Appendix II: Gauging influential points in LRRR

Mark de Rooij

Preamble

Load necessary libraries

```
library(mvtnorm)
library(ggplot2)
```

Population

We generate again the benchmark data set of Appendix I, a sample from a population model defined by the logistic reduced rank model. The sample has 1000 observations, 4 predictors and 3 response variables. The predictors are simply drawn from a multivariate normal distribution with mean zero and covariance matrix equal to an identity matrix (i.e., the variables are independent and have variance equal to one). For reproducibility, we set a seed.

```
N = 1000
P = 4
R = 3
set.seed(1234)
X = rmvnorm(N, rep(0,4), diag(4))
m = runif(R, min = -1, max = 1)
V = matrix(c(1,0,
             .5, .25,
             -.25, .5), 3, 2, byrow = TRUE)
```

```
V = V/sqrt(rowSums(V^2))
B = matrix(c(1, 1,
             -1, 1,
             1.25, -.75,
             -.75, -1.25), 4, 2, byrow = T)
U = X %*% B
```

The responses are drawn from the binomial distribution. Therefore, we first need to compute the canonical form, take the logistic transform to obtain probabilities and

```
theta = outer(rep(1, N), m) + X %*% B %*% t(V)
pi = plogis(theta)
Y = matrix(NA, N, R)
for(r in 1:R){ Y[, r] = rbinom(N, 1, pi[, r]) }

colnames(X) = c("x1", "x2", "x3", "x4")
colnames(Y) = c("y1", "y2", "y3")
```

```
dist2origin = sqrt(rowSums(U^2))
ix = sort(dist2origin, index.return = T)$ix
idx = c(ix[1], ix[500], ix[1000])
```

LRRR analysis

Before analyzing the data, we throw away everything except **X** and **Y**. We also load the functions for analysis.

```
rm(list=ls()[! ls() %in% c("Y","X", "ix", "idx")])
source("~/surfdrive/LogitMDA/lrrr-diagnostics/lrrr.R")
source("~/surfdrive/LogitMDA/lrrr-diagnostics/diagnosis.R")
source("~/surfdrive/LogitMDA/lmap-package/new/R/procx.R")
source("~/surfdrive/LogitMDA/lrrr-diagnostics/plot.lrrr.R")
```

To analyse the data using a logistic reduced rank model. we call the function lpca() from the lmap-package. Reduced rank regresssion can be considered a constraint or restricted PCA, where the object points lie in the column space of **X**.

```
lrrr.out = lpca(Y = Y, X = X, S = 2)
```

The fitted model has deviance 2698.78. Its estimated weight matrix is

```
round(lrrr.out$B, digits = 2)
```

```
      [,1]  [,2]
[1,] -0.57 -0.44
[2,]  0.48 -0.50
[3,] -0.57  0.45
[4,]  0.44  0.59
```

and the estimated loading matrix is

```
round(lrrr.out$V, digits = 2)
```

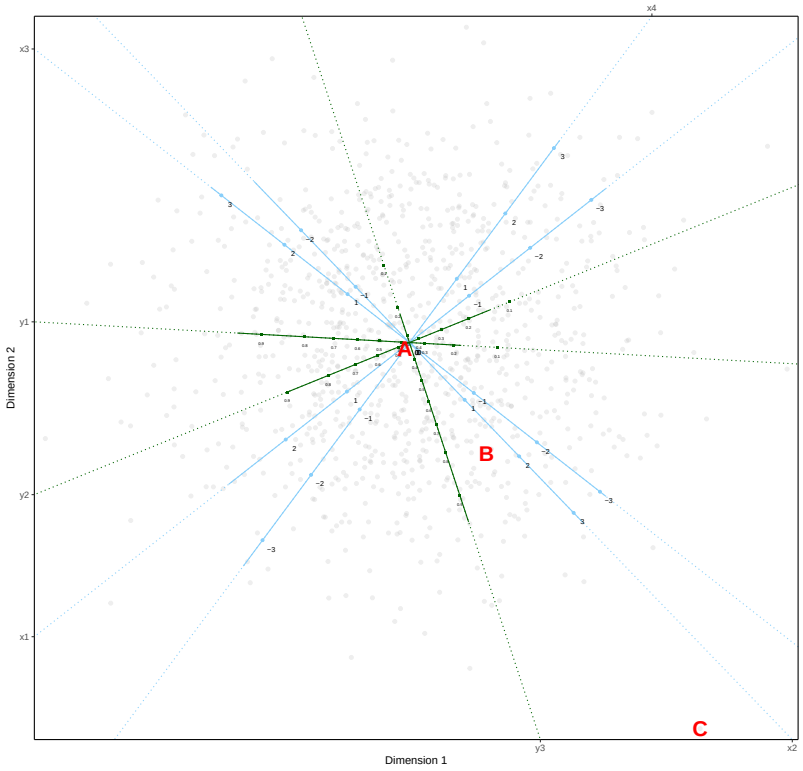
```
      [,1]  [,2]
[1,] -2.04  0.11
[2,] -1.86 -0.76
[3,]  0.62 -1.89
```

Finally, we can make a biplot.

```
a = plot.lpcah(lrrr.out, res = FALSE)
```

We highlight three cases, that we will inspect further in the Sections to come. One point in the center of the display, one on the boundary and one somewhere in the middle of the first two. The three observations are highlighted in the following biplot.

```
Udf = data.frame(lrrr.out$U[idx, ])
colnames(Udf) = c("dim1", "dim2")
a + geom_text(data = Udf, aes(x = dim1, y = dim2, label = c("A", "B", "C")), size = 7.5, fon
```



The observed responses and the estimated probabilities for these three observations are

```
lrrr.out$Y[idx, ]
```

```
     y1 y2 y3
[1,]  1  0  0
[2,]  0  1  1
[3,]  0  0  1
```

```
lrrr.out$probabilities[idx, ]
```

```
            [,1]       [,2]      [,3]
[1,] 0.381506667 0.47438970 0.3436149
[2,] 0.108502556 0.31701181 0.8349883
[3,] 0.001743394 0.07620317 0.9994890
```

Finally, the hat values are

```
H = hat.lrrr(lrrr.out)
rowSums(H[idx, -1])
```

```
        378         801         122
0.005197379 0.006719362 0.010565443
```

Observation A

What we will do now is change the response variable values for these three cases, one by one, and see the influence on the statistics. We start in this section with observation A. This participant had responses 1, 0, 0, and we change it to 0, 1, 1. As the estimated probabilities for this person are all small and the person is near the center of the biplot, that is, has about average values for the predictors, changing the outcomes should not matter much.

```
YY = Y
YY[ix[1], ] = ifelse(Y[ix[1], ] == 1, 0, 1) # point near origin
lrrr.outA = lrrr.out = lpca(Y = YY, X = X, S = 2)
```

Fit measures

First, we look at fit measures. The deviance of this model is 2699.28 compared to 2699.28 for our analysis before we altered the responses of observation A.

```
fit = fit.lrrr(lrrr.out)
fitA = fit.lrrr(lrrr.outA)
```

We can compare the fit measures for the two analysis. For the analysis with altered data and original data we obtain

```
cbind(fitA$R2.overall, fit$R2.overall)
```

```
          [,1]      [,2]
[1,] 0.3385269 0.3385269
```

```
rbind(fitA$R2.variables, fit$R2.variables)
```

```
            y1        y2        y3
[1,] 0.3394665 0.3328095 0.3434701
[2,] 0.3394665 0.3328095 0.3434701
```

```
rbind(fitA$QoR, fit$QoR)
```

```
           y1        y2        y3
[1,] 0.9988442 0.9989767 0.9997668
[2,] 0.9988442 0.9989767 0.9997668
```

We can also compare the regression weights

```
round(cbind(lrrr.outA$B, lrrr.out$B), digits = 2)
```

```
      [,1]  [,2]  [,3]  [,4]
[1,] -0.57 -0.44 -0.57 -0.44
[2,]  0.48 -0.50  0.48 -0.50
[3,] -0.57  0.45 -0.57  0.45
[4,]  0.44  0.60  0.44  0.60
```

and the loadings

```
round(cbind(lrrr.outA$V, lrrr.out$V), digits = 2)
```

```
      [,1]  [,2]  [,3]  [,4]
[1,] -2.04  0.11 -2.04  0.11
[2,] -1.86 -0.75 -1.86 -0.75
[3,]  0.62 -1.89  0.62 -1.89
```

All these values are very similar.

Residuals

```
residuals = residuals.lrrr(lrrr.outA)
residuals$pearson[idx[1], ]
```

```
        y1         y2         y3
-0.7817304  1.0479460  1.3699924
```

```
residuals$deviance[idx[1], ]
```

```
        y1         y2         y3
-0.9766459  1.2174362  1.4537580
```

Hat values

```
hatA = hat.lrrr(lrrr.outA)
hat = hat.lrrr(lrrr.out)
rbind(hatA[ix[1], ], hat[ix[1], ])
```

```
     idx         y1          y2          y3
378  378 0.00172092 0.001784141 0.001698108
3781 378 0.00172092 0.001784141 0.001698108
```

Influential Points

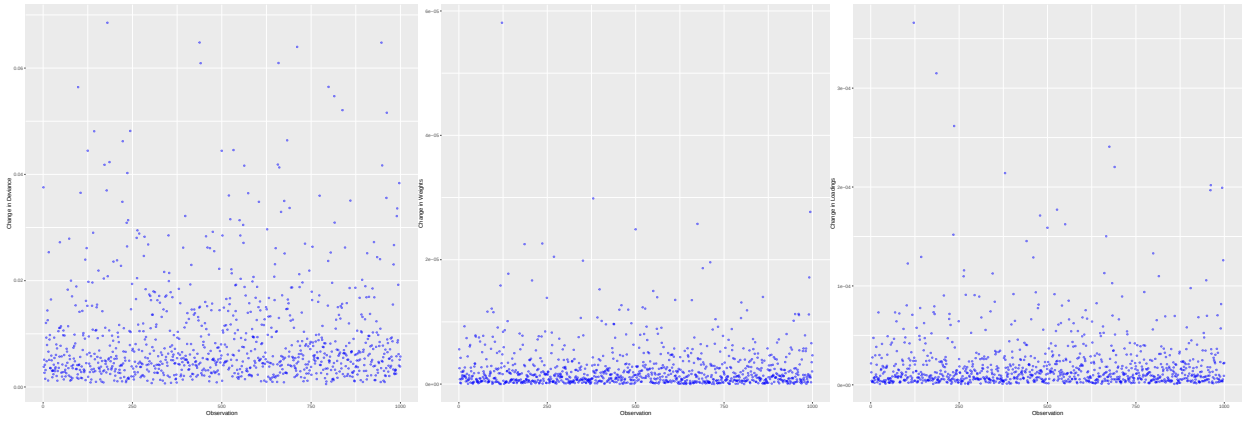
Finally, we verify whether any of the observatyions is influential. Note that observation A has number 378. We do not see anything suspicious.

```
deletion = influence.lrrr(lrrr.outA)
```

```
ggplot(deletion, aes(x = idx, y = deviance)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Deviance")

ggplot(deletion, aes(x = idx, y = B)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Weights")

ggplot(deletion, aes(x = idx, y = V)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Loadings")
```



(a) Change in Deviance (b) Change in Weights (c) Change in Loadings

Figure 1: Deletion statistics

Observation C

What we will do now is change the response variable values for these three cases, one by one, and see the influence on the statistics. We start in this section with observation A. This participant had responses 0, 0, 1, and we change it to 1, 1, 0. As the estimated probabilities for this person are all extreme, i.e., close to zero or one and the person is far from the center of the biplot, changing the outcomes should have a large influence.

```
YY = Y
YY[idx[3], ] = ifelse(Y[idx[3], ] == 1, 0, 1) # point near origin
lrrr.outC = lpca(Y = YY, X = X, S = 2)
```

Fit measures

First, we look at fit measures. The deviance of this model is 2731.04 compared to 2699.28 for our analysis before we altered the responses of observation C.

```
fit = fit.lrrr(lrrr.out)
fitC = fit.lrrr(lrrr.outC)
```

We can compare the fit measures for the two analysis. For the analysis with altered data and original data we obtain

```
cbind(fitC$R2.overall, fit$R2.overall)
```

```
          [,1]      [,2]
[1,] 0.3306822 0.3385269
```

```
rbind(fitC$R2.variables, fit$R2.variables)
```

```
           y1        y2        y3
[1,] 0.3302485 0.3293794 0.3324671
[2,] 0.3394665 0.3328095 0.3434701
```

```
rbind(fitC$QoR, fit$QoR)
```

```
           y1        y2        y3
[1,] 0.9990347 0.9991633 0.9997997
[2,] 0.9988442 0.9989767 0.9997668
```

We can also compare the regression weights

```
round(cbind(lrrr.outC$B, lrrr.out$B), digits = 2)
```

```
      [,1]  [,2]  [,3]  [,4]
[1,] -0.58 -0.42 -0.57 -0.44
[2,]  0.46 -0.52  0.48 -0.50
[3,] -0.56  0.47 -0.57  0.45
[4,]  0.46  0.58  0.44  0.60
```

and the loadings

```
round(cbind(lrrr.outC$V, lrrr.out$V), digits = 2)
```

```
      [,1]  [,2]  [,3]  [,4]
[1,] -1.98  0.17 -2.04  0.11
[2,] -1.87 -0.70 -1.86 -0.75
[3,]  0.52 -1.86  0.62 -1.89
```

We see some changes in the statistics, although at first glance not very big. Note that, once we look in real data, we might have to look at similar changes.

Residuals

```
residuals = residuals.lrrr(lrrr.outC)
residuals$pearson[idx[3], ]
```

```
       y1         y2         y3
21.258244   3.227757 -38.667515
```

```
residuals$deviance[idx[3], ]
```

```
      y1        y2        y3
3.497342  2.206911 -3.823786
```

Hat values

```
hatC = hat.lrrr(lrrr.outC)
rbind(hatC[idx[3], ], hat[idx[3], ])
```

```
     idx           y1         y2           y3
122  122 0.0004444404 0.01121805 0.0001706577
1221 122 0.0003599605 0.01012165 0.0001383651
```

Influential Points

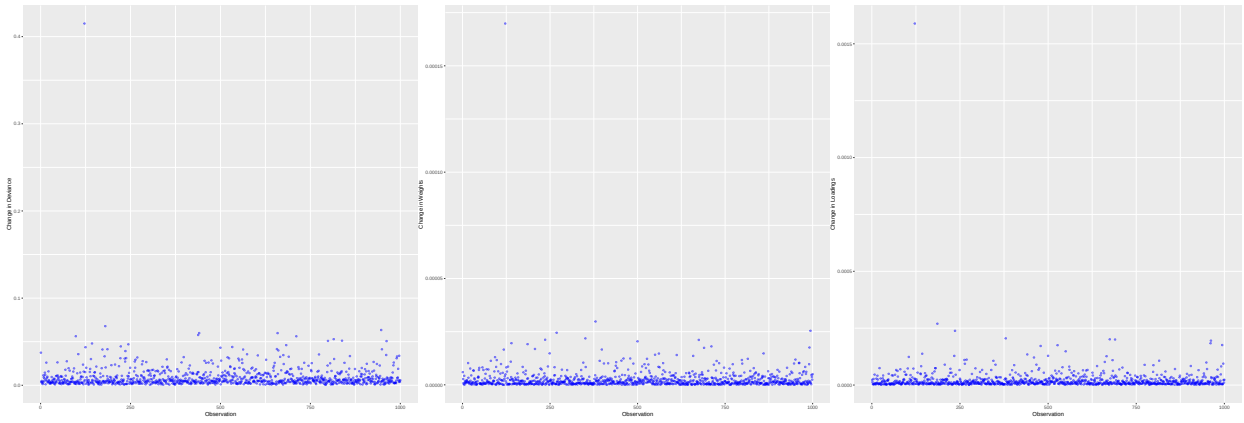
Finally, we verify whether any of the observations is influential. Note that observation C has number 122.

```r
deletion = influence.lrrr(lrrr.outC)
```

```r
ggplot(deletion, aes(x = idx, y = deviance)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Deviance")

ggplot(deletion, aes(x = idx, y = B)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Weights")

ggplot(deletion, aes(x = idx, y = V)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Loadings")
```



(a) Change in Deviance (b) Change in Weights (c) Change in Loadings

Figure 2: Deletion statistics

Observation B

What we will do now is change the response variable values for these three cases, one by one, and see the influence on the statistics. We start in this section with observation B. This participant had responses 0, 1, 1, and we change it to 1, 1, 0. Here, we do not really know what to expect.

```
YY = Y
YY[idx[2], ] = ifelse(Y[idx[2], ] == 1, 0, 1) # point near origin
lrrr.outB = lpca(Y = YY, X = X, S = 2)
```

Fit measures

First, we look at fit measures. The deviance of this model is 2704.65 compared to 2699.28 for our analysis before we altered the responses of observation C.

```
fit = fit.lrrr(lrrr.out)
fitB = fit.lrrr(lrrr.outB)
```

We can compare the fit measures for the two analysis. For the analysis with altered data and original data we obtain

```
cbind(fitB$R2.overall, fit$R2.overall)
```

```
          [,1]      [,2]
[1,] 0.3370664 0.3385269
```

```
rbind(fitB$R2.variables, fit$R2.variables)
```

```
           y1        y2        y3
[1,] 0.3362856 0.3338042 0.3412245
[2,] 0.3394665 0.3328095 0.3434701
```

```
rbind(fitB$QoR, fit$QoR)
```

```
           y1        y2        y3
[1,] 0.9990236 0.9991549 0.9998097
[2,] 0.9988442 0.9989767 0.9997668
```

We can also compare the regression weights

```
round(cbind(lrrr.outB$B, lrrr.out$B), digits = 2)
```

```
      [,1]  [,2]  [,3]  [,4]
[1,] -0.57 -0.44 -0.57 -0.44
[2,]  0.48 -0.50  0.48 -0.50
[3,] -0.57  0.45 -0.57  0.45
[4,]  0.45  0.59  0.44  0.60
```

and the loadings

```
round(cbind(lrrr.outB$V, lrrr.out$V), digits = 2)
```

```
      [,1]  [,2]  [,3]  [,4]
[1,] -2.02  0.12 -2.04  0.11
[2,] -1.88 -0.73 -1.86 -0.75
[3,]  0.60 -1.88  0.62 -1.89
```

Residuals

```
residuals = residuals.lrrr(lrrr.outB)
residuals$pearson[idx[2], ]
```

```
       y1         y2         y3
2.8274090 -0.6752169 -2.2245095
```

```
residuals$deviance[idx[2], ]
```

```
       y1         y2         y3
2.0959888 -0.8667602 -1.8884541
```

Hat values

```
hatB = hat.lrrr(lrrr.outB)
rbind(hatB[idx[2], ], hat[idx[2], ])
```

```
     idx          y1          y2          y3
801  801 0.001529931 0.002307127 0.002900011
8011 801 0.001504949 0.002317110 0.002881778
```

Influential Points

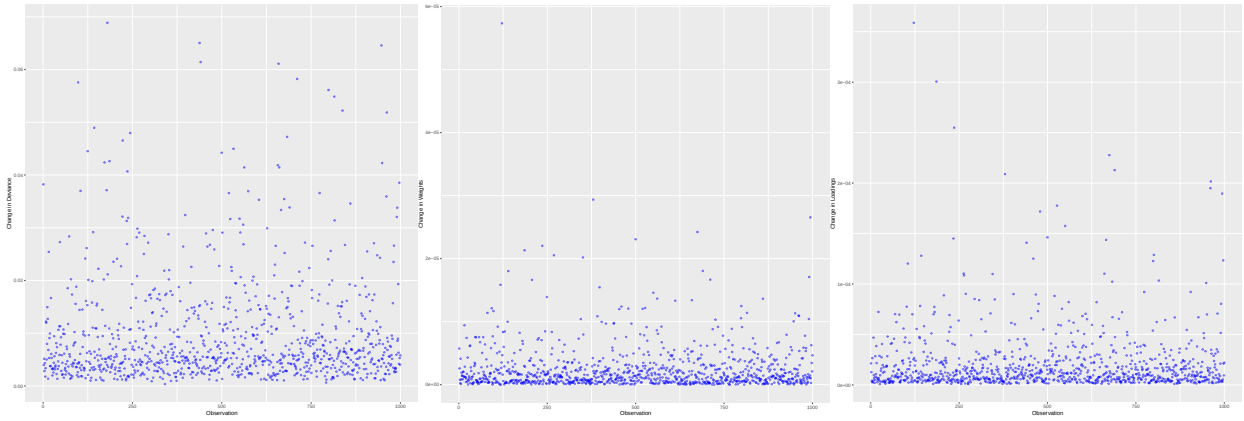
Finally, we verify whether any of the observations is influential. Note that observation B has number 801.

```
deletion = influence.lrrr(lrrr.outB)
```

```
ggplot(deletion, aes(x = idx, y = deviance)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Deviance")

ggplot(deletion, aes(x = idx, y = B)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Weights")

ggplot(deletion, aes(x = idx, y = V)) +
  geom_point(size = 1, col = "blue", alpha = 0.5) +
  labs(x = "Observation",
       y = "Change in Loadings")
```



(a) Change in Deviance (b) Change in Weights (c) Change in Loadings

Figure 3: Deletion statistics