Screening, Transforming and Fitting Predictors for the Cumulative Logit Model

MiSUG 2018 | Bruce Lund

onemagnify
insight to impact

Topics

In 20 minutes …

Present methods to screen predictors for cumulative logit model

- SAS® Macros for NOD predictors (Nominal, Ordinal, Discrete with “few” levels)
- When there are dozens, hundreds of NOD predictors

- Discuss binning and transforming of predictors … briefly
  - Binning: Nominal, Ordinal, Discrete (NOD)
  - Transforming: Continuous
- Discuss methods of selecting predictors for fitting models … briefly
  - PROC LOGISTIC
  - PROC HPLOGISTIC, PROC HPGENSELECT

Familiarity with PROC LOGISTIC is assumed.

Cumulative Logit Model

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

Let Target 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 equation for A
- $\text{Log}((p_A + p_B) / p_C) = \alpha_B + \beta * X1 + \lambda * X2$ ... response equation for B

3 Levels ➔ 3-1 = 2 response equations. (J ➔ J-1 equations)

★ Here, coefficients unchanged across equations ("Equalslopes")
This is Proportional Odds (PO) cum logit model ("PO"? see paper)

Cumulative Logit Model with PO

- Proportional Odds (PO) assumption *may be wrong*.
- PROC LOGISTIC has Test for "Proportional Odds" (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.

  Like Binary, CUM LOGIT is fit by maximum likelihood

  See Allison (2012) "Logistic Regression using SAS …" chapter 6 for introduction

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)

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 ?

S counts predictors and J counts Target levels. Test statistic is chi-square with (J-2)*S d.f. Small values reject the proportional odds assumption.

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;
```

```
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

Y has 3 levels → 2 response equations

Review: Model "c" for Cum Logit (both PO and PPO)

Model "c" measures fit. Values: 0.5 to 1.0. Higher is better.

No interpretation as "Area under ROC curve"

For each observation:

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

Terminology: NOD v. Continuous Predictors

NOD: Nominal, Ordinal, Discrete

Typically few levels (unique values) … typically ≤ 20

- Nominal has no ordering … e.g. yellow, green, blue
- Ordinal is ordered … e.g. good, better, best
- Discrete is numeric … e.g. a count

Continuous Predictors: Lots of numeric levels

- E.g. money, distance, time

Saturated PPO Cum Logit Model with 1 NOD Predictor

```
PROC LOGISTIC DATA=Test;
CLASS X1; ★
MODEL Y = X1 /
UNEQUALSLOPES = (X1); ★
```

X1	Y: A	B	C	Tot
1	2	1	1	4
	.50	.25	.25	
2	1	1	3	5
	.20	.20	.60	
3	2	1	2	5
	.40	.20	.40	

Analysis of Maximum Likelihood Estimates

Parameter		Y	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept		A	1	-0.5973	0.5853	1.0412	0.3075
Intercept		B	1	0.3662	0.5774	0.4023	0.5259
X1	1	A	1	0.5973	0.8221	0.5277	0.4676
X1	1	B	1	0.7324	0.8819	0.6897	0.4063
X1	2	A	1	-0.7890	0.8714	0.8200	0.3652
X1	2	B	1	-0.7717	0.7817	0.9744	0.3236

Model c =	0.635

Testing Global Null Hypothesis: BETA=0

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

Can be computed in Data Step!

Why the Saturated Model?

- All Information about Y that is contained in X is used in Model
- Two ways to measure "all information" are:
  - Likelihood ratio chi-square (LRCS)
  - Model c
- These measures allow predictors to be screened.
  - If Saturated X is weak (LRCS / Model c), then eliminate X
- If X passes this screening, the use of X in Model may not be as "saturated". Further analysis is needed.
  - Unequalslopes may not be needed
  - X might transformed or binned

%CUM_LOGIT_SCREEN_1 (Dataset, Target, Input);

For Target with ≥ 2 levels this macro computes:

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

Target: At least 2 levels (missing ignored)
Input: Predictors numeric or character
(any number of X's, efficient processing)

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

X1 numeric with 3 levels
X2 character with 3 levels
Y is target with 3 ordered levels

```
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);

Var_ name	Lev els	Log_Like (Intercept)	Log_Like (Full)	LRCS	Pr > ChiSq	Model c	ASE
X1	3	-14.853	-14.185	1.337	0.855	0.6349	0.1123
X2	3	-14.853	-13.322	3.063	0.547	0.5556	0.0624

★ • **Pr>ChiSq of LRCS** (right tail probability) ranks the predictors

• No absolute "cut-off" - Significance influenced by "n"

★ • **Model c**: Higher is better but what is "poor"? … Model c < 0.6?

★ • RANK the X's by **Pr>ChiSq and Model c**

  • Likely different RANKINGS … Look X with high rank on BOTH

★ • Use with second Macro … to be presented on a later slide

Review: Information Value and WOE for BINARY

Weight of Evidence Coding of X: X_woe

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**

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. (2006). *Credit Risk Scorecards*

Review: c-Statistic for X (ordered) vs. Y for BINARY

- IP = "Informative Pairs" of obs (r, s) where Targets $Y_r \neq Y_s$
- If $Y_r > Y_s$ and $X_r > X_s$, then CONCORDANT
- If $Y_r > Y_s$ and $X_r < X_s$, then DISCORDANT
- Else TIE …. And c-Statistic = {CONCORDANT + 0.5*TIE} / IP

X	Y = 0	Y = 1	Concordant	Ties		
X1	2	1	2*1=2, 2*6=12	2		
X2	1	1	1*6=6	1		
X3	5	6		30	IP	c-Statistic
SUM	8	8	20	33	64	**0.57**

WOE's, IV's, C-STAT's, Model c for Cum Logit

- ★ Binary "splits" of Target (levels A, B, C) … (A vs. BC, AB vs. C)
- ★ WOE's, IV's / C-Stat's / Model c's are defined for each "split"
- ★ Numeric X is Monotonic for a "split" when WOE is monotonic vs. X

	Y=										
X	A	B	C	Binary: A vs. BC WOE1_X =	Binary: AB vs. C WOE2_X =	IV1	IV2	C-stat1	C-stat2	Model c1	Model c2
1	4	1	1	0.811	0.734	0.225	0.159				
2	3	1	3	-0.170	-0.588	0.012	0.157				
3	1	2	1	-0.981	0.223	0.204	0.011				
			★	Monotonic	NOT Mono	**0.441**	**0.327**	**0.674**	**0.567**	**0.674**	**0.650**

%CUM_LOGIT_SCREEN_2 (Dataset, Target, N_Input, C_Input, IV_ADJ);

This macro computes:

- IV for each binary split of cumulative logits (A vs. BC, AB vs. C)
- ★ c-statistic for binary splits (A vs. BC, AB vs. C)
- Model "c" for binary split for the binary saturated model

★ Is X "strong" for at least one split?

Target: At least 2 levels (missing ignored)
★ N_Input: Numeric Predictors (any number of X's, efficient processing)
C_Input: Character Predictors (any number of X's, efficient processing)
IV_ADJ: YES, added 0.1 to a zero cell to allow IV calculation

%CUM_LOGIT_SCREEN_2 (Test2, Y, X1, X2, YES);

- ★ High IV or MODEL "c" for a split shows predictor is strong for this split.
- ★ C-STAT for split ➔ degree of monotonic tendency
- ★ Monotonic=Yes when WOE is monotonic v. X for split

VAR _NAME	SPLIT _POINT	VL	NOM -INAL	MONO TONIC	C_STAT	MODEL "c"	IV
X1	A - B	3	NO	★ NO	0.533	0.644	0.312
X1	B - C	3	NO	NO	0.552	0.656	0.347
X2	A - B	3	YES	n/a	n/a	0.633	0.564
X2	B - C	3	YES	n/a	n/a	0.542	0.034

```
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;
```

Example for CUM_LOGIT_SCREEN 1 and 2

SIM_1 (created here) is a data set for a cum logit model

Target = Y
Predictors = X1-X5 and character C1

```
%LET ERROR = 0.01;
%LET SLOPE1 = 0.01;
%LET SLOPE2 = 0.05;
%LET SLOPE3 = 0.10;
%LET SLOPE4 = 0.20;
%LET SLOPE5 = 0.99;
%LET P_Seed = 5;
%MACRO SIM(NUM);
%DO Seed = 1 %TO &NUM;
 DATA SIM_&Seed;
/* Continued on Right */
```

```
do i = 1 to 8000;
   X1 = floor(12*ranuni(2)) - 1.5;
   X2 = floor(2*ranuni(2)) - .5;
   X3 = floor(2*ranuni(2)) - .5;
   X4 = floor(2*ranuni(2)) - .5;
   X5 = floor(2*ranuni(2)) - .5;
   C1 = put(floor(4*ranuni(2)),z2.);
   C1_all = &SLOPE1*(C1='00') + &SLOPE2*(C1='01') + &SLOPE3*(C1='02');
   rannorx = rannor(&Seed);
   T = exp(0 + C1_all + &SLOPE1*X1 + &SLOPE2*X2 + &SLOPE3*X3 + &SLOPE4*X4 + &SLOPE5*X5 + &ERROR*rannorx);
   U = exp(1 + C1_all + &SLOPE1*X1 + &SLOPE2*X2 + &SLOPE3*X3 + &SLOPE4*X4 + &SLOPE1*X5 + &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);
```

By construction, X5 has unequalslopes.

The next slides will show the results of running CUM_LOGIT_SCREEEN 1 and 2

Example of CUM_LOGIT_SCREEN_1

%CUM_LOGIT_SCREEN_1(SIM_1, Y, C1 X1 X2 X3 X4 X5, LRCS);

Obs	Var_Name	Levels	Log_L_ Intercept	Log_Like lihood	LRCS	df	Pr>ChiSq (Num)	Pr > ChiSq	MODEL_ C	MODEL_ C_ASE
1	X5	2	-8209.1	-7822.8	772.64	2	0.0000	<.0001	0.573	0.0048
2	X4	2	-8209.1	-8198.4	21.42	2	0.0000	<.0001	0.522	0.0048
3	X1	12	-8209.1	-8190.8	36.63	22	0.0260	0.0260	0.523	0.0055
4	X2	2	-8209.1	-8205.7	6.82	2	0.0331	0.0331	0.510	0.0048
5	C1	4	-8209.1	-8205.1	8.00	6	0.2383	0.2383	0.513	0.0054
6	X3	2	-8209.1	-8207.7	2.68	2	0.2613	0.2613	0.508	0.0048

The predictors are sorted by Pr > ChiSq (parameter=LRCS). The best ranked predictor is X5. The Model_c for X5 is 0.573 (best among the six). Predictor X5 would probably be retained for further analysis. A question to consider for X5 would be whether unequalslopes are required.

Example of CUM_LOGIT_SCREEN_2

%CUM_LOGIT_SCREEN_2(SIM_1, Y, X1 X2 X3 X4 X5, C1, YES);

Obs	Split_Point	Var_Name	Levels	NOMINAL	MONO TONIC	C_STAT	MODEL c	IV (Info Value)
1	A - B	C1	4	YES	N/A	n/a	0.515	0.003
2	B - C	C1	4	YES	N/A	n/a	0.516	0.003
3	A - B	X1	12	NO		0.518	0.524	0.011
4	B - C	X1	12	NO		0.517	0.532	0.014
5	A - B	X2	2	NO	YES	0.514	0.514	0.003
6	B - C	X2	2	NO	YES	0.506	0.506	0.001
7	A - B	X3	2	NO	YES	0.509	0.509	0.001
8	B - C	X3	2	NO	YES	0.507	0.507	0.001
9	A - B	X4	2	NO	YES	0.525	0.525	0.010
10	B - C	X4	2	NO	YES	0.523	0.523	0.009
11	A - B	X5	2	NO	YES	0.621	**0.621**	**0.240**
12	B - C	X5	2	NO	YES	0.500	**0.500**	**0.000**

Only predictor X5 has strength for any binary split.

(C-Stat = Model c for any binary predictor)

Predictor is X5 is very strong for split A v. BC (but weak for AB v. C). This is further reason to keep X5. The difference in strength of X5 for the 2 binary splits suggests that X5 could have unequalslopes.

After Screening a NOD Predictor … What is Next?

After weak predictors are eliminated based on screening …

- Often, Binning of NOD predictors (reducing number of levels)
  - Parsimony
  - Logical relationships (e.g. monotonicity)

See **APPENDIX A** for slides about %CUMLOGIT_BIN

%CUMLOGIT_BIN bins NOD predictors for Cum Logit (PO and PPO)

See Lund (2017) SESUG

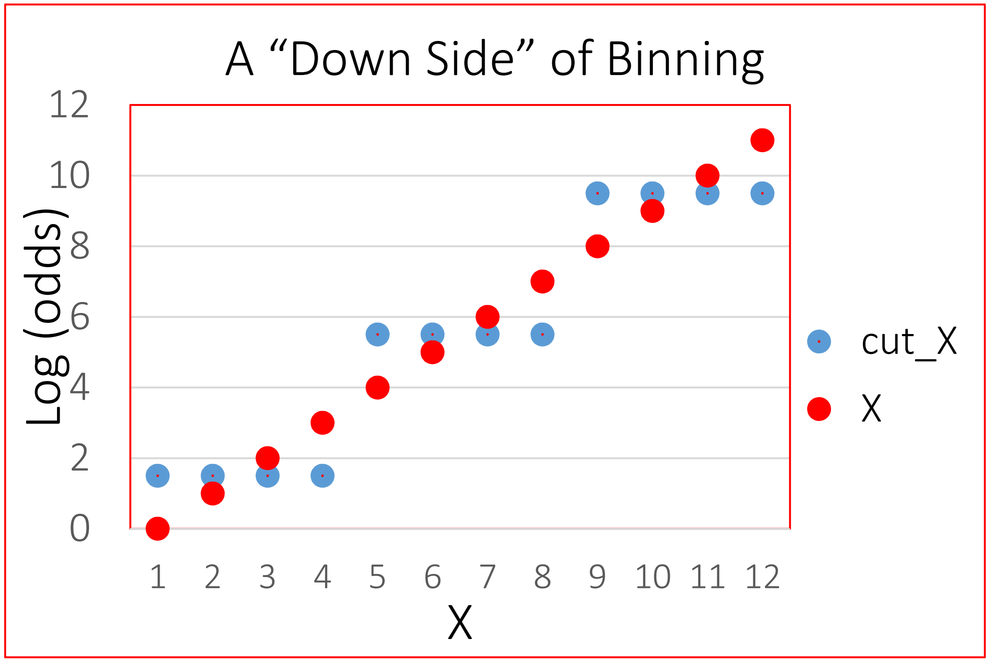
After Screening a NOD Predictor … What is Next?

Two Decisions for a NOD predictor (after any Binning)

1. WOE's or DUMMIES (creates different models)
2. EQUAL or UNEQUALSLOPES

See Lund (2017) SESUG for discussion

Logistic Regression Predictor X … Bin or Transform?



Binning is often advocated even when X is continuous. But …

- Binning ("cut-points") creates arbitrary discontinuities
- True trend may be obscured
- For small samples the bins may not validate on a Validation sample

If not Binned, X usually requires a transformation for best fit

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

- %FSP_8LR implements FSP for PO
- %FSP_8LR_PPO implements FSP for PPO

See **APPENDIX B** for slides about %FSP_8LR and %FSP_8LR_PPO

See Lund (2018) SGF for discussion

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 2013)

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

Fitting the Cumulative Logit Models (PO and PPO)

PROC LOGISTIC can “decide” on the use of UNEQUALSLOPES during predictor selection using FORWARD, STEPWISE, BACKWARD.

This requires a “trick”. For any predictor considered for unequalslopes, a duplicate predictor is created;

```
DATA WORK2; set WORK;
  X1_Duplicate = X1;          ★
  run;
PROC LOGISTIC DATA= WORK2;
MODEL Y= X1  X2  X3  X4  X1_Duplicate
/ UNEQUALSLOPES= (X1_Duplicate)
  SELECTION= FORWARD;          ★
run;
```

The predictor X1_Duplicate appears in UNEQUALSLOPES.

If X1_Duplicate enters the model by FORWARD selection, then X1 will ★ have unequalslopes in the model.

See Bob Derr (2013) SGF paper

Fitting the Cumulative Logit Models (PO)

PROC HPLOGISTIC and PROC HPGENSELECT:

Support only Cumulative Logit PO

- All predictor SELECTION options apply to Cumulative Logit PO
  e.g. SL, SBC, AIC, Validate, …, LASSO (hpgenselect)

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

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

Fitting the Cumulative Logit Models (PO and PPO)

★ See APPENDIX C for example using PROC LOGISTIC

+++++

★ In APPENDIX D …

Can HPLOGISTIC / HPGENSELECT be tricked in running PPO?
Yes, a data coding "trick" can make this work !!

This allows advanced SELECTION methods (SBC, LASSO, etc.) to be used for PPO models.

A robust testing plan is needed to determine limitations and issues.

onemagnify

insight to impact

Bruce Lund
blund@onemagnify.com
blund_data@mi.rr.com

APPENDIX A

%CUMLOGIT_BIN: Bins X for Cum Logit Model

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
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

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

Macro Call: %CUMLOGIT_BIN

DATASET: Data set to be processed

TARGET: Target with numeric or character levels

X: Predictor (numeric or character)

W: A frequency variable if present in DATASET. Otherwise enter 1

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

METHOD: IV or LL: Defines the rule for selecting an eligible pair for collapsing. Choices are TOTAL_IV (sum of the "cum split IV") and -2*LOG(L) Both IV and LL are computed for the saturated model

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.

%CUMLOGIT_BIN (BACKACHE, SEVERITY, AGE_GROUP, W, A, IV, , ,)

Bins	-2*LL	Total _IV	IV_1	IV_2	Corr_ woe
9	339.5	0.614	0.138	0.476	0.581
8	339.5	0.613	0.138	0.476	0.581
7	339.6	0.609	0.135	0.474	0.578
6	339.9	0.605	0.135	0.470	0.579
5	340.0	0.598	0.134	0.464	0.578
4	341.0	0.561	0.133	0.429	0.638
3	342.4	0.493	0.111	0.381	0.607
2	350.4	0.324	0.103	0.221	1

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

MODE = A
METHOD = IV (i.e. TOTAL_IV)

Stopping at Step=5 because IV drops past Step 5 (*)

WOE1 and WOE2 are computed by %CUMLOGIT_BIN.

WOE Correlation = 0.578 (modest)

(*) Needed: Good Stopping Rules

Unequalslopes is Indicated for AGE_GROUP

```
PROC LOGISTIC DATA = BACKACHE_5;
CLASS  AGE_GROUP;
MODEL SEVERITY = AGE_GROUP;
```

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	7.553	4	0.109

```
PROC LOGISTIC DATA = BACKACHE_5;
CLASS  AGE_GROUP;
MODEL SEVERITY = AGE_GROUP
/ UNEQUALSLOPES = (AGE_GROUP);
```

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	17.086	8	0.029

Model Comparison Test:

ChiSq (4 d.f.) = 17.086 - 7.553 = 9.533

Pr > ChiSq = 0.049

UNEQUALSLOPES is significant.

Enter Binned Age_Group using Dummies

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","36andUP") then Age_group_bin = 5;

This code is produced by the Macro

```
DATA BACKACHE_5; SET BACKACHE;
<insert code above>;
PROC LOGISTIC DATA = BACKACHE_5;
CLASS Age_group_bin  <>;
MODEL Y = Age_group_bin  <>
          / Unequalslopes = (Age_group_bin  <> );
```

Enter Binned Age_Group using 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","36andUP" ) then Age_group_woe1 = -0.066691374 ;
if Age_group in ( "33to35","36andUP" ) then Age_group_woe2 = -1.174985267 ;
```

```
PROC LOGISTIC DATA = BACKACHE_5;
CLASS <>;
MODEL Y = Age_Group_woe1  Age_Group_woe2   <>
/ Unequalslopes=(Age_Group_woe1  Age_Group_woe2  <>);
```

This code is produced by the Macro

WOE and Dummies do not give the same model

🡐 Use DATA Step to insert this code.

WOE1 and WOE2 were moderately correlated (=0.578).

In the event the correlation was high (e.g. > 0.75), then one of the WOE's should be omitted

APPENDIX B

FSP: Looks for the best transformation of X

- First, translate X (if needed) to make min(X) at least 1.
- Form the *Fractional Polynomials* (FP) as the transforms of X:

  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.

- Two groups of transforms are created: FP1 and FP2

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

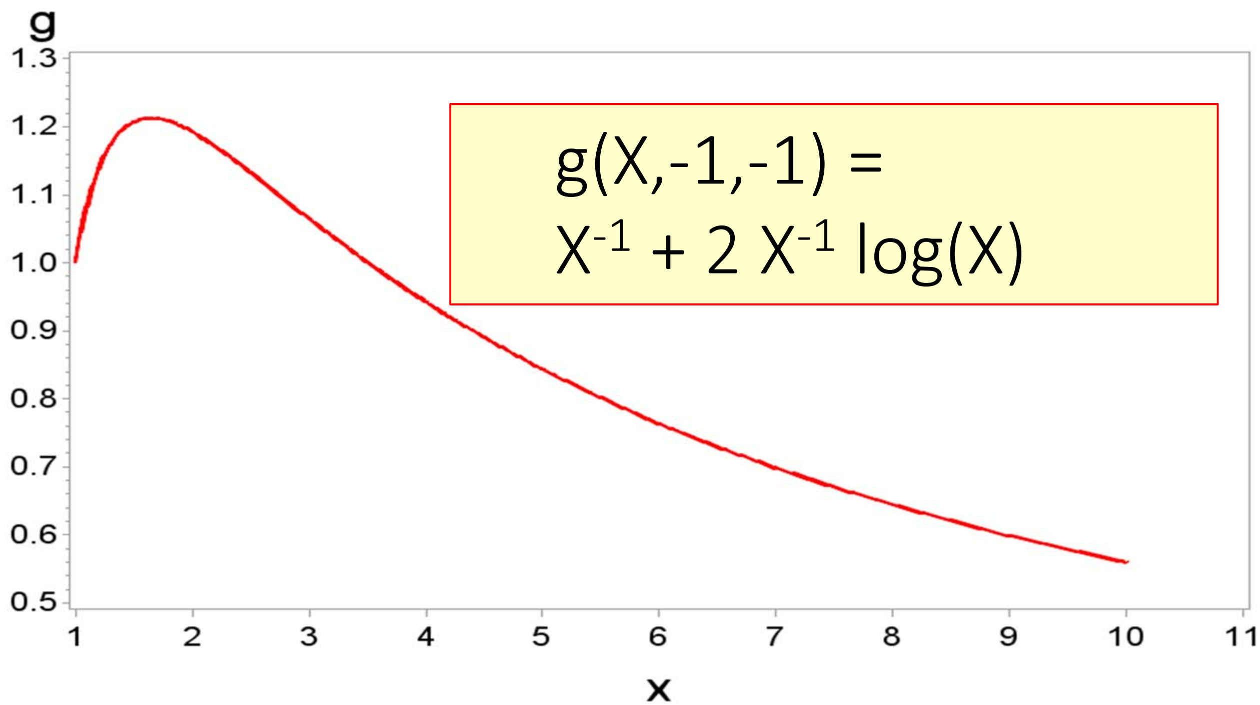
  36 FP2:

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

Shapes of FP2

FP2 transforms can produce non-monotonic curves (recall: $X \geq 1$)



$$g(X,-1,-1) = X^{-1} + 2\,X^{-1}\log(X)$$

FP1 produces only monotonic curves because $X \geq 1$ and $g(X,p) = \beta_0 + \beta_1 X^p$

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

Significance Testing (More discussion and example on later slide):

- Performs a 3-step test to determine the **final** transform …

Test Statistics for 3-Step Testing

- Royston and Sauerbrei provide Test-Statistics for Binary Logistic for 3-Step Testing
- These Test-Statistics extend to Cum Logit Model
  - Justification is based on simulations (… see my SGF Paper)

Code Generates an Cum Logit Example Dataset

Example Dataset ➔

Makes data for a cumulative logit

- Target Y with 3 Levels
- Predictor X
- Dataset is an “FP2” model:
  - $0.2*LOG(X) - 0.5*X^{-1}$ + error

Now, apply %FSP_8LR to this data.

```
%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);
```

Macro Call

%FSP_8LR (DATASET, TARGET, INPUT, VERBOSE, ORDER);

Parameter definitions:

DATASET: The data set containing the target and predictors

TARGET: Target variable (character or numeric). >= 2 levels

INPUT: Numeric predictors (at least 1). Delimited by a space
e.g. INPUT = X W A1 - A6

VERBOSE: YES … “YES" produces more output

ORDER: A | D … Default is A. Order for modeling the TARGET
(A=ascending, D=descending)

%FSP_8LR (WORK_1, Y, X, NO, A);

Summary Report - 3 Step Testing

TEST	-2*Log(L)	TEST _STAT	d f	P-VALUE	Trans 1	Trans 2
Eliminate X	15824.5	172.0	4	0.000		
Use Linear	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

Recall: Data was generated by X^{-1} and Log(X)

STEP 1: Test for "Eliminate X"

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"

%FSP_8LR (WORK_1, Y, X, NO, A);

TEST	-2*Log(L)	TEST _STAT	d f	P-VALUE	Trans 1	Trans 2
Eliminate X	15824.5	172.0	4	0.000		
Use Linear	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: Use X (linear)? ... NO, P-Value = 0

» STEP 3: Use FP1 Solution? ... YES, P-Value = 0.387

- or FP2 Solution ? ... NO, See above

... Final solution is $X^{-0.5}$

Proportional Odds (PO) Assumption

PROC LOGISTIC gives a Test of “Proportional Odds”.

This test is included in %FSP_8LR Report (See next slide)

If test fails (rejects PO):

- The predictor slopes “coefficients” may have different values across equations

➔ In this case, consider *Partial* PO Model (PPO)

Testing the Proportional Odds (PO) Assumption

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 Rejection

Perhaps “unequalslopes” is needed for $X^{-0.5}$

FP1 Solution with PPO

```
DATA TEST; SET WORK;
  g = X**(-0.5);
run;
PROC LOGISTIC Data = TEST;
  MODEL Y = g / UNEQUALSLOPES =(g);
run;
```

Parameter	Y	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	A	1	0.814	0.054	224.99	<.0001
Intercept	B	1	1.931	0.062	959.06	<.0001
g	A	1	-1.352	0.118	130.85	<.0001
g	B	1	-1.542	0.126	150.71	<.0001

Why is %FSP_8LR Efficient?

%FSP_8LR: Process *Multiple* X's Efficiently

... All Translations / Transforms in ***2*** preliminary data steps

Runs PROC LOGISTIC *only 8 times* per predictor (not 44 times)

Finds best FP1 and (usually) best FP2 ... (see my paper for detail)

➔ When "best" FP2 is not found, the effect is immaterial

Practical to use %FSP_8LR to screen 50 X's in data with 5000 obs.

APPENDIX C

FORWARD Selection, Fitting PPO with PROC LOGISTIC

```
DATA Random;
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;
DATA RANDOM2; set RANDOM;
   X1_Duplicate = X1;
   run;
```

```
PROC LOGISTIC DATA= RANDOM2;
MODEL Y= X1  X2  X3  X4  X1_Duplicate
/ UNEQUALSLOPES= (X1_Duplicate)
  SELECTION= FORWARD SLE= 0.15;
run;
```

Analysis of Maximum Likelihood Estimates				
Parameter	Y	DF	Estimate	Pr > ChiSq
Intercept	A	1	2.250	<.0001
Intercept	B	1	3.982	<.0001
X3		1	-5.170	<.0001
X1_Duplicate	A	1	-1.162	<.0001
X1_Duplicate	B	1	-0.674	<.0001

X1 is “selected” to have unequalslopes

APPENDIX D

DATA CODING TRICK

```
DATA Random;
Do ID = 1 to 5000;
   random = ranuni(1);
   If random < .5 then Y = "A";
   else if random < .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;
```

The data recoding Trick:

The Split Variable identifies the split for the cum logits.
Split=0: The split of A vs BC
Split=1: The split of AB vs B

For Split=0
Target = 0 for A, else Target=1 for BC
For Split=1
Target = 0 for AB, else Target=0 for C

Two observation are output for each input observation

```
DATA Recode; Set Random;
Do; if Y="A" then TARGET=0; else TARGET=1; Split=0; output; end;
Do; if Y="A" or Y="B" then TARGET=0; else TARGET=1; Split=1; output; end;
run;
```

DATA CODING TRICK

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

The DATA CODING is used by SDK to fit CUM LOGIT PPO using **PROC GENMOD**.

PROC GENMOD is successful in fitting a PPO model because of the TRICK and the **REPEATED** statement which adjusts for correlation of the RESPONSE within ID. (In data set **Recode**, each ID has two values of the TARGET. See SDK for discussion)

Can the Trick be used so that PROC HPLOGISTIC, PROC HPGENSELECT can fit a Cum Logit PPO Model (but without a **REPEATED** statement)

This would be good, if true, since advanced Predictor Selection methods (SBC, LASSO, and more) could be employed.

HPLOGISTIC/HPGENSELECT fitting PPO with Trick

```
/* #1 */
PROC LOGISTIC DATA = Random;
MODEL Y =  X1 X2 X3 X4
/ UNEQUALSLOPES = X1;
/* #2 */
PROC HPLOGISTIC DATA = Recode;
CLASS Split;
MODEL Target =  Split X1 X2 X3 X4 X1*Split;
/* #3 */
PROC HPGENSELECT DATA = Recode;
CLASS Split;
MODEL Target =  Split X1 X2 X3 X4 X1*Split /
DISTRIBUTION = BINARY;
run;
```

	MODEL #1	MODEL #2	MODEL #3
Intercept A	2.2504	**2.2051**	**2.2051**
Intercept B	3.9821	**3.9326**	**3.9326**
X1 A	-1.1611	**-1.1240**	**-1.1240**
X1 B	-0.6728	**-0.6415**	**-0.6415**
X2	0.0332	**0.0308**	**0.0308**
X3	-5.1511	**-5.2855**	**-5.2855**
X4	-0.0427	**-0.0646**	**-0.0646**

Models #2 and #3 are equal.

They have coefficients similar to Model #1, the “gold standard”.

➔The proof of success is whether Models #2 and #3 produce very similar Prob’s as Model #1. This is true (not shown).

Mapping of Results to Coefficients

MODEL #3 LOGISTIC			
Analysis of Maximum Likelihood Estimates			
Parameter		DF	Estimate
Intercept		1	3.0686
Split	0	1	-0.8636
X1		1	-0.8827
X2		1	0.0308
X3		1	-5.285
X4		1	-0.0646
X1*Split	0	1	-0.2413

The tables below show the correspondence between Model #3 results and equivalent formulation to the Model #2 results

MODEL #2 Equivalent Formulation		Formula
Intercept A	2.2050	=3.0686 - 0.8636
Intercept B	3.9322	=3.0686 + 0.8636
X1 A	-1.1240	=-0.8827 - 0.2413
X1 B	-0.6414	=-0.8827 + 0.2413
X2	0.0308	
X3	-5.2850	
X4	-0.0646	

Tactics of using HPLOGISTIC, HPGENSELECT for PPO

- After employing the data coding trick, HPLOGISTIC and HPGENSELECT might give a good approximation to an "ideal" PPO Model.
- The benefit of using HPLOGISTIC and HPGENSELECT is the availability of the many predictor variable selection methods SBC, AIC, Validate (HPLOGISTIC), Lasso (HPGENSELECT) and the usage of validation samples.
- Once candidate models using HPLOGISTIC / HPGENSELECT are obtained, they would be refit using PROC LOGISTIC with UNEQUALSLOPES.

HPLOGISTIC/HPGENSELECT fitting PPO with Trick

Data coding trick *worked* for the example given on the prior slides.

But more testing is needed.

Does the method work when there are many predictors designated for having unequalslopes?

The next slide shows how HPLOGISTIC / HPGENSELECT predictor SELECTION methods could be used when a fitting PPO model

Selection: HPLOGISTIC, HPGENSELECT for PPO Model

```
PROC HPLOGISTIC DATA = Recode; CLASS Split;
MODEL Target =  SPLIT X1 X2 X3 X4 X1*SPLIT / INCLUDE=1;
SELECTION METHOD = FORWARD (SELECT=AIC CHOOSE=AIC STOP=NONE);
run;
PROC HPGENSELECT DATA = Recode LASSOSTEPS= 40; CLASS SPLIT;
MODEL Target =  SPLIT X1 X2 X3 X4 X1*SPLIT / INCLUDE=1 DISTRIBUTION = BINARY;
SELECTION METHOD = LASSO (CHOOSE=AIC STOP=NONE);
run;
```

- HPLOGISTIC with SELECT and CHOOSE by AIC
- HPGENSELECT with LASSO and CHOOSE by AIC

Same predictor selection by both.

- Next Step: Fit these predictors by PROC LOGISTIC with UNEQUALSLOPES = (X1)

	HPLOGISTC	HPGENSELECT
Intercept A	2.2052	2.1874
Intercept B	3.9327	3.9144
X1 A	-1.1241	-1.1170
X1 B	-0.6417	-0.6436
X2	0.0308	0.0254
X3	-5.3177	-5.2620
X4	n/a	n/a