Stat 5421 Lecture Notes: Completion of Exponential Families

Charles J. Geyer

December 16, 2020

Contents

1 License

2 R

```
# library(alabama)
library(rcdd)
```

```
## If you want correct answers, use rational arithmetic.
## See the Warnings sections added to help pages for
##     functions that do computational geometry.
```

- The version of R used to make this document is 4.0.3.

- The version of the `alabama` package used to make this document is 2015.3.1.
- The version of the `numDeriv` package used to make this document is 2016.8.1.1.
- The version of the `rcdd` package used to make this document is 1.2.2.
- The version of the `rmarkdown` package used to make this document is 2.6.

3 Introduction

The purpose of this handout is to present two toy problems from Agresti (2013) which are two-dimensional exponential families (hence graphable) and illustrate situations where the maximum likelihood estimate (MLE) for the canonical parameter does not exist, although it does exist as a limit of distributions in the family. The MLE canonical parameter values can be thought of being "at infinity." The MLE mean value parameter values are on the relative boundary of convex support of the family (the smallest closed convex set containing the canonical statistic with probability one).

Agresti (2013) calls these models examples of "complete separation" and "quasi-complete separation." These terms do not generalize beyond logistic regression (or classification with two classes). The general terminology that applies to all exponential families that describes these toy models is

- MLE distribution concentrated at a single point (instead of complete separation) and
- MLE distribution not concentrated at a single point but concentrated on a proper subset of the support of the original model (instead of quasi-complete separation).

We use the methods of Geyer (2009) to analyze these toy models.

4 Example I

4.1 Data

Section 6.5.1 of Agresti (2013) introduces the notion of complete separation with the following example.

```
x <- seq(10, 90, 10)
x <- x[x != 50]
x
```

```
## [1] 10 20 30 40 60 70 80 90
```

```
y <- as.numeric(x > 50)
y
```

```
## [1] 0 0 0 0 1 1 1 1
```

The following figure shows these data.

4.2 Attempt to Fit Model

Suppose we want to do "simple" logistic regression (one predictor x plus intercept, so the model is two-dimensional). Let's try to do it naively.

```
gout <- glm(y ~ x, family = binomial)
```

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

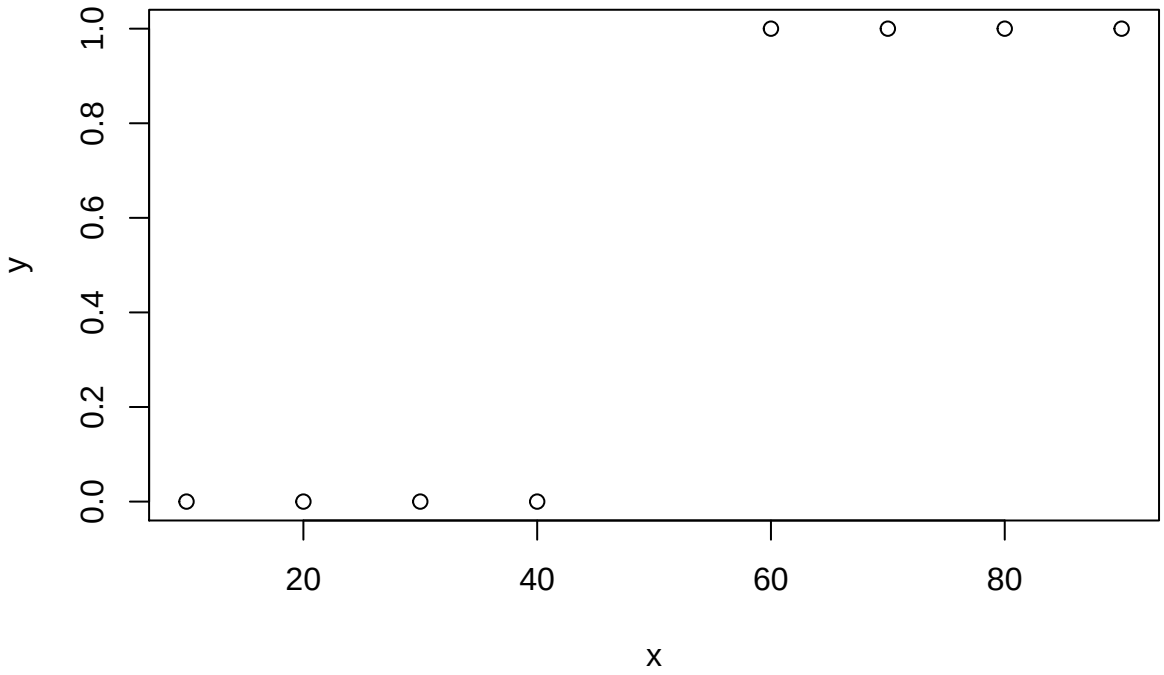

Figure 1: Logistic Regression Data for Example I

```
summary(gout)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.045e-05  -2.110e-08   0.000e+00   2.110e-08   1.045e-05
##
## 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
```

R function `glm` does give a warning. But what are you supposed to do about it? If the output of the R function `summary` is to be taken seriously, you cannot tell whether either regression coefficient is nonzero. As we shall see, that is complete nonsense. Both parameters have to go to infinity (one to plus infinity, the other to minus infinity) in order to maximize the likelihood.

In fact, these data are not analyzable by the R function `glm` and its associated functions (generic functions having methods for class `"glm"`). So we will use the theory of Barndorff-Nielsen completions of exponential

families from Geyer (2009).

4.3 Submodel Canonical Statistic

That theory tells us that we must look at the set of all possible values of the canonical statistic $M^T y$ where M is the model matrix and y is the response vector. For the model, M has two columns: the first column is all ones (the "intercept" column) and the second column is x. So let's find that set. There are 2^n possible values where n is the dimension of the response vector because each component of y can be either zero or one. The following code makes all of those vectors.

```
yy <- matrix(0:1, nrow = 2, ncol = length(x))
colnames(yy) <- paste0("y", x)
yy <- expand.grid(as.data.frame(yy))

head(yy)
```

```
##   y10 y20 y30 y40 y60 y70 y80 y90
## 1   0   0   0   0   0   0   0   0
## 2   1   0   0   0   0   0   0   0
## 3   0   1   0   0   0   0   0   0
## 4   1   1   0   0   0   0   0   0
## 5   0   0   1   0   0   0   0   0
## 6   1   0   1   0   0   0   0   0
```

```
dim(yy)
```

```
## [1] 256   8
```

But there are not so many distinct values of the submodel canonical statistic.

```
m <- cbind(1, x)
mtyy <- t(m) %*% t(yy)
t1 <- mtyy[1, ]
t2 <- mtyy[2, ]
t1.obs <- sum(y)
t2.obs <- sum(x * y)
```

Figure 2 shows these possible values of the submodel canonical statistic.

And now we are stuck. Figure 2 seems to show that the observed data vector is an extreme value, but we cannot easily figure out the direction of recession.

4.4 Computing the Limiting Conditional Model

Following Geyer (2009) we compute the support of the limiting conditional model (LCM) and a generic direction of recession (GDOR) as follows

```
tanv <- m
tanv[y == 1, ] <- (- tanv[y == 1, ])
vrep <- cbind(0, 0, tanv)
lout <- linearity(vrep, rep = "V")
p <- ncol(tanv)
hrep <- cbind(-vrep, -1)
hrep <- rbind(hrep, c(0, 1, rep(0, p), -1))
objv <- c(rep(0, p), 1)
```

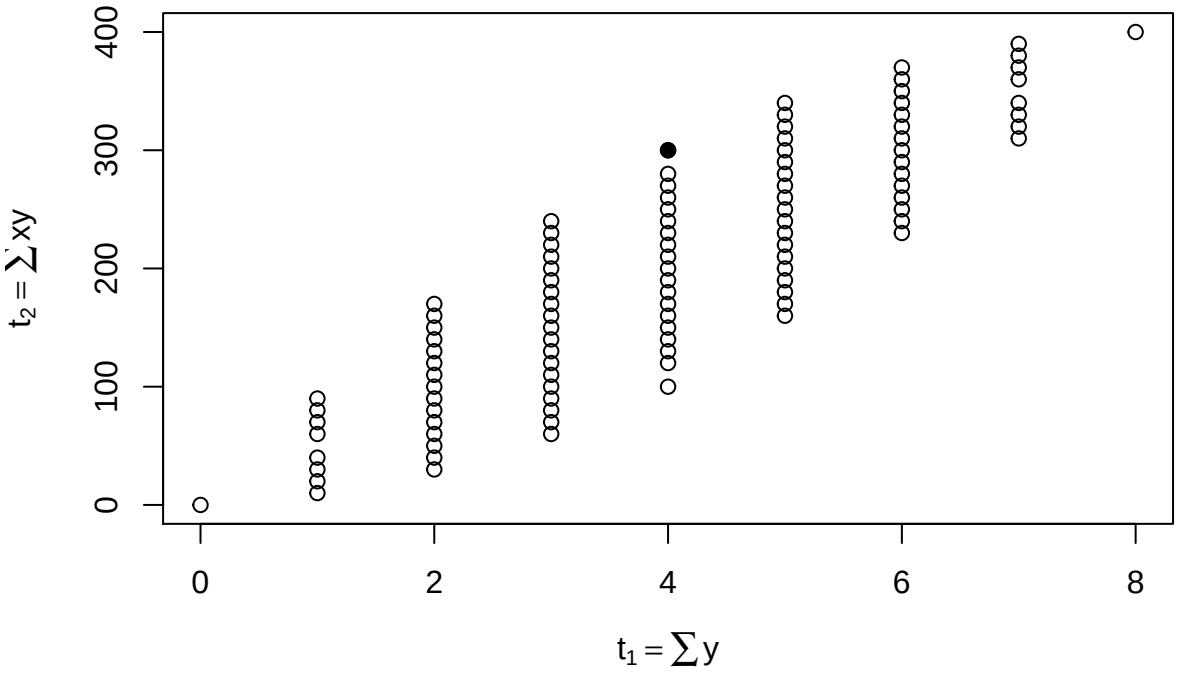

Figure 2: Possible values of the submodel canonical statistic vector $M^T y$ for the data shown in Figure 1. Solid dot is the observed value of the submodel canonical statistic vector.

```
pout <- lpcdd(hrep, objv, minimize = FALSE)
gdor <- pout$primal.solution[1:p]
```

This computes two vectors `lout` and `gdor`. The first is an index vector giving the components of the response vector that are random in the LCM (not conditioned on being equal to their observed values).

```
lout
```

```
## integer(0)
```

Here we have found that no components of the response vector are random in the LCM. The LCM is the completely degenerate distribution that says the response vector could not have had any value other than its observed value. The second is a GDOR.

```
gdor
```

```
## [1] -5.0  0.1
```

This shows the direction the parameters ("coefficients") go to infinity to obtain the LCM. The first coefficient (the intercept) goes to minus infinity 50 times faster than the second coefficient (the slope) goes to plus infinity.

This computation of the GDOR is not general. It only works in this complete separation case.

4.5 Confidence Regions

4.5.1 Theory

Geyer (2009) proposes a method of making confidence regions when the MLE for the canonical parameter does not exist (when the data are on the relative boundary of the convex support). Suppose δ is a GDOR, Y is the response vector, y its observed value, M is the model matrix, and $\eta = M\delta$. If we use $\langle Y, \eta \rangle$ as a

test statistic for an upper-tailed test, then we know $\langle Y, \eta \rangle \le \langle y, \eta \rangle$ almost surely by definition of direction of recession. We denote by H the hyperplane that is the support of the LCM, the event $\langle Y, \eta \rangle = \langle y, \eta \rangle$.

The P-value for the hypothesis test is $\Pr(Y \in H)$ since its observed value is its largest possible value. We can make a $100(1 - \alpha)\%$ confidence region by inverting the level α test. So the confidence region is the set of parameter values that put probability at least α on the support of the LCM.

4.5.2 For Submodel Canonical Parameters

Let θ denote the saturated model canonical parameter vector and β the submodel canonical parameter vector, which are related by $\theta = M\beta$.

Define a function f by

$$f(\beta) = \log \Pr_\beta(Y \in H)$$

We want to plot the curve which is the locus of points β such that $f(\beta) = \log \alpha$. Of course, we have

$$f(\beta) = \sum_{i=1}^{n} \bigl[y_i \log p_i + (1 - y_i) \log(1 - p_i) \bigr] \tag{1}$$

As in the notes on Agresti Chapter 4 we are careful computing these probabilities.

Hence we can write this probability calculation as an R function as follows.

```
alpha <- 0.05
fred <- function(beta) {
    theta <- m %*% beta
    logp <- ifelse(theta < 0, theta - log1p(exp(theta)),
        - log1p(exp(- theta)))
    logq <- ifelse(theta < 0, - log1p(exp(theta)),
        - theta - log1p(exp(-theta)))
    sum(logp[y == 1]) + sum(logq[y == 0]) - log(alpha)
}
```

Now we find one point inside the confidence region. We can do that by starting at any point and moving in the GDOR.

```
beta <- c(0, 0)
while (fred(beta) < 0) beta <- beta + gdor
beta
```

```
## [1] -5.0  0.1
```

```
fred(beta)
```

```
## [1] 1.981878
```

Then we find points on the curve in various directions by solving a univariate equation.

```
beta.start <- beta
theta <- seq(0, 2 * pi, length = 10001)
ss <- double(length(theta))
for (i in seq(along = theta)) {
    sally <- function(s) {
        beta <- beta.start + s * c(sin(theta[i]), cos(theta[i]))
        fred(beta)
    }
    uout <- try(uniroot(sally, c(0, 1), extendInt = "downX"), silent = TRUE)
```

```
    ss[i] <- if (inherits(uout, "try-error")) NaN else uout$root
}
beta1 <- beta.start[1] + ss * sin(theta)
beta2 <- beta.start[2] + ss * cos(theta)
```

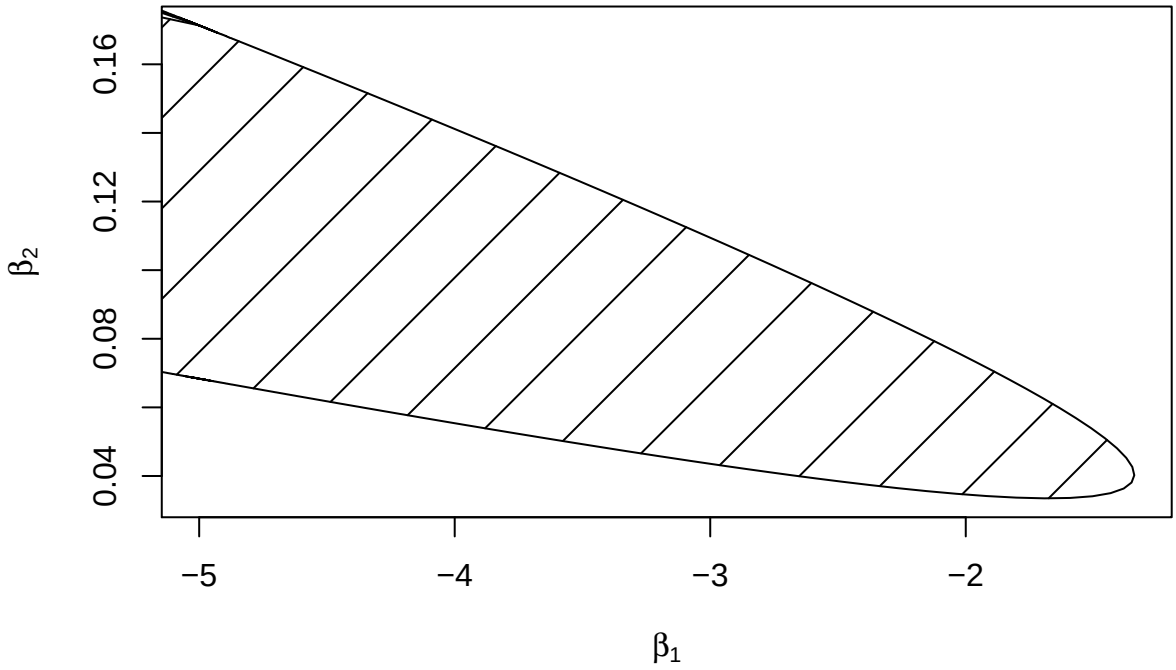

Figure 3: Confidence Region for Coefficients

4.5.3 For Submodel Mean Value Parameters

Let us map the confidence region shown in Figure 3 to the submodel mean value parameter scale. We know from exponential family theory that if Y is the response vector, and $\mu = E(Y)$ is the saturated model mean value parameter vector, and M is the model matrix, then the submodel mean value parameter vector is $\tau = M^T y$.

```
betas <- cbind(beta1, beta2)
thetas <- m %*% t(betas)
thetas <- t(thetas)
# as with betas, rows of thetas are parameter vectors
mus <- 1 / (1 + exp(- thetas))
taus <- t(m) %*% t(mus)
taus <- t(taus)
# as with betas, rows of taus are parameter vectors
# in this case submodel mean value parameter vectors
dim(taus)
```

```
## [1] 10001     2
```

And we plot this confidence region along with the support of the submodel canonical statistic vector (Figure 4).

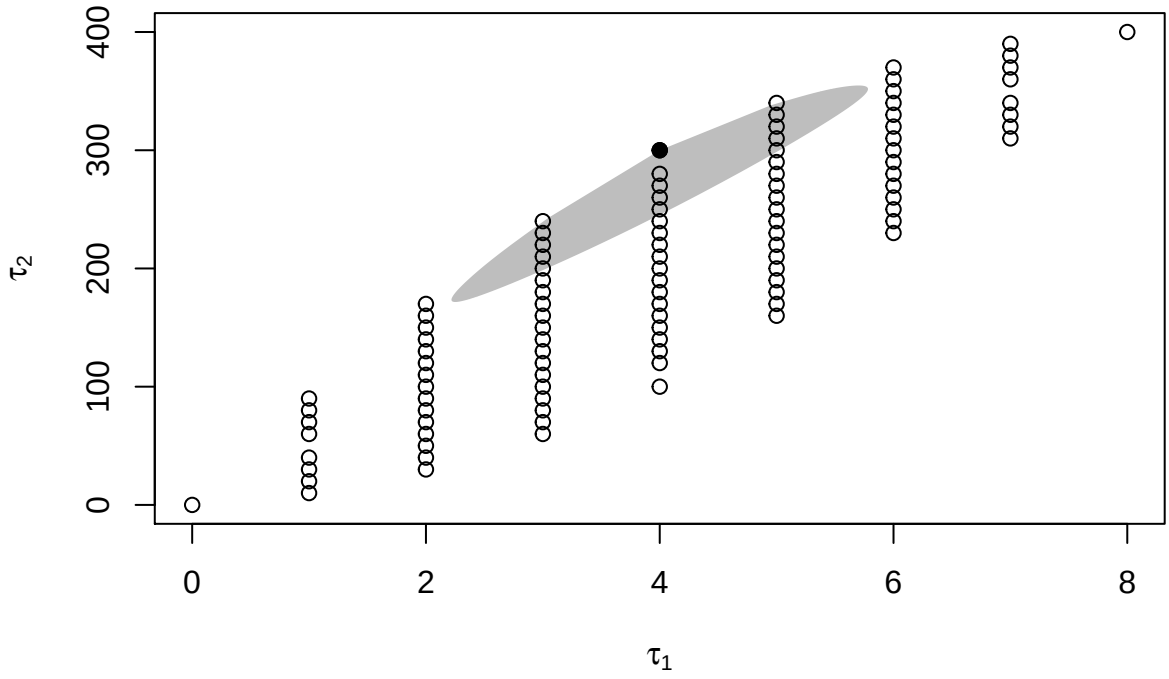

Figure 4: Confidence Region for Submodel Mean Value Parameter (gray). Dots as in Figure 2.

4.5.4 For Saturated Model Mean Value Parameters

After all of the above, the analogous confidence region for the saturated model mean value parameter vector $\mu = E(y)$, where y is the response vector, goes easier (the curve has already been calculated for x values in the observed data (the R object `mus`), but we would like to have predictions for all possible x values, not just those in the observed data). Thus we are also doing "prediction intervals" for unobserved predictor values x. These have the form

$$\mu(x) = \text{logit}^{-1}(\beta_1 + \beta_2 x), \tag{2}$$

where logit^{-1} denotes the inverse of the logit function.

In order for the calculations to not get out of hand, we do calculate on a grid of values. Although (2) makes sense for any real x, we only cover x values near the observed predictor values (interpolating not extrapolating).

```
x.new <- seq(-20, 120, length = 1401)
# treat x = 40 and x = 60 specially
x.new <- c(x.new, c(40, 60) + 1e-3, c(40, 60) - 1e3)
x.new <- sort(x.new)
m.new <- cbind(1, x.new)
thetas.new <- m.new %*% t(betas)
thetas.new <- t(thetas.new)
mus.new <- 1 / (1 + exp(- thetas.new))
dim(mus.new)
```

```
## [1] 10001  1405
```

Somewhere along these curves of μ vectors, maximum and minimum values are achieved. As with the τ vectors, we take an (assumed) limit by rounding and hoping it is correct, without doing the required real analysis.

```
mu.new.low <- apply(mus.new, 2, min, na.rm = TRUE)
mu.new.hig <- apply(mus.new, 2, max, na.rm = TRUE)
```

So here is the plot (Figure 5).

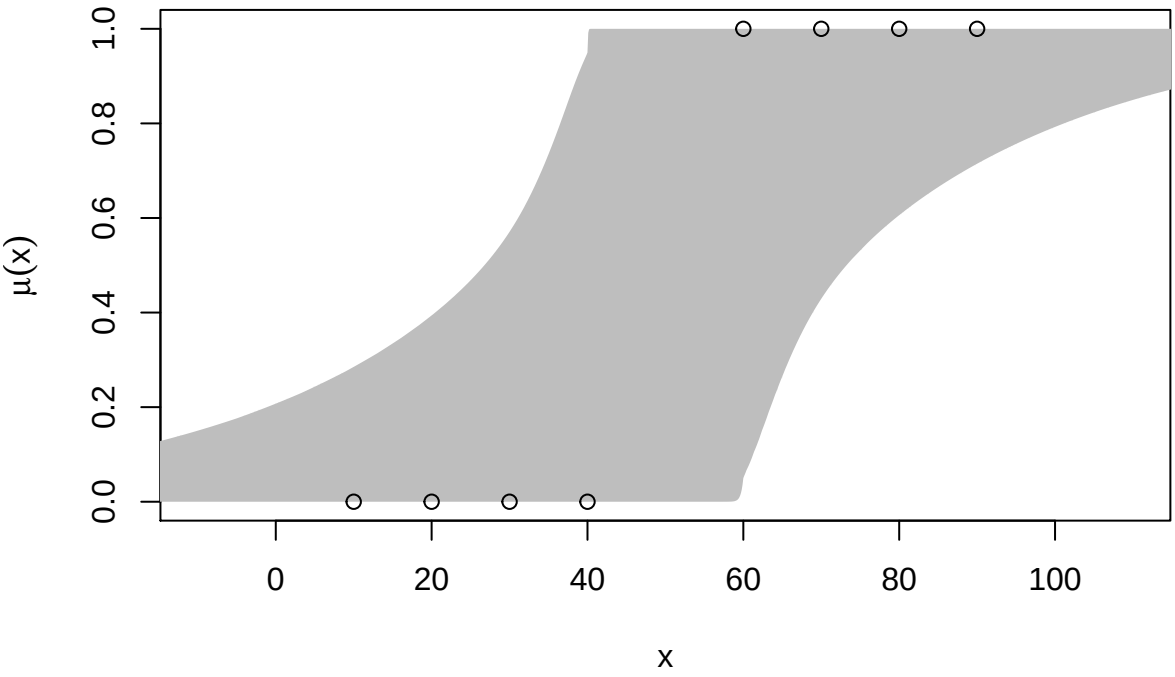

Figure 5: 95% Confidence Region (gray) for Saturated Model Mean Value Parameter (2) for Example I. Dots are as in Figure 1

This figure may seem more interpretable than Figure 3 or Figure 4, but it actually carries a lot less information. One-dimensional confidence intervals do not contain all of the information provided by the (two-dimensional) confidence region. Here, even though we have an infinite number of confidence intervals, one for $\mu(x)$ for each x, all of these intervals together do not tell us what either two-dimensional confidence region does (Figure 3 or Figure 4). Confidence intervals are just less informative than confidence regions (everywhere in statistics, not just in this context).

Because these confidence intervals come from a confidence region, they have simultaneous 95% coverage, even though there are an infinite number of them.

Another way to say what is wrong with Figure 5 is the following. We think the true unknown β is in the confidence region shown in Figure 3 (and, of course, are wrong about this 5% of the time). Any such β corresponds to a smooth curve $x \mapsto \mu(x)$ given by (2). So how Figure 5 should be interpreted is the collection of such smooth curves that fit inside the gray region are the confidence region (of curves).

5 Example II

5.1 Data

Agresti (2013) introduces the notion of quasi-complete separation with the following example, which adds two data points to the data for Example I.

```
x <- c(x, 50, 50)
y <- c(y, 0, 1)
```

Figure 6 shows these data.

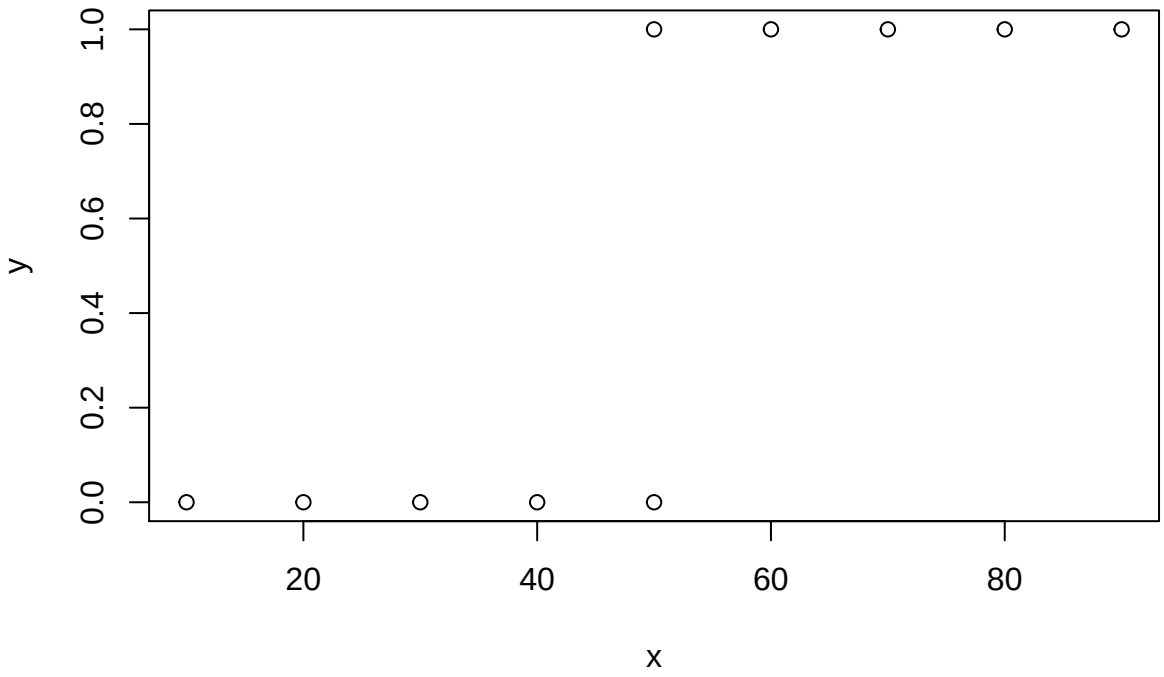

Figure 6: Logistic Regression Data for Example II

5.2 Attempt to Fit the Model

Again we want to do "simple" logistic regression (one predictor x plus intercept, so the model is two-dimensional). Again, if we try to do it naively, the R function `glm` complains.

```
gout <- glm(y ~ x, family = binomial)
```

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

```
summary(gout)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.177   0.000   0.000   0.000   1.177
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -98.158  39288.592  -0.002    0.998
## x               1.963    785.772   0.002    0.998
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 13.8629  on 9  degrees of freedom
## Residual deviance:  2.7726  on 8  degrees of freedom
## AIC: 6.7726
##
## Number of Fisher Scoring iterations: 21
```

But, again, the warning is useless because R does not give us any help in doing anything valid with these data.

5.3 Submodel Canonical Statistic

As in Section 4 we find the support of the submodel canonical statistic.

```
yy <- matrix(0:1, nrow = 2, ncol = length(x))
colnames(yy) <- paste0("y", x)
yy <- expand.grid(as.data.frame(yy))

head(yy)
```

```
##   y10 y20 y30 y40 y60 y70 y80 y90 y50 y50
## 1   0   0   0   0   0   0   0   0   0   0
## 2   1   0   0   0   0   0   0   0   0   0
## 3   0   1   0   0   0   0   0   0   0   0
## 4   1   1   0   0   0   0   0   0   0   0
## 5   0   0   1   0   0   0   0   0   0   0
## 6   1   0   1   0   0   0   0   0   0   0
```

```
dim(yy)
```

```
## [1] 1024   10
```

```
m <- cbind(1, x)
mtyy <- t(m) %*% t(yy)
t1 <- mtyy[1, ]
t2 <- mtyy[2, ]
t1.obs <- sum(y)
t2.obs <- sum(x * y)
```

Figure 7 shows these possible values of the submodel canonical statistic.

5.4 Computing the Limiting Conditional Model

Again following Geyer (2009) we compute the support of the limiting conditional model (LCM) and a generic direction of recession (GDOR) as follows

```
tanv <- m
tanv[y == 1, ] <- (- tanv[y == 1, ])
vrep <- cbind(0, 0, tanv)
lout <- linearity(vrep, rep = "V")
p <- ncol(tanv)
hrep <- cbind(0, 0, -tanv, -1)
hrep[lout, 1] <- 1
hrep[lout, p + 3] <- 0
hrep <- rbind(hrep, c(0, 1, rep(0, p), -1))
```

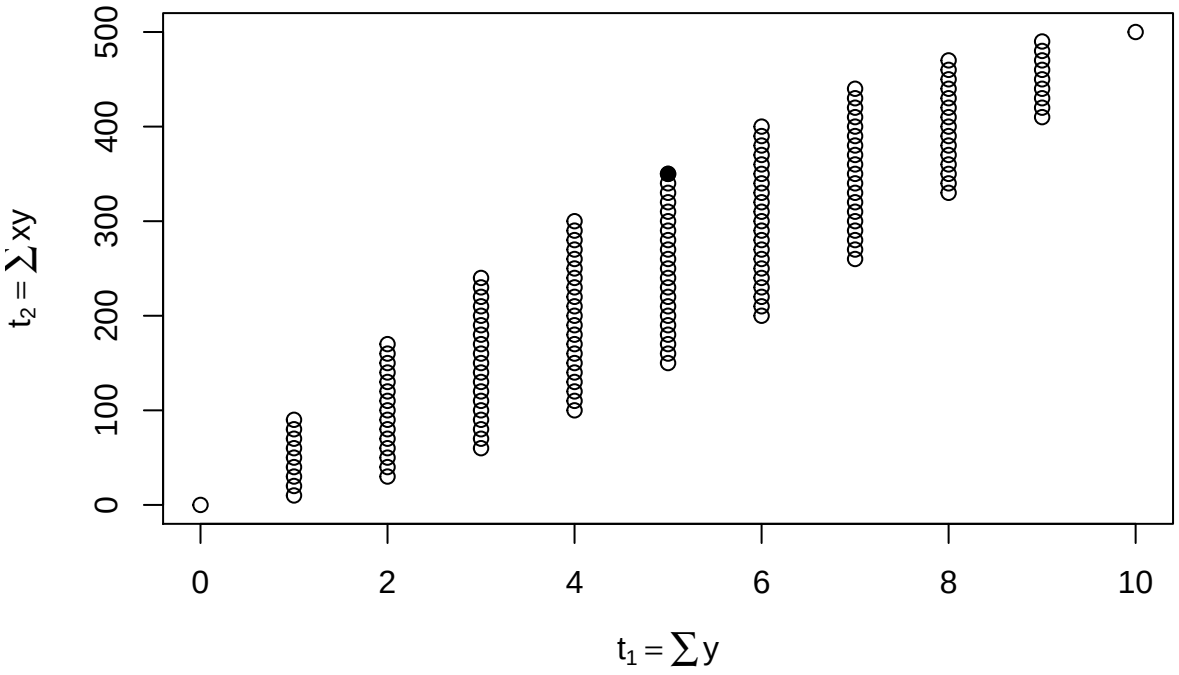

Figure 7: Possible values of the submodel canonical statistic vector $M^T y$ for the data shown in Figure 6. Solid dot is the observed value of the submodel canonical statistic vector.

```
objv <- c(rep(0, p), 1)
pout <- lpcdd(d2q(hrep), d2q(objv), minimize = FALSE)
gdor <- q2d(pout$primal.solution[1:p])
```

As before, this computes two vectors `lout` and `gdor`. The first is an index vector giving the components of the response vector that are random in the LCM (not conditioned on being equal to their observed values).

```
lout
```

```
## [1]  9 10
```

Here we have found that two components of the response vector (those for predictor $x = 50$) are random in the LCM. The LCM is a partially but not completely degenerate distribution. The second is a GDOR.

```
gdor
```

```
## [1] -5.0  0.1
```

This happens to be the same GDOR as before because the data that are conditioned on being equal to their observed values in the LCM are the same in both models.

5.5 Calculating the Maximum Likelihood Estimate

Having found the MLE (for the canonical parameter vector) does not exist, we find the MLE in the Barndorff-Nielsen completion by finding the MLE in the LCM. Following Section 3.14.2 in Geyer (2009), this is simple.

```
gout.lcm <- glm(y ~ x, family = binomial, subset = lout)
summary(gout.lcm)
```

```
##
```

```
## Call:
## glm(formula = y ~ x, family = binomial, subset = lout)
##
## Deviance Residuals:
##      9      10
## -1.177   1.177
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 4.710e-16  1.414e+00       0        1
## x                  NA         NA      NA       NA
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.7726  on 1  degrees of freedom
## Residual deviance: 2.7726  on 1  degrees of freedom
## AIC: 4.7726
##
## Number of Fisher Scoring iterations: 2
```

This says that for the two data points that are treated as random in the LCM (those for $x = 50$) the MLE has zero intercept and slope, which corresponds to success probabilty one-half, which is no surprise because there was one success and one failure for these two data points.

5.6 Confidence Regions

5.6.1 For Submodel Canonical Parameters

Now we have two kinds of confidence regions.

- One is our new kind of confidence region, the theory of which is expounded in Section 4.5.1 above.
- The other is the same old confidence interval calculated from the LCM.

Of course there are many recipies for old-style confidence intervals. We will use the Wald interval for the beta in the LCM.

```
foo <- summary(gout.lcm)$coefficients
old.ci.low <- foo[1, "Estimate"] - qnorm(1 - alpha / 2) * foo[1, "Std. Error"]
old.ci.hig <- foo[1, "Estimate"] + qnorm(1 - alpha / 2) * foo[1, "Std. Error"]
c(old.ci.low, old.ci.hig)
```

```
## [1] -2.771808  2.771808
```

Those are the endpoints of our "old" 95% confidence interval for β_1. Recall that β_2 is constrained to be zero in the LCM (it is not unknown). But we do have a direction of constancy of the LCM. It is the GDOR of the OM. (When only one parameter is "not defined because of singularities" as R function `summary` applied to an object produced by R function `glm` says, the GDOR is the only direction of constancy of the LCM. When there are more such parameters, there are more directions of constancy of the LCM.) Hence our confidence region in the original parameter space consists of all points of the form

$$(\beta_1, 0) + s(\eta_1, \eta_2),$$

where η is the GDOR, s is any real number, and β_1 is in our "old" confidence interval having endpoints calculated above.

Now that we have two confidence regions (the "new" one that says how close to to infinity we are in the OM, and the "old" one that restricts the parameters of the LCM), what do we do with them?

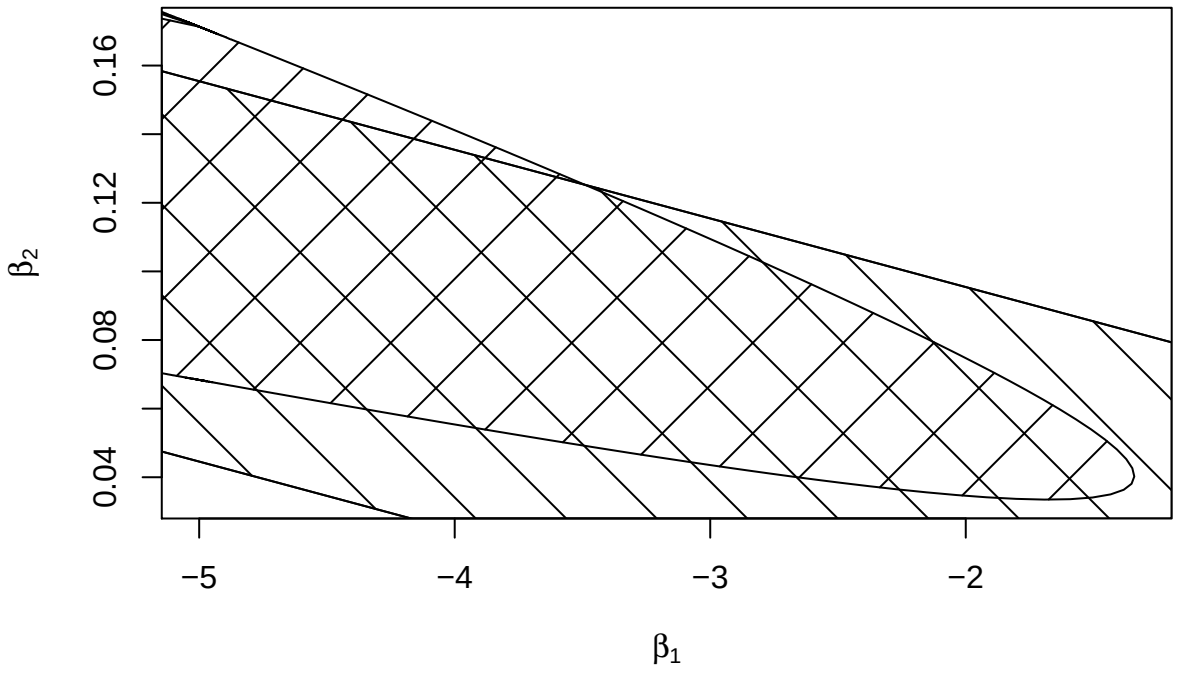

Figure 8: Confidence Regions for Coefficients for Example II. New region curved boundary. Old region straight boundaries.

Geyer (2009) suggests combining the two regions using Bonferroni correction. For example, the intersection of the two regions in the figure would be a 90% confidence interval.

In order to combine these two confidence regions we need to understand them a bit better. The upper boundary of the "old" region is the straight line

$$(\beta_1, \beta_2) = (u, 0) + s(\eta_1, \eta_2), \qquad s \in \mathbb{R},$$

where u is the upper bound of the Wald interval (2.7718076). Or

$$(\beta_1 - u, \beta_2) = s(\eta_1, \eta_2), \qquad s \in \mathbb{R},$$

or

$$\frac{\beta_1 - u}{\beta_2} = \frac{\eta_1}{\eta_2}$$

Since this is an upper bound, we have

$$\beta_2 \le \frac{\eta_2}{\eta_1}(\beta_1 - u)$$

and similarly for the lower bound, so

$$\frac{\eta_2}{\eta_1}(\beta_1 - l) \le \beta_2 \le \frac{\eta_2}{\eta_1}(\beta_1 - u)$$

where l is the lower bound (-2.7718076) So now we can calculate the combined region.

```
beta2 <- pmin(gdor[2] / gdor[1] * (beta1 - old.ci.hig), beta2)
beta2 <- pmax(gdor[2] / gdor[1] * (beta1 - old.ci.low), beta2)
```

5.6.2 For Submodel Mean Value Parameters

We skip this for these data because it won't be that much different than for the previous data.

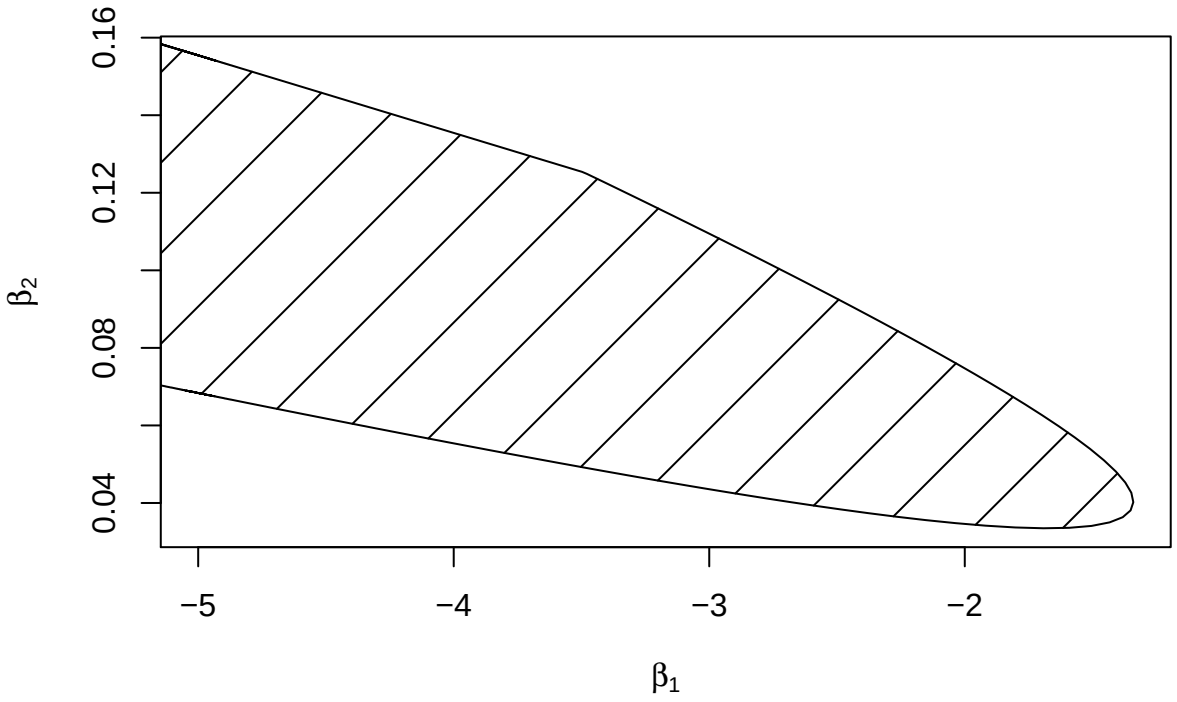

Figure 9: 90% Confidence Region for Coefficients for Example II.

5.6.3 For Saturated Model Mean Value Parameters

Now we follow Section 4.5.4 above except for using the confidence region shown in Figure 9 rather than the confidence region shown in Figure 3.

```
# treat x = 50 specially
x.new <- c(x.new, 50 + 1e-3, 50 - 1e3)
x.new <- sort(x.new)
m.new <- cbind(1, x.new)
betas <- cbind(beta1, beta2)
thetas.new <- m.new %*% t(betas)
thetas.new <- t(thetas.new)
mus.new <- 1 / (1 + exp(- thetas.new))
dim(mus.new)
```

```
## [1] 10001  1407
```

This gives all the mus that come from the confidence region, and we find the ranges as before.

```
mu.new.low <- apply(mus.new, 2, min, na.rm = TRUE)
mu.new.hig <- apply(mus.new, 2, max, na.rm = TRUE)
```

And then plot as before.

This is just one of many confidence regions we could have drawn. There is no reason why we should combine "new" and "old" confidence regions with equal coverage probabilities. If they have coverage probabilities $1 - \alpha_{\text{old}}$ and $1 - \alpha_{\text{new}}$, then Bonferroni says their intersection will have coverage probability $1 - \alpha_{\text{old}} - \alpha_{\text{new}}$. And we are free to choose α_{old} and α_{new} as we please.

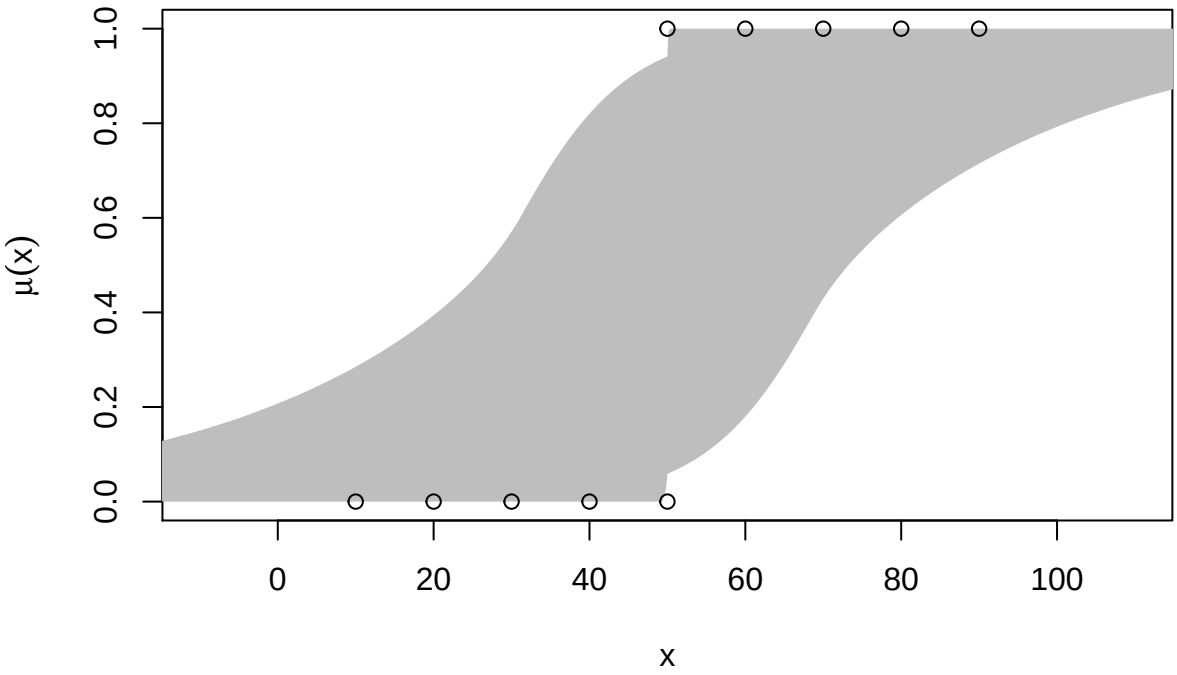

Figure 10: 90% Confidence Region (gray) for Saturated Model Mean Value Parameter (2) for Example II. Dots are as in Figure 6

Bibliography

Agresti, A. (2013) *Categorical Data Analysis*. Hoboken, NJ: John Wiley & Sons.

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