Corn data

Current Problem: glm detects the separation (i.e. our estimator is at infinity) but our method, glmdr does not. Furthermore, we want to see if detect\_separation (from brglm2 package) can detect the separation in this case. I firstly mention few findings in corndata.R in part 1then discuss what I did in part 2.

Part 1: corndata.R

To my understanding, the problem setting is that we want to fit the model using glmdr then fit the model again based on the subset that excludes problematic points (i.e. linearity == FALSE). In this case, our model should be free from complete separation as we have already removed problematic points.

Given dataset (Combined\_Final\_Product) contains 1547 x 34 and we use the subset that its column starts from 11 to 34 (i.e. 24 columns. X matrix in the line 12 in the corndata.R). Then, we remove problematic columns (I am not sure this is what you want but to my understanding you want to remove NA columns in coefficients from glm). When we fit the model with this X matrix, coefficients for S6\_82186661 (15th), S6\_82218044 (19th), S6\_82243861 (22nd), S6\_82243874 (23rd) and S6\_82243883 (24th) are “NA” (Based on my knowledge, this happens due to the multicollinearity – maybe I assume this is reason why you checked eigenvalue of matrix?). However, I think there is indexing problem in Xind – line 14 (it should not show “NA”):

.

Also, in line 17 and 22, two models are NOT the same:

A picture containing graphical user interface

Description automatically generated.

That is, m1 does not have intercept term, meanwhile, m2 have the intercept term (but this may not be important to the next step).

After all, I think problem in this code comes from our model matrix, foo, which has the exact collinearity problem.

Part 2: corndata2.R

In this file, I removed the duplicated column and compare glm, glmdr and detect separation (as of now, author separate detect\_separation function from brglm2 and created the new package called, “detectseparation.” Thus, you may need to install this package from CRAN).

Firstly, I checked which columns have the same values (line 14-18 in corndata2.R) to confirm my interpretation (multicollinearity) in the glm function. It turns out 1) 14th and 15th 2) 18th and 19th 3) 21, 22, 23, and 24th columns have the same value. Thus, I removed 15, 19, 22, 23, 24th column from our model matrix and created the foo while column-wise binding (kernel.color and pop.structure).

Q: Without statistical or mathematical reason, is it okay to drop the column even though they have the same column considering we are handling genetic data?

To detect the complete separation, I firstly fit the model using 1) glm (named as \_glm) then 2) send them to the infinity by changing maxit and epsilon in glm function (named as \_inf). If any of estimators goes to infinity, we can see there is a complete separation problem. If the estimator are close or the same between 1) and 2), then there is no separation problem.

I used 3) detect\_separation function and 4) glmdr.

Interestingly, result fomr the detect\_separation says that all parameters expect the “Pop.structure” lie at infinity.

Table

Description automatically generated

Our glmdr shows that there is the complete separation and there are **74** problematic points (in corndata.R, we saw only 3 problematic points).

Therefore, combining results from 3) and 4), suppose we fit the model while EXCLUDING the 74 problematic points we identified in 4), we will be able to estimate the parameter for Pop.structure (other columns will be NA) ⬄ we should NOT see separation problem.

As we expected we can estimate parameters of pop structure (line 39 – 40, and there is no warning message (glm.fit: fitted probabilities numerically 0 or 1 occurred) as well, although it does not guarantee the exist of the infinite estimator). Notice that m2\_inf (line 42 - 43) still shows different results with m2\_glm even though there are no complete separation. This is because our data, foo[m1\_glmdr$linearity,] contain the all same value in every column except the pop structure (line 44).

A picture containing table

Description automatically generated

*\* Column\_name.value and number of frequencies. E.g. Kernel.color.0: 256 => in Kernel.color column there are 256 zeros (0).*

In 3) and 4), we can see both functions say MLE exists in the conventional sense in the OM. Therefore, all results reach the same conclusion.