Delta Method for Difference in Survival Rates

WILD 502

Once you've produced an estimate of one or more parameters in a model of interest, you often want to use those estimates to obtain estimates of the mean and variance of transformations of those parameter estimates. Our attention here will be on estimating the variance of the transformed variable. One common approach is to use the Delta method, which propagates the errors or uncertainty about the estimated parameters into the variance of the transformed quantity based on those estimates or random variables. From page B-7 of CW: ". the Delta method rests on the assumption the first-order Taylor expansion around the parameter value is effectively linear over the range of values likely to be encountered."

In this week's lab, you will use the Delta method to produce estimates of the SE and 95% confidence limits for the difference in survival rate for swifts in the good and poor colonies. the Delta method uses a first-order Taylor series expansion. To do so, you need to bring two types of information together:

1. How uncertain are we of the estimates being used in the equation that generates the transformed quantity? This is measured by the appropriate values in the variance-covariance matrix.
2. How much does that uncertainty matter to the transformed quantity? This is measured using calculus, specifically derivatives.

The R code below shows how to do the calculations in several ways. You can read more about the Delta method in Appendix F of WNC (esp. F.4) & Appendix B of CW.

Difference in survival rates

The difference itself is easy to calculate.

$$\widehat{Diff} = \hat{S}_{good} - \hat{S}_{poor}$$

The SE and confidence limits take a bit more work becaus we have to take into account the variance of each rate and their covariance.

1. Obtain the partial derivatives of transformation with respect to each of the random variables involved:

$$\frac{\partial Diff}{\partial S_{good}} = 1 \cdot S_{good}^{1-1} = 1$$

$$\frac{\partial Diff}{\partial S_{poor}} = -1 \cdot S_{poor}^{1-1} = -1$$

2. Use matrix multiplication on the vector of derivatives and values in the variance-covariance matrix.

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} var(S_{good}) & cov(S_{good}, S_{poor}) \\ cov(S_{good}, S_{poor}) & var(S_{poor}) \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\widehat{Var}_{Diff} = \begin{bmatrix} 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0.0015590689 & 0.0001537251 \\ 0.0001537251 & 0.0061633030 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 0.007424921$$

$$\widehat{SE}_{Diff} = \sqrt{\widehat{var}_{Diff}} = 0.08616798$$

$$\widehat{Var}_{Diff} = var(S_{good}) + var(S_{poor}) - 2 \cdot cov(S_{good}, S_{poor})$$

R code for calculating the variance of a difference

If you know what the partial derivatives are, you can provide those and the variance-covariance matrix to R. Then, it's a simple matter of having R do the calculations. Here, you could just use $\widehat{Var}_{Diff} = var(S_{good}) + var(S_{poor}) - 2 \cdot cov(S_{good}, S_{poor})$ as we know the equation to use. But, if you had more complex derivatives to deal with, having software do the calculations can be helpful.

```
sigma <- matrix(c(0.0015690689, 0.0001537251, 0.0001537251, 0.006163303), 2, 2)
derivs <- matrix(c(1, -1), 2, 1)
VarDiff <- t(derivs) %*% sigma %*% derivs
sP <- 0.5770598
sG <- 0.7699526
Diff <- sG - sP
SE_diff <- sqrt(VarDiff)
lciDiff <- Diff - 1.96*SE_diff
uciDiff <- Diff + 1.96*SE_diff
round(SE_diff, 3)
```

```
##       [,1]
## [1,] 0.086
```

```
round(c(lciDiff, Diff, uciDiff), 3)
```

```
## [1] 0.024 0.193 0.362
```

If you don't know the partial derivatives or don't want to figure them out, you can use one of several packages (e.g., 'rmd' and 'emdbook') that have functions for the Delta method to do even more of the work for you. The example below uses the 'rmd' package.

```
library(msm)
sigma <- matrix(c(0.0015690689, 0.0001537251, 0.0001537251, 0.006163303), 2, 2)
sP <- 0.5770599
sG <- 0.7699526
Diff <- sG - sP
SE_Diff <- deltamethod(~x1 - x2, c(sG, sP), sigma, ses = TRUE)
lciDiff <- Diff - 1.96*SE_Diff
uciDiff <- Diff + 1.96*SE_Diff
round(SE_Diff, 3)
```

```
## [1] 0.086
```

```
round(c(lciDiff, Diff, uciDiff), 3)
```

```
## [1] 0.024 0.193 0.362
```