Testing GOF & Estimating Overdispersion

When you read Chapter 5 of Cooch & White, you will find several important ideas.

CHAPTER 5

Goodness of fit testing...

On page 5-1, you'll read the following important point.

"… a necessary first step to insure that the most general model in your candidate model set (see Chapter 4) adequately fits the data. Comparing the relative fit of a general model with a reduced parameter model provides good inference only if the more general model adequately fits the data."

On page 5-6, you'll read a quote by Gary White that presents an important idea that he presented in an article that appeared in the Journal of Applied Statistics in 2002 (volume 29, pages 103-106).

> In a recent paper, Gary White commented:
>
> "*The Achilles' heel...in capture- recapture modeling is assessing goodness-of-fit (GOF). With the procedures presented by Burnham & Anderson (1998), quasi-likelihood approaches are used for model selection and for adjustments to the variance of the estimates to correct for over-dispersion of the capture-recapture data. An estimate of the over-dispersion parameter, c, is necessary to make these adjustments. However, no general, robust, procedures are currently available for estimating c. Although much of the goodness-of-fit literature concerns testing the hypothesis of lack of fit, I instead view the problem as estimation of c.*"

- **Your Most General Model Needs to Fit the Dataset**
    - model is a benchmark when evaluating other models
    - evaluate via Goodness-of-Fit (GOF) testing
    - Diagnostics and tests are less developed for generalized linear models

- **GOF**
    - Observed versus Expected values
        - Work withi m_{ij} array if have enough data

observed array

```
11   2   0   0   0   0
    24   1   0   0   0
        34   2   0   0
            45   1   2
                51   0
                    52
```

expected values under CJS model, i.e., phi(t), p(t) given estimates and R_i

```
11.0   1.9  0.1  0    0    0
      24.1  0.9  0.1  0    0
           34.0  1.8  0.1  0
                45.1  2.8  0.1
                     49.1  1.9
                          52.0
```

- In Program MARK, you can call Program RELEASE to look at breakdowns of data for CJS modeling
- For many data types, such tests simply don't exist
- For known-fate, can do tests such as the Hosmer-Lemeshow test and more modern but related tests

Model = $\dfrac{e^{12.351+0.497 \cdot Length}}{1+e^{12.351+0.497 \cdot Length}}$

Predict Survival for each animal in study, bin animals by predicted values (e.g., based on length), and use those values to calculate the expected number of survivors and deaths in each bin. Finally, compare observed and expected values for each fate in each bin, use the values to calculate a test statistic, and evaluate how probable such a test statistic value is under the null hypothesis that the model fits the data. It can be challenging to choose the binning, especially with multiple covariates, and expected values can get small, which can cause problems.

Length (cm)	Observed Survived	Expected Survived	Observed Died	Expected Died
<23.5	5	3.64	9	10.36
23.25-24.25	4	5.31	10	8.69
24.25-25.25	17	13.78	11	14.22
25.25-26.25	21	24.23	18	14.77
26.25-27.25	15	15.94	7	6.06
27.25-28.25	20	19.38	4	4.62
28.25-29.25	15	15.65	3	2.35
>29.25	14	13.08	0	0.92

Pearson $X^2 = 5.3$ and $G^2 = 6.2$. $df = 6$ (counts for 8 levels predicted from 2 parameters).
$P(X^2 \geq 5.3$ or $G^2 \geq 6.2)$ when model fits the data is > 0.4. Conclude no problem with GOF.

- **GOF**
  - Deviance
    - Need a baseline model that fits perfectly: use saturated model
    - Saturated model often not in model list but can be conceived of
      - E.g., CJS with 3 occasions and 2 releases
        - 6 data points:

lnL= $[Y_{111} \cdot \ln(Y_{111}/R_1) + Y_{110} \cdot \ln(Y_{110}/R_1) + Y_{101} \cdot \ln(Y_{101}/R_1) + Y_{100} \cdot \ln(Y_{100}/R_1) + Y_{011} \cdot \ln(Y_{011}/R_2) + Y_{010} \cdot \ln(Y_{010}/R_2)]$

EH	111	110	101	100	011	010
Y_i	6	46	21	427	20	80

$\ln L_{\text{saturated model}}$ = 6 ln(6/500) + 46 ln(46/500) + 21 ln(21/500) + 427 ln(427/500) + 20 ln(20/100) + 80 ln(80/100)

$\ln L_{\text{saturated model}}$ = -320.2945, $-2 \ln L_{\text{saturated model}}$ = 640.589

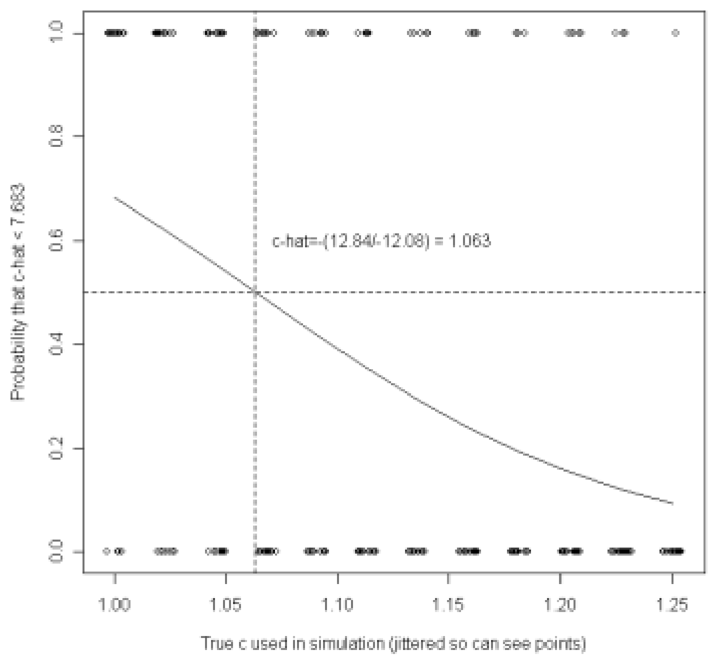
- Calculate deviance: $(-2\ln L_{\textit{fitted model}}) - (-2\ln L_{\textit{saturated model}})$
  - E.g., Deviance $\phi(t)p(t)$ = 642.41 - 640.589 = 1.8219
- Deviance measures how far from perfect fit you are
- Smaller deviance is desired, but how small is small enough?

  - Uses of Deviance in GOF
    - Likelihood Ratio Tests
    - Deviance/df
      - Assume distributed as chi-squared
      - *$P(X^2>1.82)$, 1 df* = 0.823
      - <u>Assumption not reasonable unless have large sample sizes</u>
    - Deviance residuals can be examined
    - Overdispersion and $\hat{c}$ = our focus

- **Overdispersion**
  - Two sources that relate directly to our IID assumptions
    - Lack of independence
    - Heterogeneity in rates
  - Effect is to underestimate variances
    - Leads to overfitting
    - Point estimates tend to be unbiased
  - Estimating $\hat{c}$ can be done for <u>*some (but not all)*</u> data types, e.g., CJS
    - Median c-hat procedure – example for Swifts data

	Value of true overdispersion used in simulation					
	1	**1.023**	**1.045**	**1.068**	**1.09**	**1.11**
$\hat{c}$ **< 7.683***	16	12	10	8	9	5
$\hat{c}$ **> 7.683***	4	8	10	12	11	15
Ppn. of sims with $\hat{c}$ **< 7.683***	0.80	0.60	0.50	0.40	0.45	0.25

***For real data, dev/df = 7.683**



Example of graphical output for another set of simulated data used to estimate $\hat{c}$

- QAICc for Model selection $QAICc = \frac{-2\ln L}{\hat{c}} + 2k\frac{2k(k+1)}{n-k-1}$

- Variances are inflated by $\hat{c}$

- More complex models are less supported as overdispersion increases