary split IV's. **= IV1 + IV2**
LL = -2*log(likelihood)
... LL is computed for **saturated** model

A	J	Age_Group
OK	NO	15to19
		20to22
	NO	23to24
		25to26
	OK	27to28
	OK	29to30
		31to32
		33to35
OK		36andUP

"A" & "IV" AND STEP 1
THERE ARE 36 PAIRS = $\binom{9}{2}$
FIND THE 1 WITH MAX "IV"

%CUMLOGIT_BIN Results

Bins	-2*LL	(Total) IV	IV_1	IV_2
9	339.5	0.614	0.138	0.476
8	339.5	0.613	0.138	0.476
7	339.6	0.609	0.135	0.474
6	339.9	0.605	0.135	0.470
5	340.0	0.598	0.134	0.464
4	341.0	0.561	0.133	0.429
3	342.4	0.493	0.111	0.381
2	350.4	0.324	0.103	0.221

MODE = **A**

METHOD = **IV [i.e. (TOTAL) IV]**

Start with all 9 levels, then …

Collapse to 8 (details not shown),

Etc.,

… when to STOP?

Stopping at BIN=5

IV = 0.598

Avoids large drop at BIN=4

BIN1	BIN2	BIN3	BIN4	BIN5
15to19_23to24	20to22	25to26_27to28	29to30_31to32	33to35_36andUP

After Binning … Transforming

Two Decisions for a (possibly binned) NOD predictor

1. WOE's or DUMMIES (creates different models)
   If target has J levels, then J-1 WOE's
   If X has L levels, then L-1 dummies
2. EQUAL or UNEQUALSLOPES or In-Between

See Lund (2017) SESUG for more discussion

UNEQUALSLOPES comes at a "price":

Consider dummies:

If X has L levels, Target has J levels, then unequalslopes gives:

Number of Parameters = (J-1) * (L-1)

Topic: Transforming Continuous Predictors

Suppose there is predictor X for cum logit model

- How should X be used in Model?
  - X^2, Log(X), 1/X, or something more complicated ?

This discussion is also useful for binary logistic !

Function Selection Procedure and %FSP_8LR

FSP was developed in the 1990's. Bio-medical applications.

Multivariate Model-building (2008) by Royston and Sauerbrei

FSP: For continuous predictor X for *Cumulative Logit* model:

- Selects a final transform of X … (44 transforms are checked)

  Or

- Eliminates X from further consideration as a predictor

- My macro %FSP_8LR implements FSP for PO
  - *Multiple* X's can be processed efficiently: %FSP_8LR(DATA, Y, X1 - X50)
- %FSP_8LR_PPO implements FSP for PPO (not discussed today)

FSP: Looks for the best transformation of X

- First, translate X to make min(X)=1. If min(X) ≥ 1, then leave as is.
- Form the *Fractional Polynomials* (FP):

  X^p for **p** in S = {-2, -1, -0.5, 0, 0.5, 1, 2, 3} where "0" = log(x)

  There are **8** p's.

- Create two groups of transforms: FP1 and FP2

  8 FP1: $g(X,p) = \beta_0 + \beta_1 X^p$

  36 FP2:

  $g(X,p_1,p_2) = \beta_0 + \beta_1 X^{p1} + \beta_2 X^{p2}$ $\quad p_1 \neq p_2$ … 28

  $g(X,p_1,p_1) = \beta_0 + \beta_1 X^{p1} + \beta_2 X^{p1} \log(X)$ $\quad p_1 = p_2$ … 8

FSP: Looks for the best transformation of X

Selection: Fit each of the 8 FP1 and 36 FP2 models by logistic regression … 44 models in total!

(But my macro %FSP_8LR runs only 8 times)

- *FP1 Solution* is one highest log likelihood among the 8 models
- *FP2 Solution* is one highest log likelihood among the 36 models

Code Generates an Cum Logit PO Example

This code is in paper - DON'T LOOK AT CODE

- Predictor X (levels: 1 to 16)
- Use X to create "FP2" transform
  - 0.2*LOG(X) - 0.5*X^{-1}
  - With PO (equalslopes)
  - Also random error
- Then more code and finally, …
- … produces Y with 3 levels

Now, apply %FSP_8LR to this data

… next slide

```
%LET ERROR = 0.01;
%LET SLOPE1 = 0.2;
%LET SLOPE2 = -0.5;
%LET P_Seed = 5;
%MACRO SIM(NUM);
%DO Seed = 1 %TO &NUM;
  DATA Work_&Seed;
  do i = 1 to 8000;
     X = mod(i,16) + 1;
     rannorx = rannor(&Seed);
     T = exp(0 + &SLOPE1*LOG(X) + &SLOPE2*(1/X) + &ERROR*rannorx);
     U = exp(1 + &SLOPE1*LOG(X) + &SLOPE2*(1/X) + &ERROR*rannorx);
     PA = 1 - 1/(1 + T);
     PB = 1/(1 + T) - 1/(1 + U);
     PC = 1 - (PA + PB);
/* Assign Target Values to match model probabilities */
     R = ranuni(&P_Seed);
     if R < PA then Y = "A";
      else if R < (PA + PB) then Y = "B";
      else Y = "C";
     output;
     end;
run;
  %END;
%MEND;
%SIM(1);
```

3 Steps for Selection of Transform

p=-1 and log generated the data

Summary Report from %FSP_8LR(WORK, Y, X)

TEST	-2*Log(L)	TEST -STAT	df	P-VALUE	Trans 1	Trans2
Eliminate X (intercept only➔)	15824.5	172.0	4	0.000		
Use Linear X	15709.3	56.9	3	0.000		
Use FP1 … or …	15654.3	1.9	2	0.387	p=-0.5	
Use FP2	15652.4				p=-2	log

STEP 1: Test for H_0: "Eliminate X" vs. H_A: Go to STEP 2

15824.5 - 15652.4 = 172.0 … Chi-Square with 4 d.f.

Why 4 d.f.? … 2 for exponent and 2 for coefficient

… Rejects "Eliminate X"

Three Steps for Selection of Transform

TEST	-2*Log(L)	TEST _STAT	df	P-VALUE	Trans 1	Trans 2
Eliminate X	15824.5	172.0	4	0.000		
Use Linear X	15709.3	56.9	3	0.000		
Use FP1 … or …	15654.3	1.9	2	0.387	p=-0.5	
Use FP2	15652.4				p=-2	log

- STEP 2: H_0: Use X (linear) vs. H_A: Go to STEP 3
- … 15709.3 - 15654.4 = 56.9 … P-Value = 0 … Reject H_0: Use X (linear)
- Go to Step 3

Three Steps for Selection of Transform

TEST	-2*Log(L)	TEST _STAT	df	P-VALUE	Trans 1	Trans 2
Eliminate X	15824.5	172.0	4	0.000		
Use Linear X	15709.3	56.9	3	0.000		
Use FP1 … or …	15654.3	1.9	2	0.387	p=-0.5	
Use FP2	15652.4				p=-2	log

… Final solution is $X^{-0.5}$

- STEP 3: H_0: Use FP1 Solution vs. H_A: Use FP2 Solution
- P-Value = 0.387 … Accept H_0

Testing the Proportional Odds (PO) Assumption

Must a PPO Model be Considered … Answer: "Borderline"

TEST	-2*Log(L)	TEST _STAT	df	P-VALUE	Trans 1	Trans 2	ChiSq _PO	df _PO	Prob ChiSq_ PO
Eliminate X	15824.5	172.05	4	0					
Use Linear	15709.3	56.86	3	0			1.934	1	0.164
Use FP1	15654.3	1.90	2	0.387	p=-0.5		3.112	1	0.078
Use FP2	15652.4				p=-2	log	2.479	2	0.290

Borderline test result

Perhaps "unequalslopes" for $X^{-0.5}$?

Topic: Selecting Predictors for Model

- Selecting predictors for fitting cum logit models
  - PROC LOGISTIC
  - PROC HPLOGISTIC
  - PROC HPGENSELECT

Fitting the Cumulative Logit Models (PO and PPO)

After screening, binning, transforming:

- There may be *many* candidate predictors
- A predictor SELECTION method is *needed* for model fitting

PROC LOGISTIC:

Only procedure with UNEQUALSLOPES (added in version 12.1)

- If **no** UNEQUALSLOPES statement, then:
  - **All** SELECTION options apply to PO Cum Logit (stepwise, forward, etc.)
- If UNEQUALSLOPES is used, then:
  - All SELECTION, except SELECTION = SCORE (i.e. Best Subsets)

FORWARD Selection, Fitting PPO with PROC LOGISTIC

```
DATA WORK1;
Do ID = 1 to 5000;
   random = ranuni(1);
   If random < 0.5 then Y = "A";
   else if random < 0.8 then Y = "B";
   else Y = "C";
   X1 = floor(ranuni(1)*5) * random;
   X2 = rannor(1) * random;
   X3 = ranuni(10) * random;
   X4 = X3*ranuni(10);
   output;
   end;
run;
```

See P. Hilliard (2017) SGF paper

```
PROC LOGISTIC DATA= WORK1;
MODEL Y= X1  X2  X3  X4 / SELECTION= FORWARD SLE= .05
  UNEQUALSLOPES EQUALSLOPES;
run;
```

Analysis of Maximum Likelihood Estimates				
Parameter	Y	DF	Estimate	Pr > ChiSq
Intercept	A	1	2.5651	<.0001
Intercept	B	1	3.4682	<.0001
U_X1	A	1	-1.1298	<.0001
U_X1	B	1	-0.6621	<.0001
U_X3	A	1	-7.0304	<.0001
U_X3	B	1	-3.9414	<.0001

X2 and X4 are not selected.

X1 and X3 are selected with unequalslopes

HPLOGISTIC/HPGENSELECT fitting Cum Logit PO

Remember "Evils" of Forward/Stepwise/Backward based on SIGNIFICANCE !!!

HPLOGISTIC & HPGENSELECT support **only** Cumulative Logit PO

ALL predictor SELECT= options apply to Cumulative Logit PO

e.g. HPLOGISTIC (SBC, AIC), HPGENSELECT (LASSO)

```
PROC HPLOGISTIC DATA = WORK1;
MODEL Y =  X1 X2 X3 X4;
SELECTION METHOD = FORWARD (SELECT=AIC CHOOSE=AIC STOP=NONE);
run;

PROC HPGENSELECT DATA = WORK1   LASSOSTEPS= 40   LASSORHO= 0.8 ;
MODEL Y =  X1 X2 X3 X4 / DISTRIBUTION= BINARY; /* BINARY for CUM LOGIT! */
SELECTION METHOD = LASSO (CHOOSE=AIC STOP=NONE);
run;
```

Fitting the Cumulative Logit Models for PPO

Can HPLOGISTIC / HPGENSELECT be tricked in running PPO?

This would allow advanced SELECTION methods (AIC, SBC, LASSO) to be used for PPO models.

Yes, a data coding “trick” can make this work !!

But there are some issues (see Paper for discussion)

A robust testing plan is needed to determine limitations.

This DATA CODING TRICK is given in Stokes, Davis, Koch (2000) Categorical Data Analysis, 2nd ed. P. 533

A Description of the Trick

Start with Data Set WORK1.

With Y with **3** Levels = **0, 1, 2**

Create a new Data Set WORK2

For each obs. from WORK1:

  Output **2** obs. to WORK2

  Count them: SPLIT values **0** to **1**

  Same X's on each output obs.

```
IF SPLIT ≥ Y THEN Y2 = 0  /* BINARY */
 ELSE IF SPLIT < Y THEN Y2= 1
```

DATA WORK1

ID	Y	X
1	0	A
2	1	W
3	2	V

ID = 1

DATA WORK2

ID	SPLIT	Y	Y2	X
1	0	0	0	A
1	1	0	0	A
2	0	1	1	W
2	1	1	0	W
3	0	2	1	V
3	1	2	1	V

CUM LOGIT MODEL (PPO for X1)

```
/* #1 */
PROC LOGISTIC
DATA = WORK1;
MODEL Y =  X1 X2 X3 X4
/ UNEQUALSLOPES = X1;
/* #2 */
PROC HPLOGISTIC
DATA = WORK2; /*TRICKED*/
CLASS Split;
MODEL Y2 =
Split X1 X2 X3 X4 X1*Split;
```

MODEL #2	Estimate
Intercept	3.933
SPLIT 0	-1.728
SPLIT 1	0
X1	-0.642
X2	0.031
X3	-5.286
X4	-0.065
X1*SPLIT 0	-0.483
X1*SPLIT 1	0

Convert to Model #1

	MODEL #1	MODEL #2
Intercept A	2.2504	2.2051
Intercept B	3.9821	3.9326
X1 A	-1.1611	-1.1240
X1 B	-0.6728	-0.6415
X2	0.0332	0.0308
X3	-5.1511	-5.2855
X4	-0.0427	-0.0646

X1 and X1*SPLIT ➔ "unequalslopes" for X1

#1 & #2 have similar coefficients
But not the Same!

Best of "Both Worlds" - using HPLOGISTIC & LOGISTIC for PPO

- Recode DATA using TRICK. Choose predictors (suppose X1) for unequalslopes
- Run HPLOGISTIC (here … FORWARD and SELECT=AIC).
- Re-fit Selected Predictors by PROC LOGISTIC with UNEQUALSLOPES (if needed)

Selection Summary		
Step	Effect Entered	AIC
0	Intercept	
	SPLIT	11928
1	X3	9460.96
2	X1	8329.04
3	X1*SPLIT	8267.28*
4	X2	8268.06
5	X4	8270.01

```
PROC HPLOGISTIC DATA= WORK2;
CLASS SPLIT /PARAM = ref;
MODEL Y2= SPLIT X1 X2 X3 X4 X1*SPLIT
/ INCLUDE=1;
SELECTION METHOD= FORWARD SELECT=AIC
CHOOSE=AIC STOP=NONE);
```

```
PROC LOGISTIC DATA= WORK1;
MODEL Y= X1 X3 /*X2 X4*/
 / UNEQUALSLOPES= (X1);
```

Bruce Lund
Contact: blund_data@mi.rr.com