Non-existence of MLEs in logistic regression

Intro

- Won't always be able to rely on estimates from R or SAS in logistic regression
- This problem depends on the data structure
- Could result in very large estimated coefficients and standard errors

Example data set

- From Agresti (page 262)
- Response is whether subject achieved a 3-year disease-free interval (`dfi3`)

```
osteo <- data.frame(dfi3=c(rep(1,8),rep(0,3)),
                    lyinf=c(rep(0,4),rep(1,7)),
                    gender=c(0,0,1,1,0,0,1,1,0,1,1),
                    aop=c(rep(0:1,4),1,0,1),
                    count=c(3,2,4,1,5,3,5,6,2,4,11))
```

Look at the data

```
##    dfi3 lyinf gender aop count
## 1     1     0      0   0     3
## 2     1     0      0   1     2
## 3     1     0      1   0     4
## 4     1     0      1   1     1
## 5     1     1      0   0     5
## 6     1     1      0   1     3
## 7     1     1      1   0     5
## 8     1     1      1   1     6
## 9     0     1      0   1     2
## 10    0     1      1   0     4
## 11    0     1      1   1    11
```

- Outcome is `dfi3`
- The `count` variable gives the total number of patients with that set of covariate values and outcome

Initial models

```
mod.lyinf <- glm(dfi3 ~ lyinf,data=osteo,
                 family=binomial,weights=count)
mod.gender <- glm(dfi3 ~ gender,data=osteo,
                  family=binomial,weights=count)
mod.aop <- glm(dfi3 ~ aop,data=osteo,
               family=binomial,weights=count)

univmod.tab <- data.frame(lyinf=coef(mod.lyinf),
                          gender=coef(mod.gender),
                          aop=coef(mod.aop))
rownames(univmod.tab) <- c('intercept','slope')
univmod.tab
```

```
##             lyinf gender    aop
## intercept  18.06  1.872  1.447
## slope     -17.95 -1.807 -1.527
```

What happened?

- Estimates for `gender` and `aop` seem reasonable
- Estimates for `lyinf` are very large

Model for lyinf

```
##
## Call:
## glm(formula = dfi3 ~ lyinf, family = binomial, data = osteo,
##     weights = count)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -4.063  -0.866   0.000   2.243   2.769
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)     18.1     1602.9    0.01     0.99
## lyinf          -18.0     1602.9   -0.01     0.99
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 60.603  on 10  degrees of freedom
## Residual deviance: 49.795  on  9  degrees of freedom
## AIC: 53.8
```

Numerical checking

- We can adjust aspects of the model fitting procedure to examine convergence
- Number of iterations (Fisher scoring); default is 25
- Tolerance (how close together two sequential iterations need to be with respect to log-likelihood values); default is `epsilon = 1e-8`

Control parameters

```
mod.lyinf1 <- glm(dfi3 ~ lyinf,data=osteo,
                  family=binomial,weights=count,
                  control=list(maxit=1000,epsilon=1e-4))
mod.lyinf2 <- glm(dfi3 ~ lyinf,data=osteo,
                  family=binomial,weights=count,
                  control=list(maxit=1000,epsilon=1e-10))
mod.lyinf3 <- glm(dfi3 ~ lyinf,data=osteo,
                  family=binomial,weights=count,
                  control=list(maxit=1000,epsilon=1e-16))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Results of varying tolerance: LL

```
logLik(mod.lyinf1)
```

```
## 'log Lik.' -24.9 (df=2)
```

```
logLik(mod.lyinf2)
```

```
## 'log Lik.' -24.9 (df=2)
```

```
logLik(mod.lyinf3)
```

```
## 'log Lik.' -24.9 (df=2)
```

Results of varying tolerance: number of iterations

```
summary(mod.lyinf1)$iter
```

```
## [1] 7
```

```
summary(mod.lyinf2)$iter
```

```
## [1] 21
```

```
summary(mod.lyinf3)$iter
```

```
## [1] 32
```

Parameter estimates

- Log-likelihood has apparently converged, even if we increase `iter`
- What happens to the parameter estimates?

```
##   tolerance X.Intercept.  lyinf
## 1     1e-04        9.061  -8.95
## 2     1e-10       23.062 -22.95
## 3     1e-16       33.893 -33.78
```

- Going to negative and positive infinity
- This suggests that the MLEs do not exist for this model

Separation of data points

- The problem is that all of the failures (`dfi3=0`) have the same value of the binary covariate `lyinf=1`
- Corresponds to an empty cell in the 2x2 table
- This is called separation: happens when "a logistic model perfectly or nearly perfectly predicts the response (that is, separates the response levels)" (http://support.sas.com/kb/22/599.html)

```
##       dfi3
## lyinf  0  1
##     0  0 10
##     1 17 19
```

What to do about this

- Wald tests are invalid: likelihood surface is too flat, leading to huge standard errors from information matrix
- Likelihood ratio and score tests should still be okay
- Fisher's exact test (no other covariates) or exact logistic regression (computationally intensive)
- Firth correction: will give estimates (not just inference) and is easy to implement

LR test

```
anova(mod.lyinf,test='LRT')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: dfi3
##
## Terms added sequentially (first to last)
##
##
##       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                     10       60.6
## lyinf  1     10.8         9       49.8    0.001 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Rao) score test

```
anova(mod.lyinf,test='Rao')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: dfi3
##
## Terms added sequentially (first to last)
##
##
##       Df Deviance Resid. Df Resid. Dev  Rao Pr(>Chi)
## NULL                     10       60.6
## lyinf  1     10.8         9       49.8 7.49   0.0062 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adding covariates

- What if we want to adjust for our other covariates?

```
fullmod <- glm(dfi3 ~ lyinf+gender+aop,data=osteo,
                 family=binomial,weights=count)
summary(fullmod)$coef
```

```
##             Estimate Std. Error   z value Pr(>|z|)
## (Intercept)   20.489  2469.2231  0.008298  0.99338
## lyinf        -18.381  2469.2229 -0.007444  0.99406
## gender        -1.636     0.9123 -1.793504  0.07289
## aop           -1.220     0.7712 -1.582479  0.11354
```

- Still have the problem with `lyinf`

Type II tests

- R has a function to drop each term from the model and calculate the log-likelihood
- Can use this for likelihood ratio testing or score testing
- Likelihood surface is well behaved near the null hypothesis (beta=0)

Type II LRT

```
drop1(fullmod,test='LRT')
```

```
## Single term deletions
##
## Model:
## dfi3 ~ lyinf + gender + aop
##        Df Deviance  AIC  LRT Pr(>Chi)
## <none>       42.8 50.8
## lyinf   1     49.7 55.7 6.91   0.0085 **
## gender  1     46.5 52.5 3.72   0.0537 .
## aop     1     45.4 51.4 2.64   0.1045
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type II Score test

```
drop1(fullmod,test='Rao')
```

```
## Single term deletions
##
## Model:
## dfi3 ~ lyinf + gender + aop
##        Df Deviance  AIC Rao score Pr(>Chi)
## <none>        42.8 50.8
## lyinf   1     49.7 55.7      4.88    0.027 *
## gender  1     46.5 52.5      3.57    0.059 .
## aop     1     45.4 51.4      2.62    0.106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fisher's exact test

```
##
##  Fisher's Exact Test for Count Data
##
## data:  xtabs(count ~ lyinf + dfi3, data = osteo)
## p-value = 0.008
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0000 0.6055
## sample estimates:
## odds ratio
##          0
```

- 0 estimate for odds ratio
- Still have a finite upper confidence limit
- Fisher test is good for small samples, but doesn't allow for adjustment for other covariates

Confidence intervals

- Wald CI won't work
- What if we want to get profile likelihood CI?

```
## Waiting for profiling to be done...
```

```
##                 2.5 %  97.5 %
## (Intercept) -125.299      NA
## lyinf             NA 155.854
## gender        -3.700   0.025
## aop           -2.827   0.249
```

- Still seems like there is a problem with built-in functionality, so we want to do this manually

Define profile deviance

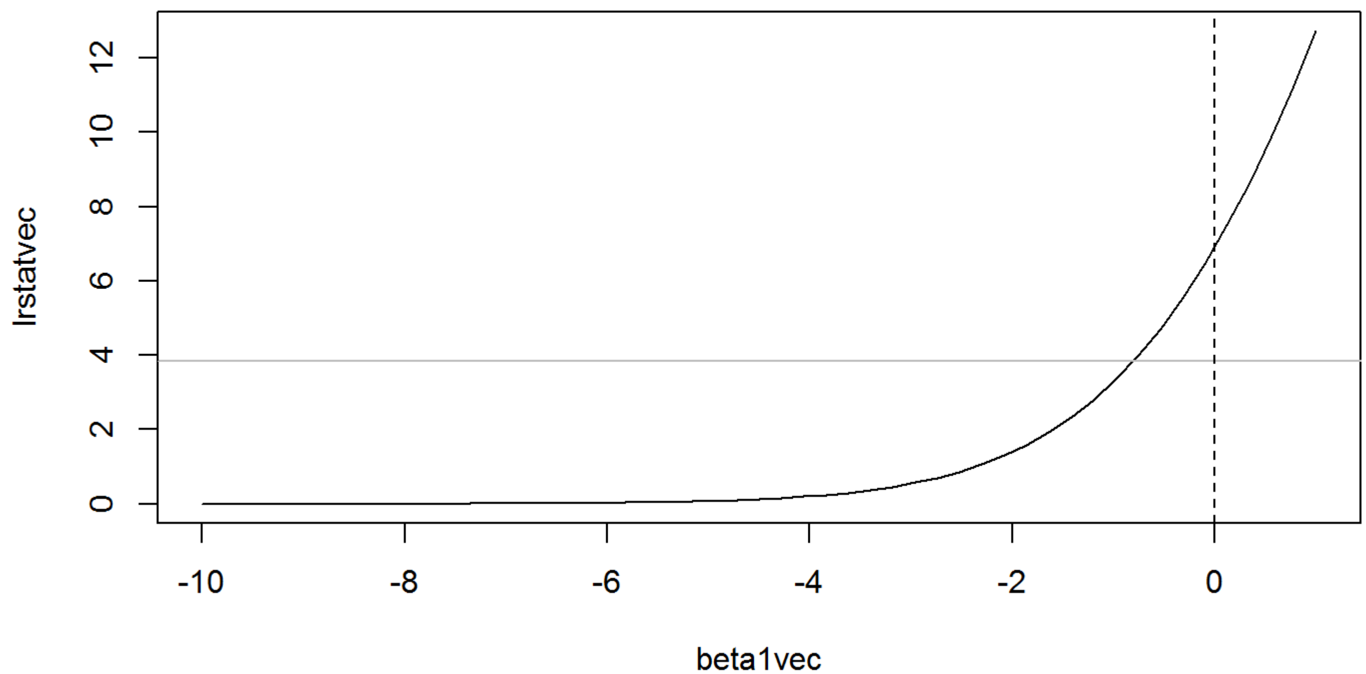
```
pr.lrt <- function(beta1) {
  2*(logLik(longmod1) -
       logLik(update(longmod1,
                     formula.=~.-lyinf+offset(beta1*lyinf),
                     data=data.frame(osteo.long,beta1=beta1))))
}
pr.lrt(0)
```

```
## 'log Lik.' 6.915 (df=4)
```

```
drop1(longmod1,test='LRT')$LRT[2]
```

```
## [1] 6.915
```

Look at the profile deviance function



24/34

Confidence interval

- Can tell from plot that there is a point to the left of 0 where we would no longer reject
- The value of the deviance falls under the chi-square(1 df) critical value
- Use `uniroot()` to find this
- This is an upper 95% profile likelihood confidence limit for the coefficient of `lyinf`

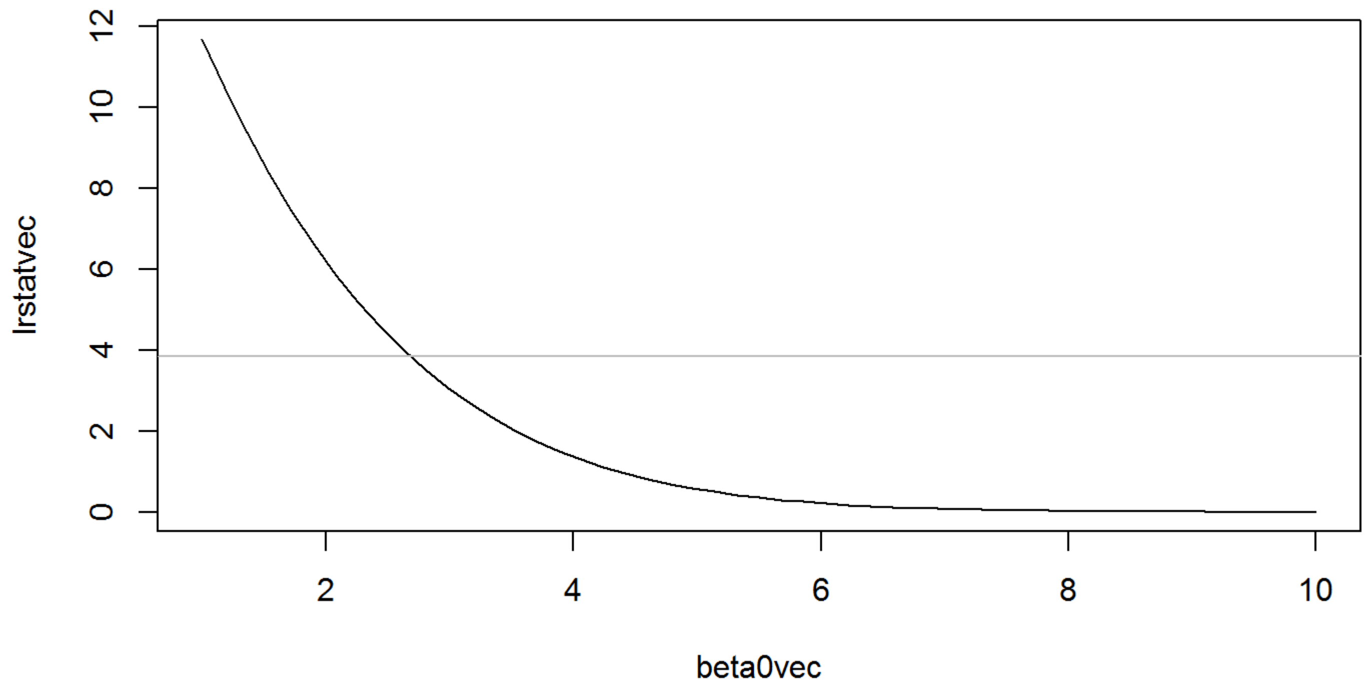
```
upper.lrci.lyinf <- uniroot(function(x)
  pr.lrt(x)-qchisq(.95,1),c(-10,0))$root
```

Compare with Fisher test

- Maybe not a fair comparison since full model adjusts for two additional covariates
- p-value from LRT is 0.0085
- p-value from Fisher test is 0.0075
- Look at odds ratio scale
- Upper confidence limit is 0.4466
- From Fisher test (not adjusting for covariates), it is 0.6055

Intercept

- We can do this for the intercept too (although generally of less interest)

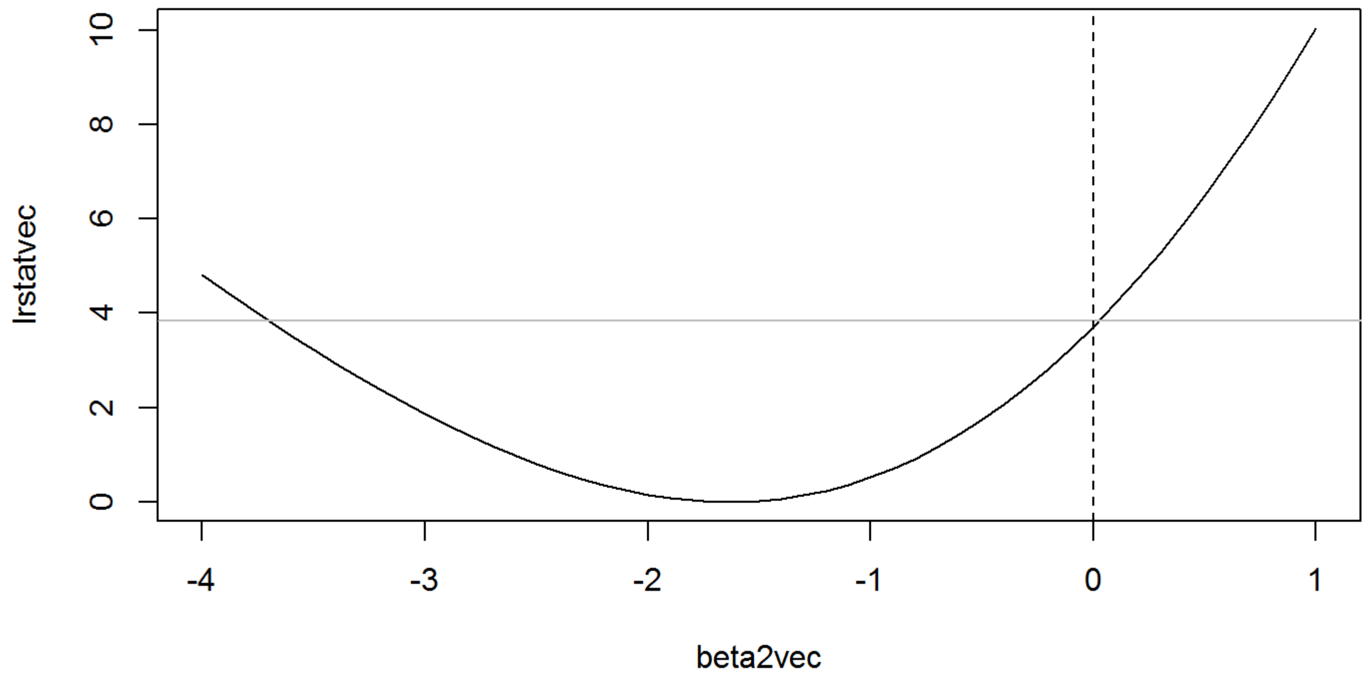


27/34

CI for intercept

- Profile likelihood lower confidence limit for intercept is 2.6793

Gender profile deviance



Gender LR CI

- Always want to make sure that what we are doing manually agrees with built-in functions
- Our deviance function value at 0 (equal to LR test stat for gender): 3.721
- Built-in LR test stat: 3.721
- CI from ur version of the deviance function: -3.6997, 0.0251
- Built-in CI: -3.6999, 0.025

Alternative methods

- This still doesn't give us estimates for the parameter(s) affected by separation
- There is a correction to the log-likelihood that will help (Firth penalization)
- This is available in the R package `logistf`

```
longmod2 <- logistf(dfi3 ~ lyinf+gender+aop,
                data=osteo.long)
```

Firth results

```
## logistf(formula = dfi3 ~ lyinf + gender + aop, data = osteo.long)
##
## Model fitted by Penalized ML
## Confidence intervals and p-values by Profile Likelihood Profile Likelihood Profile Likeli
##
##                coef se(coef) lower 0.95 upper 0.95  Chisq          p
## (Intercept)  4.290   1.6633      1.814     9.3243 16.601 4.612e-05
## lyinf       -2.461   1.5525     -7.363    -0.1887  4.660 3.088e-02
## gender      -1.415   0.8441     -3.251     0.1151  3.265 7.075e-02
## aop         -1.104   0.7316     -2.604     0.2807  2.431 1.189e-01
##
## Likelihood ratio test=14.18 on 3 df, p=0.002666, n=46
## Wald test = 8.075 on 3 df, p = 0.04449
##
## Covariance-Matrix:
##         [,1]     [,2]     [,3]     [,4]
## [1,]  2.7665 -2.19053 -0.53217 -0.25328
## [2,] -2.1905  2.41034 -0.03159 -0.10045
## [3,] -0.5322 -0.03159  0.71254  0.01478
## [4,] -0.2533 -0.10045  0.01478  0.53520
```

Compare with likelihood results

- Look at coefficient estimate for `lyinf`, with `gender` and `aop` already in the model
- From Firth model: p-value is 0.0309, CI is -7.3626, -0.1887
- LRT p-value is 0.0085, upper confidence limit is -0.806
- Score p-value is 0.0272 (more conservative than LRT)

Conclusions

- Non-existence of MLEs is most likely with small data sets
- Invalidates Wald-type inference, but not score or likelihood ratio tests
- Can still get p-values and one-sided confidence limits
- Alternative methods to deal with this problem include exact logistic regression and the Firth correction