**Section 5.4: Examples**

Example: Placekicking

(Placekick-FindBestModel\_NotInBook.R, Examine.logistic.reg.R, placekick.mb.csv)

The data set used here has 13 more observations than the data set used for past placekick examples. The reason for why these observations are included (and then eventually excluded) will be given in the analysis.

The data used here also includes these additional variables:

* altitude: Official altitude in feet for the city where the placekick is attempted (not necessarily the exact altitude of the stadium)
* home: Binary variable denoting placekicks attempted at the placekicker's home (1) vs. away (0) stadium
* precip: Binary variable indicating whether precipitation is falling at game time (1) vs. no precipitation (0)
* temp72: Fahrenheit temperature at game time, where 72° is assigned to placekicks attempted within domed stadiums

Below is a summary of the steps taken to find the “best” logistic regression model for the placekicking data set.

Variable selection

I would like to use glmulti() to search among the main-effects and two-way interactions using the genetic algorithm. Because some interactions are not plausible for this problem, I would like to examine only the following interactions:

* distance with altitude, precip, wind, change, elap30, PAT, field, and temp72
* home with wind
* precip with type, field, and temp72

The exclude argument in glmulti() should allow for the exclusion of those interactions that I do not want to consider. Unfortunately, the function that does not allow this to work.

Instead, I used dredge() to search among the main-effects models. I used AIC as my information criterion. While BIC could be used, I chose AIC because I see this more as an exploratory study. I am o.k. if more explanatory variables are included in the model that what BIC would choose.

> placekick.mb <- read.csv("C:\\data\\Placekick.mb.csv")

> head(placekick.mb)

 week distance altitude home type precip wind change

1 4 18 585 1 0 0 0 0

2 11 18 585 1 0 0 0 1

3 11 18 10 0 1 0 0 1

4 1 19 20 0 1 0 0 0

5 3 19 5 0 0 0 0 1

6 6 19 20 0 1 0 0 0

 elap30 PAT field good temp72

1 24.3167 0 0 1 72

2 13.2000 0 0 1 72

3 1.5667 0 1 0 74

4 24.6333 0 1 1 79

5 25.6000 0 0 1 72

6 18.2500 0 1 1 82

> mod.fit.full <- glm(formula = good ~ week + distance + altitude + home + type + precip + wind + change + elap30 + PAT + field, family = binomial(link = logit), data = placekick.mb, na.action = na.fail)

> # summary(mod.fit.full)

> library(MuMIn)

> allsub.aic <- dredge(global.model = mod.fit.full, rank = "AIC")

Fixed term is "(Intercept)"

> allsub.aic[allsub.aic$delta < 1,]

Global model call: glm(formula = good ~ week + distance + altitude + home + type +

 precip + wind + change + elap30 + PAT + field, family = binomial(link = logit),

 data = placekick.mb, na.action = na.fail)

---

Model selection table

 (Intrc) chang dstnc home PAT week wind

1095 4.824 -0.3257 -0.08917 1.065 -0.5064

1607 5.067 -0.3384 -0.08920 1.062 -0.02507 -0.4592

583 5.028 -0.3445 -0.08852 1.076 -0.02796

71 4.751 -0.3310 -0.08842 1.080

1093 4.675 -0.08869 1.153 -0.5158

1127 4.728 -0.3256 -0.08929 0.1951 1.068 -0.5170

 df logLik AIC delta weight

1095 5 -385.382 780.8 0.00 0.210

1607 6 -384.540 781.1 0.32 0.179

583 5 -385.562 781.1 0.36 0.175

71 4 -386.629 781.3 0.49 0.164

1093 4 -386.793 781.6 0.82 0.139

1127 6 -384.831 781.7 0.90 0.134

Models ranked by AIC(x)

The main effects chosen are distance, wind, change, and PAT.

Next, I examine interactions using forward selection. I only include the plausible interactions among those explanatory variables remaining.

> start.mod <- glm(formula = good ~ distance + wind + change + PAT, family = binomial(link = logit), data = placekick.mb)

> full.mod <- glm(formula = good ~ distance + wind + change + PAT + distance:wind + distance:change + distance:PAT, family = binomial(link = logit), data = placekick.mb)

> forw.sel <- step(object = start.mod, scope = list(upper = full.mod), direction = "forward", k = 2, trace = TRUE)

Start: AIC=780.76

good ~ distance + wind + change + PAT

 Df Deviance AIC

+ distance:wind 1 765.20 777.20

+ distance:PAT 1 765.27 777.27

<none> 770.76 780.76

+ distance:change 1 770.60 782.60

Step: AIC=777.2

good ~ distance + wind + change + PAT + distance:wind

 Df Deviance AIC

+ distance:PAT 1 759.72 773.72

<none> 765.20 777.20

+ distance:change 1 765.09 779.09

Step: AIC=773.72

good ~ distance + wind + change + PAT + distance:wind + distance:PAT

 Df Deviance AIC

<none> 759.72 773.72

+ distance:change 1 759.61 775.61

> anova(forw.sel)

Analysis of Deviance Table

Model: binomial, link: logit

Response: good

Terms added sequentially (first to last)

 Df Deviance Resid. Df Resid. Dev

NULL 1437 1024.77

distance 1 237.707 1436 787.06

wind 1 3.008 1435 784.05

change 1 4.651 1434 779.40

PAT 1 8.639 1433 770.76

distance:wind 1 5.564 1432 765.20

distance:PAT 1 5.484 1431 759.72

The model now includes distance, wind, change, PAT, distance:wind, and distance:PAT.

This is a place where alternative approaches are justifiable:

1. Use all-subsets regression (keeping the four main-effects in the model) looking at the 23 = 8 possible models that include/exclude the interactions. The end result was the same as my approach.
2. Calculate the AIC for models that include some of the other interactions that are plausible. When this is done, one would need to include the corresponding main effect as well. For example, the model that includes the previously chosen main effects and interactions with altitude and distance:altitude results in an AIC of 777.32. This AIC is not smaller than what we had previously.

Investigate possible improvements to the model

First, we need to convert the data to EVP form.

> # Convert data to EVP form; interactions are not needed

 in aggregate() because they do not change the number

 of unique combinations of explanatory variables.

> w <- aggregate(x = good ~ distance + wind + change

 + PAT, data = placekick.mb, FUN = sum)

> n <- aggregate(x = good ~ distance + wind + change

 + PAT, data = placekick.mb, FUN = length)

> w.n <- data.frame(w, trials = n$good, prop =

 round(w$good/n$good, 4))

> head(w.n)

 distance wind change PAT good trials prop

1 18 0 0 0 1 1 1.00

2 19 0 0 0 3 3 1.00

3 20 0 0 0 15 15 1.00

4 21 0 0 0 11 12 0.92

5 22 0 0 0 7 8 0.88

6 23 0 0 0 15 15 1.00

> nrow(w.n) # Number of EVPs (M)

[1] 124

> sum(w.n$trials) # Number of observations

[1] 1438

> # Estimates here match those had before converting data

 to EVP form (not shown)

> mod.prelim1 <- glm(formula = good/trials ~ distance +

 wind + change + PAT + distance:wind + distance:PAT,

 family = binomial(link = logit), data = w.n, weights =

 trials)

> summary(mod.prelim1)

Call:

glm(formula = good/trials ~ distance + wind + change + PAT +

 distance:wind + distance:PAT, family = binomial(link = logit),

 data = w.n, weights = trials)

Deviance Residuals:

 Min 1Q Median 3Q Max

-2.2382 -0.5815 0.1895 0.8595 2.2814

Coefficients:

 Estimate Std. Error z value Pr(>|z|)

(Intercept) 4.49640 0.48142 9.340 < 2e-16 \*\*\*

distance -0.08069 0.01143 -7.062 1.64e-12 \*\*\*

wind 2.92477 1.78500 1.639 0.10131

change -0.33205 0.19454 -1.707 0.08785 .

PAT 6.71190 2.11372 3.175 0.00150 \*\*

distance:wind -0.09183 0.04570 -2.009 0.04449 \*

distance:PAT -0.27171 0.09800 -2.773 0.00556 \*\*

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

 Null deviance: 382.86 on 123 degrees of freedom

Residual deviance: 117.80 on 117 degrees of freedom

AIC: 268.36

Number of Fisher Scoring iterations: 5

Are there transformations of the explanatory variables that may be helpful? Obviously, there are no transformations of wind, change, and PAT that need to be investigated because there are binary explanatory variables. With respect to distance, we can plot the standardized Pearson residuals vs. distance to look for trends. If trends exist, this suggests a transformation may be needed.

> # Plot of standardized Pearson residuals vs. distance

> stand.resid <- rstandard(model = mod.prelim1, type =

 "pearson")

> plot(x = w.n$distance, y = stand.resid, ylim = c(min(-3,

 stand.resid), max(3, stand.resid)), ylab =

 "Standardized Pearson residuals", xlab = "Distance")

> abline(h = c(3, 2, 0, -2, -3), lty = 3, col = "blue")

> smooth.stand <- loess(formula = stand.resid ~ distance,

 data = w.n, weights = trials)

> ord.dist <- order(w.n$distance)

> lines(x = w.n$distance[ord.dist], y =

 predict(smooth.stand)[ord.dist], lty = "solid", col =

 "red")



There is no discernable trend, so I do not think a transformation is necessary. I also tried fitting a model that included a squared distance term, and obtained an AIC = 775.62. This value is larger than the AIC for the model without the squared term.

When you do this information criteria comparison, it is VERY important that you calculate both values either using the EVP form or the binary form of the data. Why?

Assessing the model fit – preliminary model

> one.fourth.root <- function(x) {

 x^0.25

 }

> save.info1 <- examine.logistic.reg(mod.fit.obj =

 mod.prelim1, identify.points = TRUE, scale.n =

 one.fourth.root, scale.cookd = sqrt)



Plots:

* (1,1): A few points are outside or close to these ±2 and ±3 lines
* (1,2): #120 has very large values in comparison to the others, so we definitely want to investigate it further; #117 has large leverage

* (2,1): Like in (1,1), we see a few points outside the lower threshold as we would expect; however, these points have relatively small plotting symbols (except perhaps #15) indicating that they may not really be unusual.

* (2,2): Notice that the very large Cook's distance value (#120, the largest circle on plot) has a relatively small squared standardized residual; this could indicate a very influential EVP that is “pulling” the model toward it to better its fit.

* Deviance/df: No evidence of any problems with the overall fit of the model

Next, I examined those EVPs identified in the plot further:

> w.n.diag1 <- data.frame(w.n, pi.hat =

 round(save.info1$pi.hat, 2), std.res =

 round(save.info1$stand.resid, 2), cookd =

 round(save.info1$cookd, 2), h = round(save.info1$h, 2), tail.prob = round(save.info1$tail.prob, 2))

> p <- length(mod.prelim1$coefficients)

> ck.out <- abs(w.n.diag1$std.res) > 2 | w.n.diag1$cookd >

 4/nrow(w.n) | w.n.diag1$h > 3\*p/nrow(w.n) # "|" means

 "or"

> extract.EVPs <- w.n.diag1[ck.out,]

> extract.EVPs[order(extract.EVPs$distance),] # Order by

 distance

 distance wind change PAT good trials prop pi.hat

60 18 0 1 0 1 2 0.50 0.94

117 20 0 0 1 605 614 0.99 0.98

121 20 1 0 1 42 42 1.00 0.99

123 20 0 1 1 94 97 0.97 0.98

101 25 1 1 0 1 2 0.50 0.94

119 29 0 0 1 0 1 0.00 0.73

120 30 0 0 1 3 4 0.75 0.65

103 31 1 1 0 0 1 0.00 0.85

15 32 0 0 0 12 18 0.67 0.87

48 32 1 0 0 0 1 0.00 0.87

87 45 0 1 0 1 5 0.20 0.63

55 50 1 0 0 1 1 1.00 0.23

 std.res cookd h tail.prob

60 -2.57 0.01 0.01 0.12

117 0.32 0.06 0.81 0.53

121 0.52 0.01 0.19 0.80

123 -0.75 0.02 0.23 0.34

101 -2.73 0.06 0.05 0.11

119 -1.76 0.07 0.13 0.27

120 0.83 0.31 0.76 0.57

103 -2.43 0.03 0.03 0.15

15 -2.67 0.06 0.05 0.02

48 -2.62 0.02 0.02 0.13

87 -2.03 0.02 0.03 0.07

55 1.90 0.04 0.07 0.23

Comments:

* Most of these EVPs have a very small number of trials, so standard normal and chi-square approximations are not going to work well. For example, I am not concerned about #60 having rm = -2.57 because there are only two trials associated with it. A standard normal approximation for it does not make sense.
* The large Cook’s distance for #117 is likely due to its relatively large numbers of trials. This EVP has 614 of the total 1438 trials in the data set! Of course, this should be influential.
* If one wanted to examine hm, the large values for #121 and #123 are likely due to the relatively large numbers of trials.
* #15 has a low number of successes in comparison to its estimated probability of success, and it has a moderate number of trials. This causes me to have a little concern about it, but notice the standardized Pearson residual is between ±3. Note that this is the only EVP with tail.prob value less than 0.05 (see program for code).
* #120 really concerns me. Its explanatory variables correspond to an unusual type of placekick too – a PAT which is not 20 yards! Interestingly, #119 is a PAT not at 20 yards too, and it has a larger Cook’s distance value. We definitely need to investigate these problems further.

To determine how influential these #119 and #120 truly are, I TEMPORARILY removed them from the data set and refit the model. I chose to do both at the same time because they are both non-20 yard PATs.

> mod.prelim1.wo119.120 <- glm(formula = good/trials ~ distance + wind + change + PAT + distance:wind + distance:PAT, family = binomial(link = logit), data = w.n[-c(119, 120),], weights = trials)

> summary(mod.prelim1.wo119.120)

Call:

glm(formula = good/trials ~ distance + wind + change + PAT + distance:wind + distance:PAT, family = binomial(link = logit), data = w.n[-c(119, 120), ], weights = trials)

Deviance Residuals:

 Min 1Q Median 3Q Max

-2.2388 -0.5791 0.1443 0.8419 2.2821

Coefficients:

 Estimate Std. Error z value Pr(>|z|)

(Intercept) 4.49852 0.48164 9.340 < 2e-16 \*\*\*

distance -0.08074 0.01143 -7.064 1.62e-12 \*\*\*

wind 2.87702 1.78664 1.610 0.1073

change -0.33078 0.19446 -1.701 0.0889 .

PAT -12.07030 49.21687 -0.245 0.8063

distance:wind -0.09072 0.04570 -1.985 0.0471 \*

distance:PAT 0.66661 2.46070 0.271 0.7865

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

 Null deviance: 377.95 on 121 degrees of freedom

Residual deviance: 113.93 on 115 degrees of freedom

AIC: 262.76

Number of Fisher Scoring iterations: 8

There is a dramatic change in the regression parameter estimate for distance:PAT! When both of the EVPs were in the data set, the estimate was -0.2717 with a Wald test p-value of 0.0056. It appears the presence of this interaction in the model was due to these EVPs consisting of just five placekicks.

Choices:

1. Leave these EVPs in the data set and keep the interaction in the model. In this case, I will just need to live with them being influential.
2. Take ALL non-20 yard PATs out of the data set and remove the interaction from the model. This will limit my population of inference.
3. Remove ALL PATs from the data set! Find the best model for just the field goals. Depending on interest, one could also find the best model for these PATs too in their own separate data set.

Overall, all three choices are justifiable. My preferences from best to worse are 2, 3, 1. Thus, I decided to go with 2.

With respect to 1, there is not enough information here (only 13 placekicks) to determine if the need for the interaction is a real trend or an anomaly of binary response data.

With respect to 3, one could do this if there was belief that the effects of the other variables differ based on whether or not the placekick was a PAT or field goal. I do not think this is the case. Plus, I think it is an important finding later in this example that the probability of success for 20-yard placekicks is different for field goals and PATs.

BIG NOTE: Removing observations from a data set should only occur when there is VERY GOOD justification for it! Just because an observation is not fit well by the model (i.e., an outlier) or is influential does NOT mean it should automatically be removed from the data set!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Assessing the model fit – revised model

First, the non-20 yard PATs need to be removed.

> # Remove non-20 yard PATs - "!" negates and "&" means

 "and"

> placekick.mb2 <- placekick.mb[!(placekick.mb$distance

 != 20 & placekick.mb$PAT == 1),]

> nrow(placekick.mb2) # Number of observations after

 13 were removed

[1] 1425

Now, the variable selection needs to be repeated due to the change in the data set! The code is in the corresponding program. As you might expect due to the small number of observations removed, the same terms for the model as before resulted (excluding the distance:PAT interaction). Below is information about the estimated model:

> # EVP form

> w2 <- aggregate(formula = good ~ distance + wind + change

 + PAT, data = placekick.mb2, FUN = sum)

> n2 <- aggregate(formula = good ~ distance + wind + change

 + PAT, data = placekick.mb2, FUN = length)

> w.n2 <- data.frame(w2, trials = n2$good, prop =

 round(w2$good/n2$good, 2))

> # head(w.n2)

> nrow(w.n2) # Number of EVPs

[1] 119

> sum(w.n2$trials) # Number of observations

[1] 1425

> mod.prelim2 <- glm(formula = good/trials ~ distance +

 wind + change + PAT + distance:wind, family =

 binomial(link = logit), data = w.n2, weights = trials)

> summary(mod.prelim2)

Call:

glm(formula = good/trials ~ distance + wind + change + PAT + distance:wind, family = binomial(link = logit), data = w.n2, weights = trials)

Deviance Residuals:

 Min 1Q Median 3Q Max

-2.2386 -0.5836 0.1965 0.8736 2.2822

Coefficients:

 Estimate Std. Error z value Pr(>|z|)

(Intercept) 4.49835 0.48163 9.340 < 2e-16 \*\*\*

distance -0.08074 0.01143 -7.064 1.62e-12 \*\*\*

wind 2.87783 1.78643 1.611 0.10719

change -0.33056 0.19445 -1.700 0.08914 .

PAT 1.25916 0.38714 3.252 0.00114 \*\*

distance:wind -0.09074 0.04570 -1.986 0.04706 \*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

 Null deviance: 376.01 on 118 degrees of freedom

Residual deviance: 113.86 on 113 degrees of freedom

AIC: 260.69

Number of Fisher Scoring iterations: 5

Again, examine.logistic.reg() was used to assess the fit of model, and the corresponding code is included in the program. Similar results as before were found here. Below is the corresponding plot:



Some EVPs that were “potentially” influential were temporarily removed from the data set one at a time to determine if actual changes would occur to the model. Overall, the changes were relatively small, so I am satisfied with the model! My final estimated model is



Interpreting the model

See p. 24 and 30 of Bilder and Loughin (1998) for how the model can be used to make inferences. Note that the analyses for the paper were done using SAS, so the estimates and plots may be slightly different than those obtained from R.

Because the paper uses Wald confidence intervals for the odds ratios, I will generally use profile LR intervals here. The two should not be too different due to the large sample size.

> library(package = mcprofile)

> OR.name <- c("Change", "PAT", "Distance, 10-yd

 decrease, windy", "Distance, 10-yd decrease, not

 windy", "Wind, distance = 20", "Wind, distance = 30",

 "Wind, distance = 40", "Wind, distance = 50", "Wind,

 distance = 60")

> var.name <- c("int", "distance", "wind", "change", "PAT",

 "distance:wind")

> K <- matrix(data = c(0, 0, 0, 1, 0, 0,

 0, 0, 0, 0, 1, 0,

 0, -10, 0, 0, 0, -10,

 0, -10, 0, 0, 0, 0,

 0, 0, 1, 0, 0, 20,

 0, 0, 1, 0, 0, 30,

 0, 0, 1, 0, 0, 40,

 0, 0, 1, 0, 0, 50,

 0, 0, 1, 0, 0, 60),

 nrow = 9, ncol = 6, byrow = TRUE, dimnames =

 list(OR.name, var.name))

> # K # Check matrix

> linear.combo <- mcprofile(object = mod.prelim2, CM = K)

> ci.log.OR <- confint(object = linear.combo, level = 0.90,

 adjust = "none")

> exp(ci.log.OR)

 mcprofile - Confidence Intervals

level: 0.9

adjustment: none

 Estimate lower upper

Change 0.7185 0.5223 0.991

PAT 3.5225 1.8857 6.785

Distance, 10-yd decrease, windy 5.5557 2.8871 12.977

Distance, 10-yd decrease, not windy 2.2421 1.8646 2.717

Wind, distance = 20 2.8950 0.7764 16.242

Wind, distance = 30 1.1683 0.5392 3.094

Wind, distance = 40 0.4715 0.2546 0.869

Wind, distance = 50 0.1903 0.0598 0.515

Wind, distance = 60 0.0768 0.0111 0.377

For the probability of success, below is an example of how to calculate profile LR intervals involving 20-yard placekicks under non-windy and no lead change conditions.

> K <- matrix(data = c(1, 20, 0, 0, 1, 0,

 1, 20, 0, 0, 0, 0), nrow = 2, ncol =

 6, byrow = TRUE, dimnames = list(c("PAT", "FG"),

 var.name))

> linear.combo <- mcprofile(object = mod.prelim2, CM = K)

> ci.lin.pred <- confint(object = linear.combo, level =

 0.90, adjust = "none")

> # as.matrix() is needed to obtain the proper class for

 plogis()

> round(plogis(q = as.matrix(ci.lin.pred$estimate)), digits

 = 3)

 Estimate

PAT 0.984

FG 0.947

> round(plogis(q = as.matrix(ci.lin.pred$confint)), digits

 = 3)

 lower upper

[1,] 0.976 0.991

[2,] 0.921 0.966

Lin Elliott’s placekick corresponding to the paper discussion:

> K <- matrix(data = c(1, 42, 0, 1, 0, 0), nrow = 1, ncol =

 6, byrow = TRUE)

> linear.combo <- mcprofile(object = mod.prelim2, CM = K)

> ci.lin.pred <- confint(object = linear.combo, level =

 0.90, adjust = "none")

> round(plogis(q = as.matrix(ci.lin.pred$estimate)), digits

 = 3) # as.numeric() and as.vector() do not work

 Estimate

C1 0.685

> round(plogis(q = as.matrix(ci.lin.pred$confint)), digits

 = 3)

 lower upper

[1,] 0.628 0.738

Below are a few plots for when PAT = 0 (see code in programs):



