MSH4

Fundamentals of Statistical Consulting

Week 8 **Complete and quasi-complete**

Jean Yang and Jennifer Chan

What are complete separation and quasi-complete separation?

Outcome variable is binary, the sample size is small, and some cells are empty.

Complete separation occurs when a linear combination of the predictors yield a *perfect prediction* of the response variable.

For example, if $X \leq 4$ then $Y = 0$. If $X > 4$ then $Y = 1$. Perfect prediction! $P(Y=1|X>4)=1$.

ML estimates do not exist. Some model parameters are actually infinite.

Y	0	0	0	0	0	0	1	1	1	1
X	1	2	3	4	4	4	5	6	7	8

Quasi-complete separation is similar to complete separation. The predictors yield a perfect prediction of the response variable for *most values of the predictors, but not all*.

For example, in the previous data set, for one of the values where X = 4, let Y = 1 instead of 0. Now, if X < 4 then Y = 0, if X > 4 then Y = 1, but if X = 4 then Y could be 0 or 1. *This overlap in the middle range of the data makes the separation quasi-complete.*

Y	0	0	0	0	0	**1**	1	1	1	1
X	1	2	3	4	4	4	5	6	7	8

Causes and remediation

- Separation occurs when *the data set is too small* to observe events with low probabilities.
- The more predictors, particularly interaction terms, are in the model, the more likely separation is to occur because individual groups in the data have smaller sample sizes.
- The more predictors are in the model the more difficult the identification of the cause of the separation is.

The maximum likelihood estimates can fail to converge and result in large se, even without warning. 5 strategies:

1. ***Consider what the separation means.*** Complete separation and quasi-complete separation can indicate if the true probability of an event at a particular level or combination of levels is close to 0 or 1, this information is important.
2. ***Consider an alternative model.*** Check whether the exclusion of a term allows the maximum likelihood estimates to converge. If a useful model exists that does not use the term, you can continue the analysis with the new model.

3. ***Check whether you can combine categories in problematic variables.*** If there are categories that are sensible to combine, the separation can disappear from the data set.

Table 1. Data with complete separation

Fruit	Events	Trials
Grapefruit	0	10
Oranges	5	100
Apples	25	100
Bananas	40	100

Table 2. Data with overlap

Fruit	Events	Trials
Citrus	5	110
Apples	25	100
Bananas	40	100

	Grapefruit	Oranges	Apples	Bananas
Not	10	95	75	60
Event	0	5	25	40

Both	Apples	Bananas
105	75	60
5	25	40

4. ***Check if a problematic categorical variable is aggregated.*** If the relationship of the unaggregated variable to the response does not show complete separation, the substitution of the numeric data can eliminate the separation.

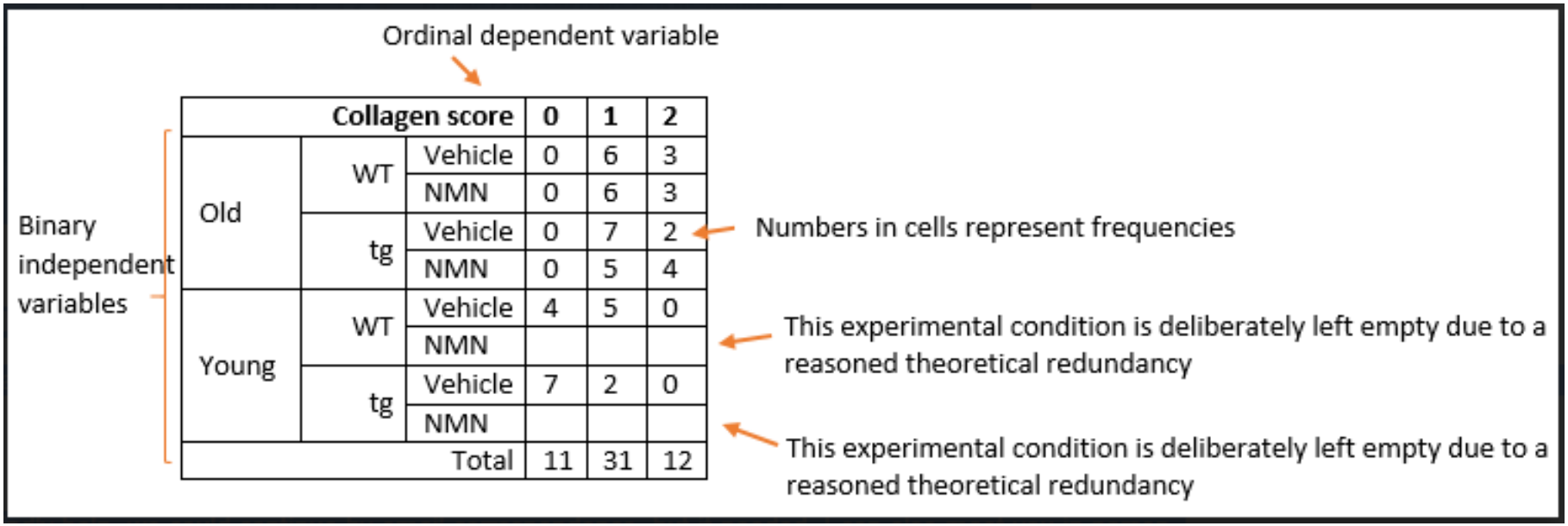
Table 3. Data with complete separation

Categories of length	Events	Trials
1–90	2	2
91–180	1	2
181–270	1	2
271–360	0	2

Exact length	Events	Trials
45	1	1
60	1	1
95	1	1
176	0	1
185	0	1
241	1	1
280	0	1
299	0	1

	1-90	91-180	181-270	271-360
Not	0	1	1	2
Event	2	1	1	0

> It’s like a perfect result with 100% certainty?



Collagen score			0	1	2
Old	WT	Vehicle	0	6	3
		NMN	0	6	3
	tg	Vehicle	0	7	2
		NMN	0	5	4
Young	WT	Vehicle	4	5	0
		NMN			
	tg	Vehicle	7	2	0
		NMN			
Total			11	31	12

Clearly, there are many 0 cells even after collapsing 0&1 2.

Combine outcome categories?

Collagen score			0-1	2	Total n
Old	WT	Vehicle	6	3	9
		NMN	6	3	9
	tg	Vehicle	7	2	9
		NMN	5	4	9
Young	WT	Vehicle	9	0	9
		NMN	-	-	-
	tg	Vehicle	9	0	9
		NMN	-	-	-

Exclusion of factors?

Collagen score		0	1	2
Old	WT	0	12	6
	tg	0	12	6
Young	WT	4	5	0
	tg	7	2	0

Collagen score		0	1	2
Old	Vehicle	0	13	5
	NMN	0	11	7
Young	Vehicle	11	7	0
	NMN	-	-	-

Collagen score		0	1	2
WT	Vehicle	4	11	3
	NMN	7	9	2
tg	Vehicle	11	7	0
	NMN	0	5	4

All still have some 0 cells.

Analyses you might consider

- ***Exact logistic regression*** - It is appropriate when the outcome variable is binary, the sample size is small, and some cells are empty. It is only practical in very simple cases, where there is just one predictor and becomes computationally intensive with additional (continuous) predictors and is rendered impractical. Furthermore, exact method adjusts only the p-value, not the parameter estimates.

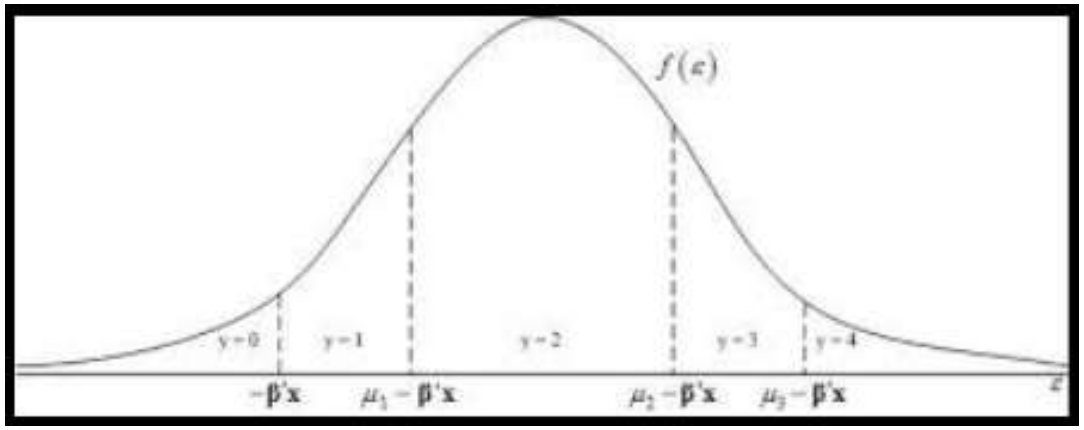
  Read my GLM lecture note section 5.6.1.

- ***Penalized likelihood method*** - proposed by Firth (1993 Biometrika 80:27-38). It adds a bias correction term which will goes to zero as the sample size increases. It provides finite and consistent estimates with separation. Use logistf package with the logistf() function.

- ***Multinomial logistic regression:*** The difference from log-linear model with Poisson distribution is that Poisson regression has the overall n fixed whereas multinomial regression has both the marginal (n_i = 9) and overall n fixed. Multinomial logistic regression differs from the ordinal logistic regression in

that the categories are nominal with no order so the ordering information is lost. Use MASS package with the multinom() function.

- ***Ordinal logistic regression:*** Take order information.



$$\text{logit}\big(P(Y \leq j)\big) = \alpha_j + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

The underlying distribution is logistic. Use package VGAM with vglm().
For proportional odds logistic regression, use package MASS with polr().

- ***Ordinal probit regression:*** Similar to ordered logistic regression but the underlying distribution is normal. Hence the interpretation of the coefficients differs. Use package MASS with polr()

>#Multinominal regression

```
> y0=c(0,0,0,0,4,7)
> y1=c(6,6,7,5,5,2)
> y2=c(3,3,2,4,0,0)
> y=cbind(y0,y1,y2)
> age=c(0,0,0,0,1,1)
> treat=c(0,0,1,1,0,1)
> veh=c(0,1,0,1,0,0)

> library(nnet)
> m1=multinom(y ~ age + treat + veh)
# weights:  15 (8 variable)
initial  value 59.325064
iter  10 value 34.143556
iter  20 value 33.616055
iter  30 value 33.615699
iter  40 value 33.614776
iter  50 value 33.613968
final  value 33.613966
```

converged

```
> summary(m1)
Call:
multinom(formula = y ~ age + treat + veh)

Coefficients:
   (Intercept)    age        treat      veh
y1    12.35728 -12.13415 -1.475828 4.006669
y2    11.40172 -21.80111 -1.475797 4.510235

Std. Errors:
   (Intercept)   age        treat      veh
y1    81.61039  81.60866  1.045370 662.7485
y2    81.61219 111.91114  1.264834  662.7486

Residual Deviance: 67.22793
AIC: 83.22793
> z <- summary(m1)$coefficients/summary(m1)$standard.errors
> p <- (1 - pnorm(abs(z), 0, 1))*2
```

```
> p
  (Intercept)       age             treat           veh
y1   0.8796460    0.8818006    0.1580160    0.9951764
y2   0.8888922    0.8455438    0.2432947    0.9945702

> cbind(fitted(m1),apply(fitted(m1),1,sum))
          y0             y1             y2
1  3.104393e-06   0.7222309   2.777660e-01   1
2  4.779131e-08   0.6111146   3.888853e-01   1
3  1.358039e-05   0.7222172   2.777692e-01   1
4  2.090679e-07   0.6111072   3.888926e-01   1
5  4.444415e-01   0.5555450   1.353367e-05   1
6  7.777621e-01   0.2222325   5.413987e-06   1
```

```
>#ordinal logistic regression
> m1 <- vglm(y~ age+treat+veh, family=cumulative)
> summary(m1)

Call:
vglm(formula = y ~ age + treat + veh, family = cumulative)

Pearson residuals:
  logit(P[Y<=1])  logit(P[Y<=2])
1   5.243e-07     -3.721e-01
2  -1.668e-06      3.419e-01
3   1.097e-06      3.721e-01
4  -3.489e-06     -3.419e-01
5   1.894e-10      8.963e-17
6  -2.264e-10     -2.301e-15

Coefficients:
                  Estimate     Std. Error     z value  Pr(>|z|)
(Intercept):1  -2.506e+01     2.352e+04      -0.001     0.999
```

(Intercept):2	9.555e-01	6.354e-01	1.504	0.133
age:1	2.484e+01	2.352e+04	0.001	0.999
age:2	2.232e+01	1.620e+04	0.001	0.999
treat:1	1.476e+00	1.045e+00	1.412	0.158
treat:2	-5.805e-11	7.121e-01	0.000	1.000
veh:1	1.981e+00	2.526e+04	0.000	1.000
veh:2	-5.035e-01	7.146e-01	-0.705	0.481

Number of linear predictors: 2
Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])
Dispersion Parameter for cumulative family: 1
Residual deviance: 0.5127 on 4 degrees of freedom
Log-likelihood: -7.9124 on 4 degrees of freedom
Number of iterations: 20

Exponentiated coefficients:

age:1	age:2	treat:1	treat:2	veh:1
6.109723e+10	4.936445e+09	4.375000e+00	1.000000e+00	7.246426e+00

veh:2

6.043956e-01

```
>#proportional ordinal logistic regression
> m0 <- vglm(y~ age+treat+veh, family=cumulative(parallel=TRUE))
> summary(m0)

Call:
vglm(formula = y ~ age + treat + veh, family = cumulative(parallel = TRUE))

Pearson residuals:
  logit(P[Y<=1])    logit(P[Y<=2])
1    -2.648e-07     -3.944e-02
2    -2.054e-07      6.890e-01
3    -3.395e-07      4.408e-02
4    -2.626e-07     -7.284e-01
5    -6.518e-01      1.664e-07
6     6.891e-01      1.295e-07

Coefficients:
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept):1	-21.8164	5019.9912	-0.004	0.997
(Intercept):2	0.7210	0.5887	1.225	0.221
age	22.0271	5019.9912	0.004	0.996
treat	0.4964	0.5791	0.857	0.391
veh	-0.5104	0.7196	-0.709	0.478

Number of linear predictors: 2
Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])
Dispersion Parameter for cumulative family: 1
Residual deviance: 1.9275 on 7 degrees of freedom
Log-likelihood: -8.6198 on 7 degrees of freedom
Number of iterations: 19

Exponentiated coefficients:

age	treat	veh
3.683222e+09	1.642786e+00	6.002739e-01

> mp013 <- vglm(y~ treat+veh, family=cumulative)

```
> summary(mp013)
Call:
vglm(formula = y ~ treat + veh, family = cumulative)

Pearson residuals:
  logit(P[Y<=1])    logit(P[Y<=2])
1    -1.461e+00       -1.5490
2    -6.412e-06        0.3419
3    -2.293e+00       -0.4247
4    -9.121e-06       -0.3419
5     1.413e+00        1.0695
6     2.357e+00        0.9024

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1 -1.210e+00  5.570e-01  -2.173  0.0298 *
(Intercept):2  1.825e+00  5.906e-01   3.090  0.0020 **
treat:1        7.070e-01  7.293e-01   0.969  0.3323
treat:2       -1.208e-09  6.827e-01   0.000  1.0000
```

```
veh:1          -1.795e+01   1.705e+03   -0.011   0.9916
veh:2          -1.373e+00   6.827e-01   -2.011   0.0444 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Number of linear predictors:  2
Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])
Dispersion Parameter for cumulative family:   1
Residual deviance: 25.6817 on 6 degrees of freedom
Log-likelihood: -20.4969 on 6 degrees of freedom
Number of iterations: 16

Exponentiated coefficients:
    treat:1         treat:2             veh:1             veh:2
2.027855e+00  1.000000e+00  1.607696e-08  2.534562e-01
```

>#Firth regression combining y0 & y1

```
> y01=y0+y1
> s01=sum(y01)
> y01l=c(rep(0,s01),rep(1,s2))
> agel=c(rep(age,y0),rep(age,y1),rep(age,y2))
> treatl=c(rep(treat,y0),rep(treat,y1),rep(treat,y2))
> vehl=c(rep(veh,y0),rep(veh,y1),rep(veh,y2))
> y01l
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
> agel
 [1] 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
> treatl
 [1] 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1
> vehl
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 1
```

```
> library(logistf)
> m2= logistf(y01l ~ agel+treatl+vehl)
> summary(m2)
logistf(formula = y01l ~ agel + treatl + vehl)
```

Model fitted by Penalized ML
Confidence intervals and p-values by Profile Likelihood Profile Likelihood Profile Likelihood Profile Likelihood

	coef	se(coef)	lower 0.95	upper 0.95	Chisq	p
(Intercept)	2.774397e+00	1.124025	2.4015986	3.2747922	0.0000	1.00000000
agel1	-1.409251e+00	1.162930	-1.9093805	-1.0077691	0.0000	1.00000000
treatl1	-4.023437e-16	1.008135	-0.3900906	0.3900906	0.0000	1.00000000
vehl1	3.457638e+00	5.858853	-4.3379722	11.5045134	12.6157	0.00038252

Likelihood ratio test=-11.17222 on 3 df, p=1, n=54
Wald test = 2.074176 on 3 df, p = 0.5571547

Covariance-Matrix:

	[,1]	[,2]	[,3]	[,4]
[1,]	1.2634317	-1.009347e+00	-5.081685e-01	-1.009347e+00
[2,]	-1.0093475	1.352407e+00	-1.779544e-18	1.009347e+00
[3,]	-0.5081685	-1.779544e-18	1.016337e+00	-6.051921e-17
[4,]	-1.0093475	1.009347e+00	-6.051921e-17	3.432616e+01

```
>#Poisson regression
> y1 = c(0, 6, 3, 0, 6, 3, 0, 7, 2, 0, 5, 4, 4, 5, 0, 7, 2, 0)
> x1 = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
> x2 = c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1)
> x3 = c(0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0)
> x4 = rep(c(0, 1, 2), 6)
> x1=as.factor(x1)
> x2=as.factor(x2)
> x3=as.factor(x3)
> x4=as.factor(x4)
> poi <- glm(y1~x1+x2+x3+x4, family=poisson(link="log"))
> summary(poi)

Call:
glm(formula = y1 ~ x1 + x2 + x3 + x4, family = poisson(link = "log"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.00000  -1.91485  -0.03686   0.65787   2.90233

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  6.061e-01  3.827e-01   1.584  0.11324
x11         -8.973e-14  3.333e-01   0.000  1.00000
x21          3.456e-14  2.722e-01   0.000  1.00000
x31          2.620e-16  3.333e-01   0.000  1.00000
x41          1.036e+00  3.510e-01   2.952  0.00315 **
x42          8.701e-02  4.174e-01   0.208  0.83488
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 51.936  on 17  degrees of freedom
Residual deviance: 38.797  on 12  degrees of freedom
AIC: 90.431
Number of Fisher Scoring iterations: 6
```