Data Separation Notes

Daniel J. Eck

Contents

Agresti example **2**
Separating hyperplane . . . 2
Optimization strangeness . . . 3
Canonical statistic on the boundary of its support . . . 5
Asymptope of log likelihood . . . 5
Fisher information degeneracy . . . 6

Inference in the complete separation setting **6**
Some theoretical details . . . 7
Preliminaries . . . 7
Completion of exponential family . . . 7
Likelihood maximizing sequences . . . 7
MLE exists in completion and Fisher Information degeneracy . . . 8
One-sided confidence intervals . . . 9
Logistic and binomial regression specifics . . . 9
Poisson regression specifics . . . 10
Return to Agresti example . . . 10
Software to perform inference . . . 10
Commentary on Agresti . . . 12

Not completely degenerate **12**
Other approaches: the problem with priors . . . 16
So which prior do we use? . . . 18
Using AIC . . . 18

Acknowledgments **20**

A problem with discrete exponential family models is that some data configurations can put a wrench in the gears of parameter estimation and statistical inference. In particular, when the observed value of the canonical statistic is on the boundary of its support than the exponential family is not identifiable. In logistic regression, this problem is known as complete or quasi-complete separation and it is informally described as the setting in which there exists a hyperplane which perfectly separates the successes from the failures. More formally, we say that perfect data separation (complete separation) occurs when there exists a vector b such that

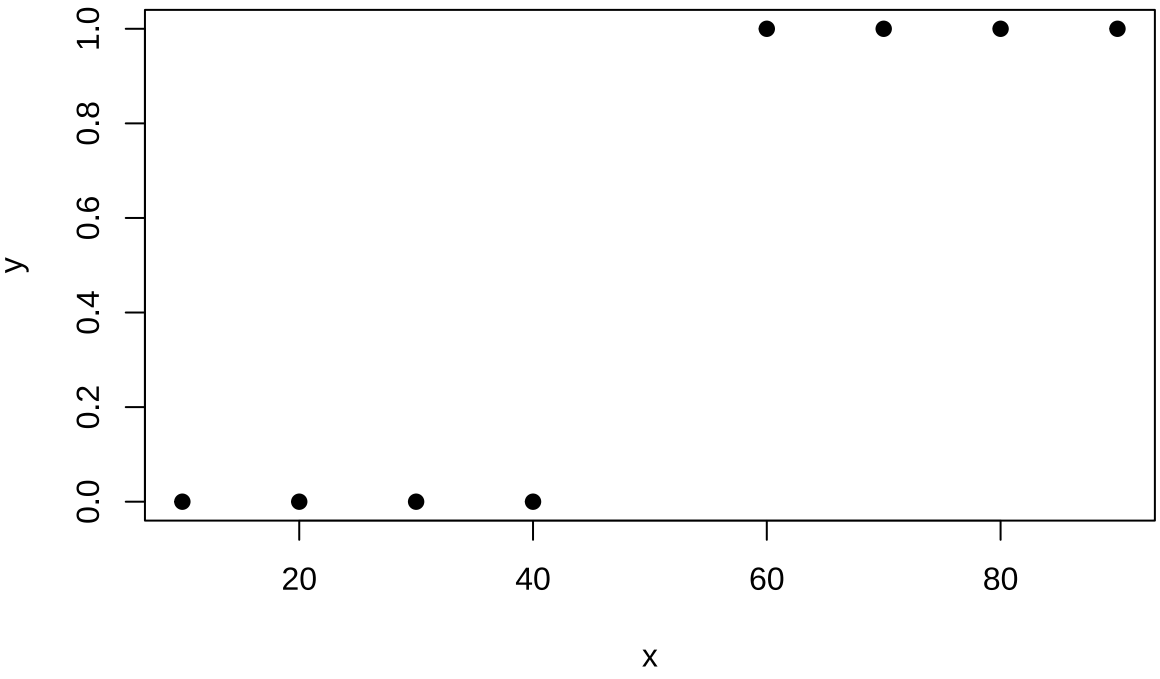
$$\begin{aligned} b^T x_i &> 0 \qquad \text{whenever } y_i = 1, \\ b^T x_i &< 0 \qquad \text{whenever } y_i = 0. \end{aligned}$$

See Section 6.5.1 in Agresti for more details.

Agresti example

Let's look at the example in Section 6.5.1 in Agresti to describe complete separation and how to deal with it under the exponential family/maximum likelihood estimation paradigm (which isn't mentioned in Agresti). In this example, the separation is immediately evident

```
x = (1:9 * 10)[-5]
y = c(0,0,0,0,1,1,1,1)
plot(x, y, pch = 19)
```



Separating hyperplane

If we consider a simple logisitc regression model with an intercept and predictor term, then the vector $b = (-50, 1)$ satisfies the above conditions for complete separation

```
## separation vector
b = c(-50, 1)

## model matrix
M = cbind(1, x)

## check condition
cbind(M %*% b, y)
```

```
##          y
## [1,] -40 0
## [2,] -30 0
## [3,] -20 0
## [4,] -10 0
## [5,]  10 1
## [6,]  20 1
## [7,]  30 1
## [8,]  40 1
```

The data exhibits complete degeneracy, and statistical inference is essentially meaningless. Luckily the computational checks have warned the user that a potential problem has occurred. Note that these warning messages do not describe what the problem is or provide any guidance for how users should handle this problem. Moreover, these checks are purely computational and do not necessarily imply that separation has occurred.

```
m1 = glm(y ~ x, family = "binomial")
```

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

```
## summary table
summary(m1)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial")
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -118.158 296046.187       0        1
## x                2.363   5805.939       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.1090e+01  on 7  degrees of freedom
## Residual deviance: 2.1827e-10  on 6  degrees of freedom
## AIC: 4
##
## Number of Fisher Scoring iterations: 25
```

```
## LRT
anova(m1, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     7      11.09
## x     1    11.09         6       0.00 0.0008678 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Optimization strangeness

Notice in the above that the number of Fisher scoring iterations is 25, which is the maximum allowable iterations under the default settings of the glm function. Look at what happens when change the defaults.

```r
m2 = glm(y ~ x, family = "binomial",
         control = list(maxit = 1e4, epsilon = 1e-100))
```

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

```r
summary(m2)
```

```
## 
## Call:
## glm(formula = y ~ x, family = "binomial", control = list(maxit = 10000, 
##     epsilon = 1e-100))
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.546e+02  4.939e+07       0        1
## x            3.092e+00  8.664e+05       0        1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1.1090e+01  on 7  degrees of freedom
## Residual deviance: 3.5527e-15  on 6  degrees of freedom
## AIC: 4
## 
## Number of Fisher Scoring iterations: 33
```

Changing the default optimization settings changed the summary output. In particular, note that the submodel canonical parameter estimates and their corresponding standard errors are diverging when we allow for more iterations. Do not worry about this finding as a general phenomenon. We will not observe this phenomenon in well-behaved data configurations with properly conditioned model matrices, the Fisher scoring algorithm converges well before 25 iterations are observed.

Returning to the model fit with the default settings, we see large canonical parameter estimates and mean value parameter estimates that are at the boundary of the closure of their parameter space $(0 < p < 1)$. Informally, estimates are “at infinity.”

```r
## submodel canonical parameter estimates
betahat = m1$coefficients
betahat
```

```
## (Intercept)           x 
##  -118.15802     2.36316 
```

```r
## saturated model canonical parameter estimates
thetahat = as.numeric(M %*% betahat)
thetahat
```

```
## [1] -94.52642 -70.89481 -47.26321 -23.63160  23.63160  47.26321  70.89481
## [8]  94.52642
```

```r
## saturated model mean value parameter estimates
phat = predict(m1, type = "response")
phat
```

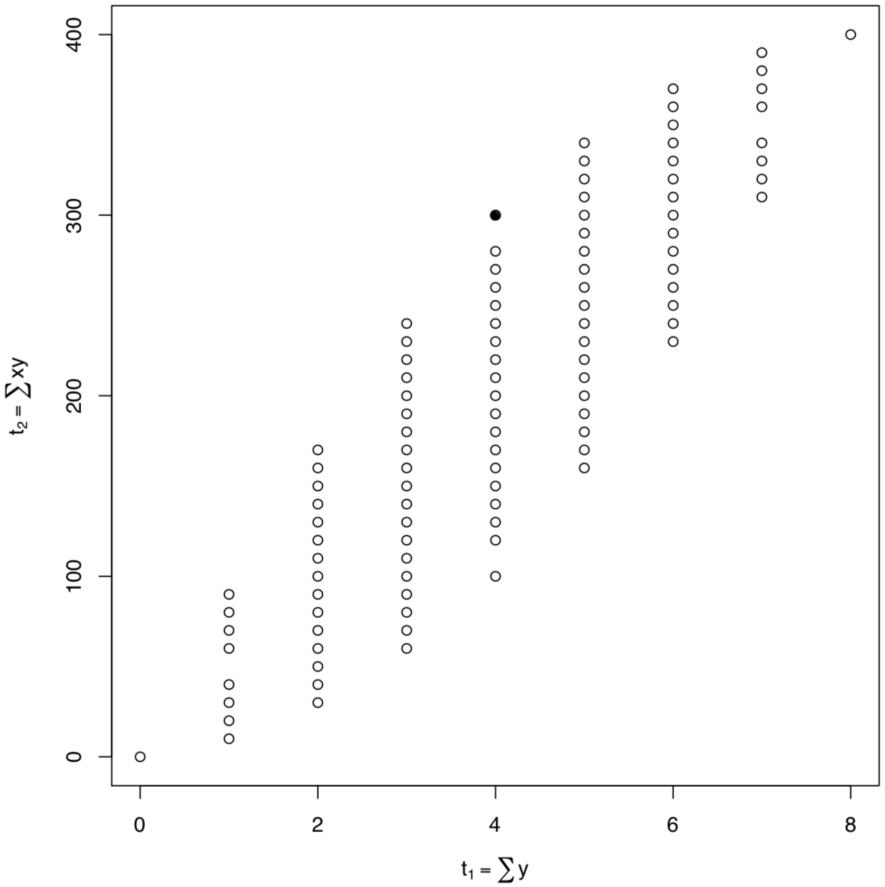
```
##            1            2            3            4            5            6 
## 2.220446e-16 2.220446e-16 2.220446e-16 5.456633e-11 1.000000e+00 1.000000e+00 
##            7            8 
## 1.000000e+00 1.000000e+00 
```

Canonical statistic on the boundary of its support

Recall that the submodel can be written as

$$\langle y, M\beta \rangle - c(M\beta) = \langle M^T y, \beta \rangle - c_\beta(\beta)$$

Also notice that the observed value of the canonical statistic $M^T y$ for the above submodel is on the boundary of the support of values for $M^T Y$. We will revisit this issue below.

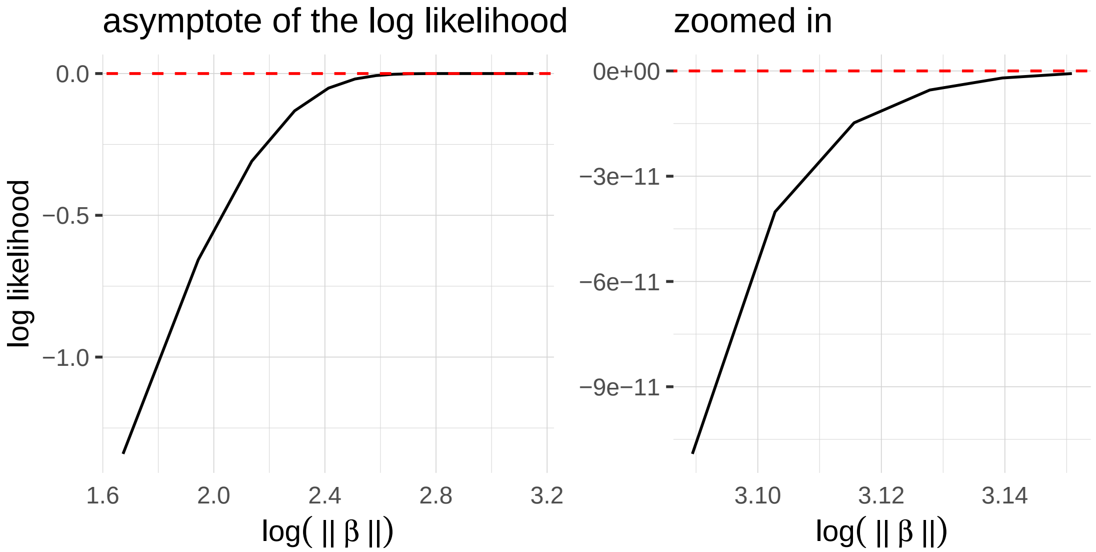


Note that, by the observed equals expected property, $M^T y$ is the MLE of the submodel mean-value parameter vector. The variance of the submodel canonical statistic is the Fisher information matrix.

Asymptope of log likelihood

```
asymptote = t(sapply(1:30, function(iter){
  m1 = glm(y ~ x, family = "binomial", control = list(maxit = iter, epsilon = 1e-20))
  c(sqrt(log(crossprod(coef(m1)))), logLik(m1))
}))
asymptote = as.data.frame(asymptote)
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Fisher information degeneracy

We now use R scripts to compute the Fisher information matrix. An eigenvector decomposition reveals that the Fisher information matrix is numerically singular.

```
invFI = vcov(m1)
FI = solve(invFI)
eigen(FI)
```

```
## eigen() decomposition
## $values
## [1] 7.715653e-07 1.140566e-11
##
## $vectors
##             [,1]        [,2]
## [1,] 0.01922747 -0.99981514
## [2,] 0.99981514  0.01922747
```

This implies that $\widehat{\text{Var}}(M^TY) = 0$. Therefore, the MLE solution to this problem is that the observed data which are on the boundary are the only possible data that could have occurred. Of course, these are estimates and not actual parameters. The problem is how to make inferential statements about model parameters given this degeneracy.

Inference in the complete separation setting

Before we discuss such an inferential approach we will give an example as motivation. The details of this example will be left for homework. Suppose that you have a coin that when flipped has a probability $0 < p < 1$ of landing heads, and that we know nothing about p. Suppose that you flip the coin four times and all four flips resulted in heads. Derive the MLE of p and the MLE of $\text{Var}(Y) = p(1-p)$ under the standard Bernoulli model. Now, for some error tolerance $0 < \alpha < 1$, derive a valid one-sided confidence interval for p making use of the statement $\mathbb{P}\left(\sum_{i=1}^4 Y_i = 4\right)$.

Some theoretical details

Preliminaries

We will consider a full exponential family with log likelihood

$$l(\theta) = \langle y, \theta \rangle - c(\theta). \tag{1}$$

Let C be the convex support of the canonical statistic, where we define the convex support as the smallest convex set that contains all values of the canonical statistic with probability equal to one. Define the tangent cone of C at observed value of the canonical statistics y as

$$T_C(y) = \mathrm{cl}\{s(w - y) : w \in C \text{ and } s > 0\},$$

where cl denotes the closure of a set.

A similar but simpler version of Theorem 4 from Geyer [2009] states that:

Theorem 1. *For a full exponential family with log likelihood* (1)*, convex support C and observed value of the canonical statistic $y \in C$, the following are equivalent:*

a) The MLE exists.

b) The tangent cone $T_C(y)$ is a vector subspace.

Thus the MLE does not exist when the canonical statistic is on the boundary of its support.

Completion of exponential family

The MLE may exist in the completion of the exponential family when it does not exist (ie when the canonical statistic is on the boundary of its support). We now define the completion of the exponential family.

Definition 1. *Let θ_n, $n = 1, \ldots,$ be a sequence of canonical parameter vectors for an exponential family having log likelihood* (1)*. Let $h_n(y) = l(\theta_n)$, and suppose that $h_n(y) \to h(y)$ pointwise as $n \to \infty$ where limits $-\infty$ and $+\infty$ are allowed. The limiting functions h form the completion of the exponential family.*

In the above definition h_n is a sequence of affine functions,

$$h_n(y) = l(\theta_n) = \langle y, \theta_n \rangle - c(\theta_n)$$

and the limiting function h is a "generalized affine function." Generalized affine functions and their properties are beyond the scope of this course. Definitions and details of generalized affine functions are presented in Section 6.1 of Eck and Geyer (2021).

Likelihood maximizing sequences

Now suppose that θ_n be a likelihood maximizing sequence of canonical parameter vectors (think of a IRLS algorithm where n is the iteration number, and θ_n is the maximizer at iteration n), that is,

$$l(\theta_n) \to \sup_{\theta \in \Theta} l(\theta), \qquad \text{as } n \to \infty,$$

where $\sup_{\theta \in \Theta} l(\theta) < \infty$.

Define $h_n(y) = l(\theta_n)$ as in the definition about the completion of exponential families. The limiting density e^h corresponds to the MLE distribution in the completion. Theoretical properties of these objects are beyond the scope of this course. Further details can be found in Eck and Geyer (2021).

MLE exists in completion and Fisher Information degeneracy

Now suppose that the MLE does not exist in the traditional exist and it exists in the completion of the exponential family, and suppose that we know the convex support set of the MLE distribution in the completion. Call this support set C.

Since the observed value of the canonical statistic is contained in C with probability one, and the canonical statistic for a GLM is M^TY, then we have $C = M^Ty + V$ where Y is the response vector, and y its observed value, and V is a vector space Geyer (2009), Section 3.9.

Then the limiting exponential family model taken along the maximum likelihood sequence θ_n (referred to as a limiting conditional model, or LCM for short) in which the MLE in the completion is found is the original model (OM) conditioned on the event

$$M^T(Y - y) \in V, \qquad \text{almost surely.}$$

Suppose that we can characterize V as a subspace where a finite set of linear equalities are satisfied

$$V = \{w \in \mathbb{R}^p : \langle w, \eta_i \rangle = 0, i = 1, \ldots, j\}.$$

Then we have that the LCM is conditioned

$$\langle M^T(Y - y), \eta_i \rangle = 0, \qquad i = 1, \ldots, j.$$

Recall that Fisher information is the variance for M^TY. From this we see that the vectors $\eta_1, \ldots, \eta_j$ span the null space of the Fisher information matrix for the LCM. The null space of the Fisher information matrix for the LCM is well approximated by the Fisher information matrix for the OM at parameter values that are close to maximizing the likelihood.

We can write

$$\langle M^T(Y - y), \eta_i \rangle = \langle (Y - y), M\eta_i \rangle.$$

Thus, the LCM is the OM conditioned on the event

$$\langle Y, M\eta_i \rangle = \langle y, M\eta_i \rangle, \qquad \text{almost surely for } i \in 1, \ldots, j. \tag{2}$$

The event (2) fixes some components of the response vector at their observed values and leaves the rest entirely unconstrained. Those components that are entirely unconstrained are those for which the corresponding component of $M\eta_i$ is zero (or, taking account of the inexactness of computer arithmetic, nearly zero) for all $i = 1, \ldots, j$. Such data points are said to form the LCM. Classical methods can be used to analyze the sample comprised of unconstrained data points which form the LCM.

In our motivating Agresti example we see that there no such points where $M\eta_i = 0$.

```r
M %*% eigen(FI)$vec
```

```
##           [,1]       [,2]
## [1,] 10.01738 -0.8075404
## [2,] 20.01553 -0.6152657
## [3,] 30.01368 -0.4229910
## [4,] 40.01183 -0.2307163
## [5,] 60.00814  0.1538331
## [6,] 70.00629  0.3461078
## [7,] 80.00444  0.5383825
## [8,] 90.00259  0.7306572
```

Therefore all components of the response are fixed at their observed value in this example. This data is completely degenerate with response variables constrained to be their observed values. If inferences are to be made then one-sided confidence intervals are necessary.

One-sided confidence intervals

We now provide a method for calculating these one-sided confidence intervals. Let

- I denote the index set of the components of the response vector on which we condition the OM to get the LCM.
- Y_I and y_I denote these components considered as a random vector and as an observed value, respectively.
- $\hat{\beta}$ is the estimated MLE of β in the LCM.
- Γ_{lim} is the null space of the Fisher information matrix.

Then endpoints for a $100(1-\alpha)\%$ confidence interval for a scalar parameter $g(\beta)$ are

$$\min_{\substack{\gamma \in \Gamma_{\text{lim}} \\ \mathbb{P}_{\hat{\beta}+\gamma}(Y_I = y_I) \geq \alpha}} g(\hat{\beta}+\gamma) \qquad \text{and} \qquad \max_{\substack{\gamma \in \Gamma_{\text{lim}} \\ \mathbb{P}_{\hat{\beta}+\gamma}(Y_I = y_I) \geq \alpha}} g(\hat{\beta}+\gamma). \tag{3}$$

In our Agresti example, $\hat{\beta}$ is 0 and $\Gamma_{\text{lim}} = \mathbb{R}^2$. Note that at least one of (3) is at the end of the range of this parameter (otherwise we can use conventional two-sided intervals).

The $100(1-\alpha)\%$ one-sided confidence intervals (3) follow from the same intuition of the "4 heads in 4 coin tosses" motivating example.

Logistic and binomial regression specifics

For logistic and binomial regression, let $p = \text{logit}^{-1}(\theta)$ denote the mean value parameter vector (here logit^{-1} operates componentwise). Then,

$$\mathbb{P}_{\beta}(Y_I = y_I) = \prod_{i \in I} p_i^{y_i}(1-p_i)^{n_i - y_i}$$

where the n_i are the binomial sample sizes. In logistic regression we have $n_i = 1$ for all i, but in binomial regression we have $n_i \geq 1$ for all i. We could take the confidence interval problem to be

$$\begin{aligned} &\text{maximize} \quad p_k, \qquad \text{subject to} \quad \prod_{i \in I} p_i^{y_i}(1-p_i)^{n_i - y_i} \geq \alpha; \\ &\text{minimize} \quad p_k, \qquad \text{subject to} \quad \prod_{i \in I} p_i^{y_i}(1-p_i)^{n_i - y_i} \geq \alpha \end{aligned} \tag{4}$$

where p is taken to be the function of γ described above. And this can be done for any $k \in I$. However, the problem will be more computationally stable if we state it as

$$\begin{aligned} \text{maximize} \quad & \theta_k \\ \text{subject to} \quad & \sum_{i \in I} \big[y_i \log(p_i) + (n_i - y_i)\log(1-p_i) \big] \geq \log(\alpha); \\ \text{minimize} \quad & \theta_k \\ \text{subject to} \quad & \sum_{i \in I} \big[y_i \log(p_i) + (n_i - y_i)\log(1-p_i) \big] \geq \log(\alpha), \end{aligned} \tag{5}$$

where $\theta_i = \text{logit}(p_i)$ is a monotone transformation and log is a monotone transformation. The two problems (4) and (5) are equivalent.

We maximize canonical rather than mean value parameters to avoid extreme inexactness of computer arithmetic in calculating mean value parameters near zero and one. We take logs in the constraint for the same reasons we take logs of likelihoods.

Poisson regression specifics

For Poisson sampling, let $\mu = \exp(\theta)$ denote the mean value parameter (here exp operates componentwise like the R function of the same name does), then

$$\mathbb{P}_\beta(Y_I = y_I) = \exp\left(-\sum_{i \in I} \mu_i\right).$$

We take the confidence interval problem to be

$$\text{maximize} \quad \mu_k, \qquad \text{subject to} \quad -\sum_{i \in I} \mu_i \geq \log(\alpha) \tag{6}$$

where μ is taken to be the function of γ described in (3). The optimization in (6) can be done for any $k \in I$.

Return to Agresti example

The exponential family in this Agresti example is completely degenerate. The MLE does not exist in the traditional sense, but does exist in the completion of the exponential family (maximized log likelihood is bounded). Conventional maximum likelihood computations come close, in a sense, to finding the MLE in the completion of the exponential family. They go uphill on the likelihood function until they meet their convergence criteria and stop.

At this point, canonical parameter estimates $\hat{\theta}$ and $\hat{\beta}$ are still infinitely far away from the MLE in the completion, but mean value parameter estimates $\hat{\mu}$ are close to the MLE in the completion, and the corresponding probability distributions are close in total variation norm to the MLE probability distribution in the completion.

Software to perform inference

The `inference` function in the R package `glmdr` (see the `glmdr` directory in the stat528resources repo; you will have to install the package locally) determines one-sided confidence intervals for mean value parameters corresponding to response values y_I for logistic and binomial regression as in (5) and Poisson regression as in (6).

```r
library(glmdr)
```

We return to the motivating Agresti example. Here we see that the Fisher information matrix has only null eigenvectors.

```r
eigen(FI)
```

```
## eigen() decomposition
## $values
## [1] 7.715653e-07 1.140566e-11
##
## $vectors
##             [,1]        [,2]
## [1,] 0.01922747 -0.99981514
## [2,] 0.99981514  0.01922747
```

In this case the MLE of the saturated model mean value parameters agree with the observed data; they are on the boundary of the set of possible values, either zero or one. Thus the LCM is completely degenerate at the one point set containing only the observed value of the canonical statistic of this exponential family. One-sided confidence intervals for mean value parameters (success probability considered as a function of the predictor x) are now computed. We first the logistic regression model using the `glmdr` fitting function instead of `glm`.

```
m_glmdr = glmdr(y ~ x)

## summary information
summary(m_glmdr)
```

```
## MLE exists in Barndorff-Nielsen completion
## it is completely degenerate
## the MLE says the response actually observed is the only
## possible value that could ever be observed
```

```
## $overview
## NULL
##
## $type
## [1] "degenerate"
##
## $linearity
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##
## attr(,"class")
## [1] "summary.glmdr"
```

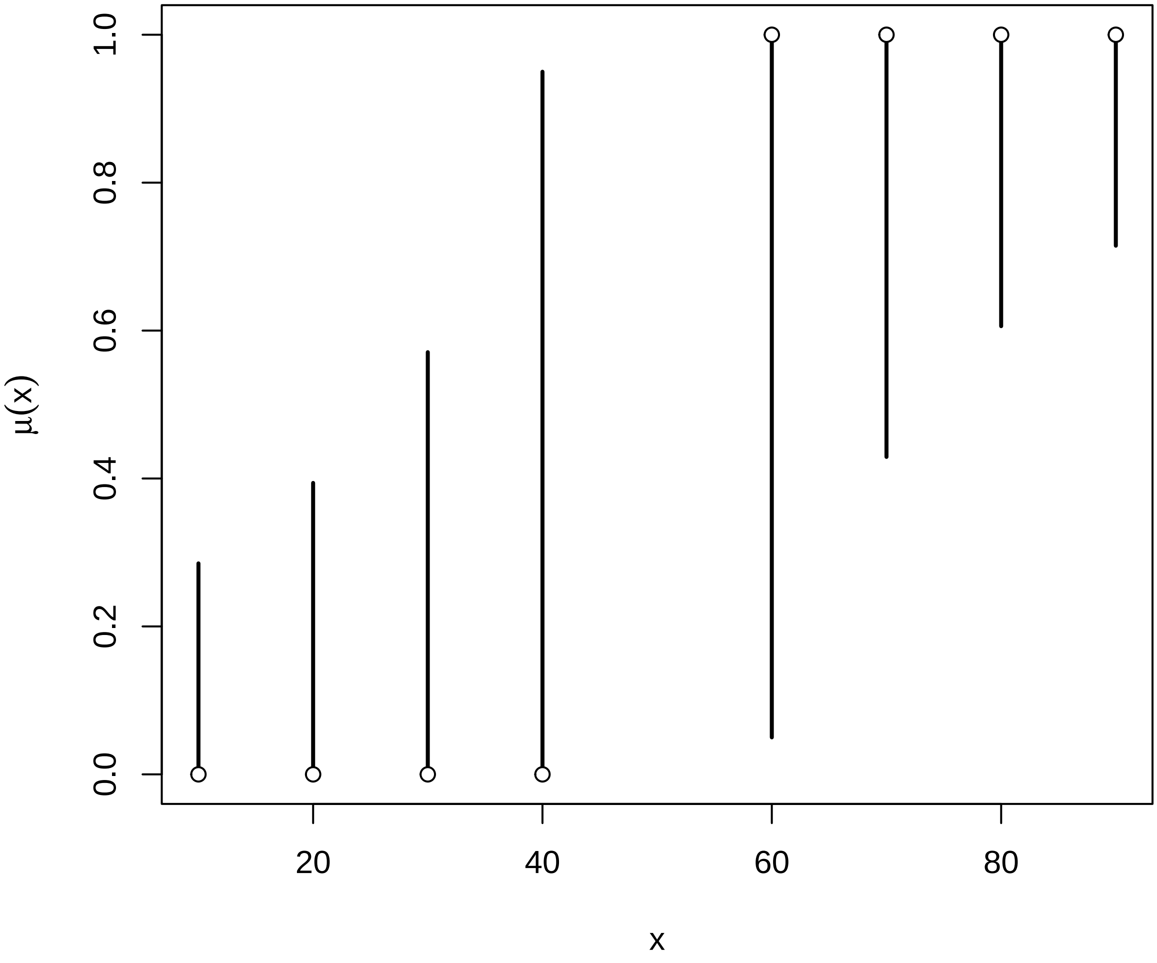
We then use the `inference` function to obtain one-sided confidence intervals for mean-value parameters corresponding to components Y_I that are constrained to be their observed values. This function performs the optimization in (3) where the function g in (3) is taken to be the map from submodel canonical parameter to saturated model mean-value parameters (ie the conditional success probabilities).

```
## one-sided CIs
CIs = inference(m_glmdr)
CIs
```

```
##       lower     upper
## 1 0.0000000 0.2852500
## 2 0.0000000 0.3940359
## 3 0.0000000 0.5708292
## 4 0.0000000 0.9500000
## 5 0.0500000 1.0000000
## 6 0.4291708 1.0000000
## 7 0.6059641 1.0000000
## 8 0.7147500 1.0000000
```

We now plot these one-sided confidence intervals.

```
bounds.lower.p = CIs$lower
bounds.upper.p = CIs$upper
par(mar = c(4, 4, 0, 0) + 0.1)
plot(x, y, axes = FALSE, type = "n",
    xlab = expression(x), ylab = expression(mu(x)))
segments(x, bounds.lower.p, x, bounds.upper.p, lwd = 2)
box()
axis(side = 1)
axis(side = 2)
points(x, y, pch = 21, bg = "white")
```



These confidence intervals are quite wide. This is due to the relatively lack of data.

Commentary on Agresti

The $n = 8$ data point example that we analyzed comes from Section 6.5.1 in Agresti. However, this textbook provides no model based inferential solution to this problem. In the above, we provided such a solution that exists within the exponential family modeling and maximum likelihood estimation paradigms. To be fair to Agresti, "solutions" to complete separation are discussed in Sections 7.4.7 and 7.4.8. This approach circumvents the model-based approach.

Not completely degenerate

In the Agresti example we noticed that the estimated Fisher information matrix was completely degenerate. This need not be so in generality, the Fisher information matrix can exhibit partial degeneracy. When this is so the LCM is not trivially degenerate like in the Agresti example. Data pairs (y_i, x_i) corresponding to response vectors which are left unconstrained form the LCM and parameter estimation can be conducted in a traditional manner. We will explore an example where with partial degeneracy.

We will consider the endometrial example in which the a histology grade and risk factors for 79 cases of endometrial cancer are analyzed.

```
library(enrichwith)
data(endometrial)
```

```
head(endometrial)
```

```
##   NV PI   EH HG
## 1  0 13 1.64  0
## 2  0 16 2.26  0
## 3  0  8 3.14  0
## 4  0 34 2.68  0
## 5  0 20 1.28  0
## 6  0  5 2.31  0
```

We begin with a standard logistic regression fit.

```
m = glm(HG ~ ., data = endometrial, family = "binomial",
         x = TRUE, y = TRUE)
summary(m)
```

```
##
## Call:
## glm(formula = HG ~ ., family = "binomial", data = endometrial,
##     x = TRUE, y = TRUE)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.30452    1.63730   2.629 0.008563 **
## NV           18.18556 1715.75089   0.011 0.991543
## PI           -0.04218    0.04433  -0.952 0.341333
## EH           -2.90261    0.84555  -3.433 0.000597 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 104.903  on 78  degrees of freedom
## Residual deviance:  55.393  on 75  degrees of freedom
## AIC: 63.393
##
## Number of Fisher Scoring iterations: 17
```

We observe **quasi-complete separation** in NV (a categorical variable with two levels), where we note that a 2 × 2 contingency table with an empty (zero) cell is an example of quasi-complete separation.

```
b = c(0,1,0,0)
library(data.table)
foo = setDT(as.data.frame(cbind(m$y, m$x %*% b)))
colnames(foo) = c("y", "NV")
foo[, .(.N), by = c("y", "NV")]
```

```
##        y    NV     N
##    <num> <num> <int>
## 1:     0     0    49
## 2:     1     0    17
## 3:     1     1    13
```

We now use `glmdr` to do our fitting.

```
m_glmdr = glmdr(HG ~ ., data = endometrial, family = "binomial")
summary(m_glmdr)
```

```
## MLE exists in Barndorff-Nielsen completion
## it is conditional on components of the response
## corresponding to object$linearity == FALSE being
## conditioned on their observed values
```

```
## $overview
## NULL
##
## $type
## [1] "lcm"
##
## $summary
##
## Call:
## stats::glm(formula = HG ~ ., family = "binomial", data = endometrial,
##     subset = c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
##     "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
##     "21", "27", "28", "29", "30", "31", "32", "33", "34", "35",
##     "36", "37", "38", "39", "40", "41", "42", "43", "44", "45",
##     "46", "47", "52", "53", "54", "55", "56", "57", "58", "59",
##     "60", "61", "62", "63", "64", "65", "66", "67", "68", "69",
##     "70", "72", "73", "74", "77", "79"), x = TRUE, y = TRUE)
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.30452    1.63720   2.629 0.008559 **
## NV                NA         NA      NA       NA
## PI          -0.04218    0.04433  -0.952 0.341310
## EH          -2.90261    0.84549  -3.433 0.000597 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 75.307  on 65  degrees of freedom
## Residual deviance: 55.393  on 63  degrees of freedom
## AIC: 61.393
##
## Number of Fisher Scoring iterations: 5
##
##
## $linearity
##     1     2     3     4     5     6     7     8     9    10    11    12    13
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##    14    15    16    17    18    19    20    21    22    23    24    25    26
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
##    27    28    29    30    31    32    33    34    35    36    37    38    39
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##    40    41    42    43    44    45    46    47    48    49    50    51    52
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE
##    53    54    55    56    57    58    59    60    61    62    63    64    65
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##    66    67    68    69    70    71    72    73    74    75    76    77    78
##  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE
##    79 
##  TRUE 
## 
## attr(,"class")
## [1] "summary.glmdr"
```

We now obtain inference for all mean-value parameters in two steps. We first use traditional methods to obtain inferences for mean-value parameters that are unconstrained. Then we can use the `inference` function to obtain one-sided confidence intervals for components of the response vector that are constrained at their observed values.

```
m2 = update(m, subset = m_glmdr$linearity)
summary(m2)
```

```
## 
## Call:
## glm(formula = HG ~ ., family = "binomial", data = endometrial, 
##     subset = m_glmdr$linearity, x = TRUE, y = TRUE)
## 
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  4.30452    1.63720   2.629 0.008559 ** 
## NV                NA         NA      NA       NA    
## PI          -0.04218    0.04433  -0.952 0.341310    
## EH          -2.90261    0.84549  -3.433 0.000597 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 75.307  on 65  degrees of freedom
## Residual deviance: 55.393  on 63  degrees of freedom
## AIC: 61.393
## 
## Number of Fisher Scoring iterations: 5
```

```
## get estimates of mean-value parameters in the LCM
preds = predict(m2, se.fit = TRUE, type = "response")
head(cbind(preds$fit, preds$se.fit))
```

```
##          [,1]        [,2]
## 1 0.268128294 0.074778103
## 2 0.050675634 0.034991612
## 3 0.005782436 0.008046224
## 4 0.007327976 0.010484202
## 5 0.436719783 0.095750088
## 6 0.068407171 0.053127413
```

```
## get one-sided CIs for constrained responses
preds_constrained = inference(m_glmdr)
cbind(endometrial[!m_glmdr$linearity, ], preds_constrained)
```

```
##    NV PI   EH HG     lower upper
## 22  1 38 0.97  1 0.7071451     1
## 23  1 22 1.14  1 0.7432730     1
## 24  1  7 0.88  1 0.9205964     1
## 25  1 25 0.91  1 0.8325905     1
## 26  1 15 0.58  1 0.9518366     1
## 48  1 22 1.44  1 0.5479196     1
## 49  1 40 1.18  1 0.5467740     1
## 50  1  5 0.93  1 0.9160397     1
## 51  1  0 1.17  1 0.8703443     1
## 71  1 49 0.27  1 0.9205150     1
## 75  1 11 1.01  1 0.8703894     1
## 76  1 21 0.98  1 0.8277338     1
## 78  1 19 1.02  1 0.8231611     1
```

We can test for the significance of the `NV` variable in the presence of quasi-complete separation using traditional means. Methods get harder when the degeneracy exists in the null model as explained in Section 3.15 of Geyer (2009).

```
m_small = glm(HG ~ PI + EH, data = endometrial, family = "binomial",
         x = TRUE, y = TRUE)
anova(m_small, m, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: HG ~ PI + EH
## Model 2: HG ~ NV + PI + EH
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        76     64.751
## 2        75     55.393  1   9.3576 0.002221 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(m); AIC(m_small)
```

```
## [1] 63.39326

## [1] 70.7509
```

Other approaches: the problem with priors

We now demonstrate inferential inconsistencies between the Bayesian methods, namely the inconsistencies with the weakly informative prior advocated here and implemented in the `bayesglm` package and Jeffrey's prior based approaches advocated for by Ioannis Kosmidis and David Firth in several papers and implemented in the `brglm2` package.

We first show that the `bayesglm` defaults produce p-values for the `NV` variable that are close to 0.05. Modest changes to these defaults can change decisions about this variable's significance when testing at the 0.05 level.

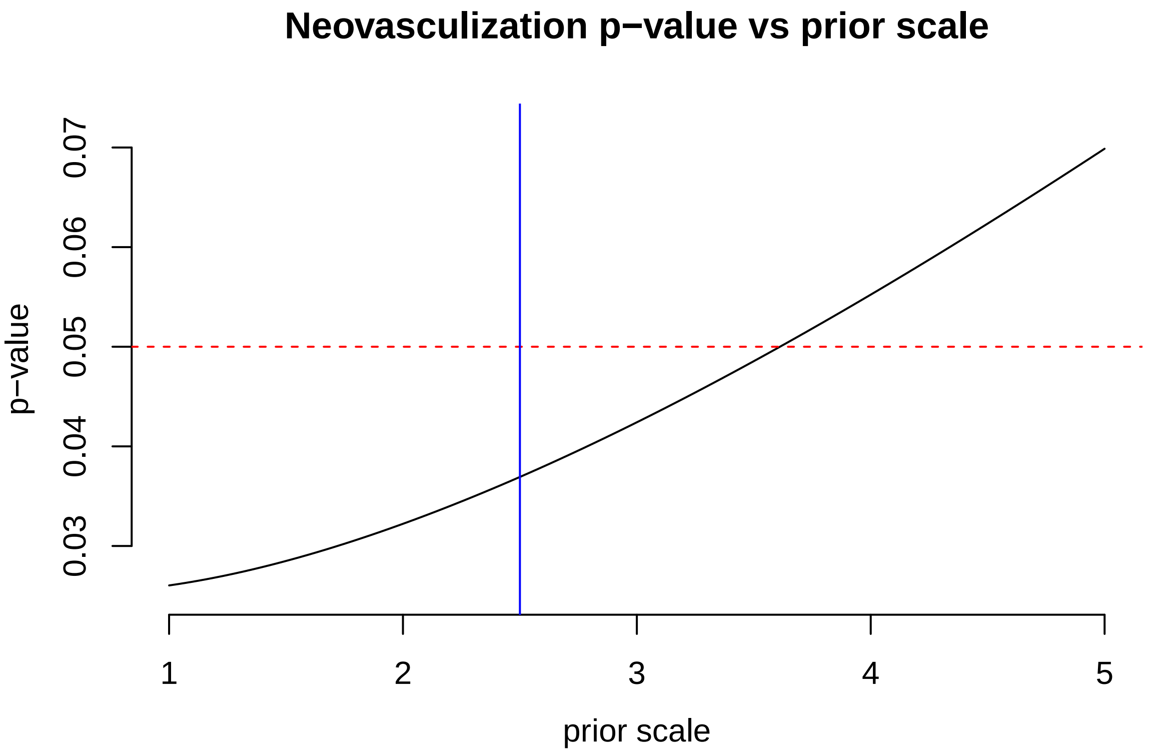
```
library(arm) # for bayesglm
library(brglm2) # for brglm2
#bayesglm
bayes_mod1 = bayesglm(HG~.,data=endometrial,family="binomial",
                       prior.scale = 1)
bayes_mod = bayesglm(HG~.,data=endometrial,family="binomial")
bayes_mod5 = bayesglm(HG~.,data=endometrial,family="binomial",
                       prior.scale = 5)
bayes_mod10 = bayesglm(HG~.,data=endometrial,family="binomial",
                       prior.scale = 10)

c(summary(bayes_mod1)$coef[2,4],
  summary(bayes_mod)$coef[2,4],
  summary(bayes_mod5)$coef[2,4],
  summary(bayes_mod10)$coef[2,4])
```

```
## [1] 0.02604166 0.03692929 0.06987578 0.15445148
```

```
xx = seq(from = 1, to = 5, length = 1e3)
foo = unlist(lapply(xx, function(j){
  summary(bayesglm(HG~.,data=endometrial,family="binomial",
                   prior.scale = j))$coef[2,4]
}))

plot.new()
plot.window(xlim = c(1,5), ylim = c(0.025, 0.0725))
title("Neovasculization p-value vs prior scale")
lines(xx, foo)
axis(1)
axis(2)
abline(h = 0.05, col = "red", lty = 2)
abline(v = 2.5, col = "blue", lty = 1)
mtext("prior scale", side = 1, line = 2.5)
mtext("p-value", side = 2, line = 2.5)
```



Different `brglm` fitting options yield different results, although these differences do not materialize in different conclusions for the `NV` variable when testing at the 0.05 significance level. However, these results conflict those produced by the `bayesglm` package.

```
#brglm2
brglm_mod = glm(HG~.,data=endometrial,family = "binomial",
                method = "brglm_fit", type = "MPL_Jeffreys")
brglm_mod_AS_mean = glm(HG~.,data=endometrial,family = "binomial",
                method = "brglm_fit", type = "AS_mean")
```

```
brglm_mod_AS_median = glm(HG~.,data=endometrial,family = "binomial",
                         method = "brglm_fit", type = "AS_median")
brglm_mod_AS_mixed = glm(HG~.,data=endometrial,family = "binomial",
                           method = "brglm_fit", type = "AS_mixed")
summary(brglm_mod)$coef[2,4]
```

```
## [1] 0.05890214
```

```
summary(brglm_mod_AS_mean)$coef[2,4]
```

```
## [1] 0.05890214
```

```
summary(brglm_mod_AS_median)$coef[2,4]
```

```
## [1] 0.09226857
```

```
summary(brglm_mod_AS_mixed)$coef[2,4]
```

```
## [1] 0.05890214
```

So which prior do we use?

- **Gelman**: Our prior [Cauchy $\mu = 0$; $\sigma = 2.5$] has the advantage of always giving answers, even when there is complete separation in logistic regression (a common problem, even when the sample size is large and the number of predictors is small), and also automatically applying more shrinkage to higher-order interactions.
- **Kosmidis and Firth**: Penalization of the likelihood by Jeffreys' invariant prior, or a positive power thereof, is shown to produce finite-valued maximum penalized likelihood estimates in a broad class of binomial generalized linear models. The class of models includes logistic regression, where the Jeffreys-prior penalty is known additionally to reduce the asymptotic bias of the maximum likelihood estimator
- **Gelman**: Our approach is similar [to Jeffrey's] but is parameterized in terms of the coefficients and thus allows us to make use of prior knowledge on that scale… We recommend this prior distribution as a default choice for routine applied use.
- **Kosmidis and Firth**: The apparent finiteness and shrinkage properties of the reduced-bias estimator, together with the fact that the estimator has the same first-order asymptotic distribution as the maximum likelihood estimator, are key reasons for the increasingly widespread use of Jeffreys-prior penalized logistic regression in applied work.
- **Gelman**: Scale-free prior distributions such as Jeffreys' do not include enough prior information.

What we have here is a failure to communicate.

Using AIC

AIC is slightly dubious as a model selection metric in this setting since models are not being evaluated at their MLE solutions. Nonetheless, we will investigate using AIC as a guide. Notice that, as expected, AIC is minimized when the standard logistic regression model is used. We also see that the `bayesglm` fit with the largest prior standard error among the candidate models minimizes AIC, and that the `brglm2` model with `type = AS_median` minimizes AIC among the competing `brglm2` models.

```
## bayesglm
AIC(bayes_mod1,
    bayes_mod,
    bayes_mod5,
    bayes_mod10)
```

```
##             df      AIC
## bayes_mod1   4 65.01979
## bayes_mod    4 64.25419
## bayes_mod5   4 63.78876
## bayes_mod10  4 63.54224
```

```
## brglm2
AIC(brglm_mod,
    brglm_mod_AS_mean,
    brglm_mod_AS_median,
    brglm_mod_AS_mixed)
```

```
##                      df      AIC
## brglm_mod             4 64.57540
## brglm_mod_AS_mean     4 64.57540
## brglm_mod_AS_median   4 63.86791
## brglm_mod_AS_mixed    4 64.57540
```

```
## MLE
AIC(m)
```

```
## [1] 63.39326
```

There is a conflict between AIC and p-values when investigating the importance of NV. This is further complicated by the `bayesglm` implementation because this procedure only considers the original scale of the covariates.

```
## bayesglm
bayes_mod10_small = bayesglm(HG ~ PI + EH,
                             data=endometrial, family="binomial",
                             prior.scale = 10)
### AIC
AIC(bayes_mod10_small, bayes_mod10)
```

```
##                    df      AIC
## bayes_mod10_small  3 70.75864
## bayes_mod10        4 63.54224
```

```
### p-value for NV
summary(bayes_mod10)$coef[2,4]
```

```
## [1] 0.1544515
```

```
## brglm2
brglm_mod_AS_median_small = glm(HG~ PI + EH,
  data=endometrial, family = "binomial",
  method = "brglm_fit", type = "AS_median")
### AIC
AIC(brglm_mod_AS_median_small, brglm_mod_AS_median)
```

```
##                            df      AIC
## brglm_mod_AS_median_small  3 70.81237
## brglm_mod_AS_median        4 63.86791
```

```
### p-value for NV
summary(brglm_mod_AS_median)$coef[2,4]
```

```
## [1] 0.09226857
```

Acknowledgments

These notes borrow materials from Charles Geyer's notes on exponential families. We also borrow materials from Agresti [2013], Geyer [2009], and Eck and Geyer [2021]. Special thanks to Suyoung Park for his work on the `glmdr` package.

References

Alan Agresti. *Categorical Data Analysis*. John Wiley & Sons, 2013.

Daniel J Eck and Charles J Geyer. Computationally efficient likelihood inference in exponential families when the maximum likelihood estimator does not exist. *Electronic Journal of Statistics*, 15(1):2105–2156, 2021.

Charles J Geyer. Likelihood inference in exponential families and directions of recession. *Electronic Journal of Statistics*, 3:259–289, 2009.