Appendix III: Gauging non-linearity points in LRRR

Mark de Rooij

Preamble

Load necessary libraries

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

Population

We generate 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))
X2 = cbind(X, X[, 4] * X[, 4])
X3 = cbind(X2, X[, 4] * X[, 4] * X[, 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)
B2 = rbind(B, c(.25, -.25))
B3 = rbind(B2, c(-.15, .15))
```

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]) }

theta = outer(rep(1, N), m) + X2 %*% B2 %*% t(V)
pi = plogis(theta)
Y2 = matrix(NA, N, R)
for(r in 1:R){ Y2[, r] = rbinom(N, 1, pi[, r]) }

theta = outer(rep(1, N), m) + X3 %*% B3 %*% t(V)
pi = plogis(theta)
Y3 = matrix(NA, N, R)
for(r in 1:R){ Y3[, r] = rbinom(N, 1, pi[, r]) }

colnames(X) = c("x1", "x2", "x3", "x4")
colnames(Y) = colnames(Y2) = colnames(Y3) = c("y1", "y2", "y3")
```

LRRR analysis

Before analyzing the data, we throw away everything except $\mathbf{X}$ and $\mathbf{Y}$. We also load the functions for analysis.

```
rm(list=ls()[! ls() %in% c("Y","Y2" ,"Y3","X")])
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 $\mathbf{X}$.

```
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.

```
plot.lpcah(lrrr.out, ocol = "red")
```

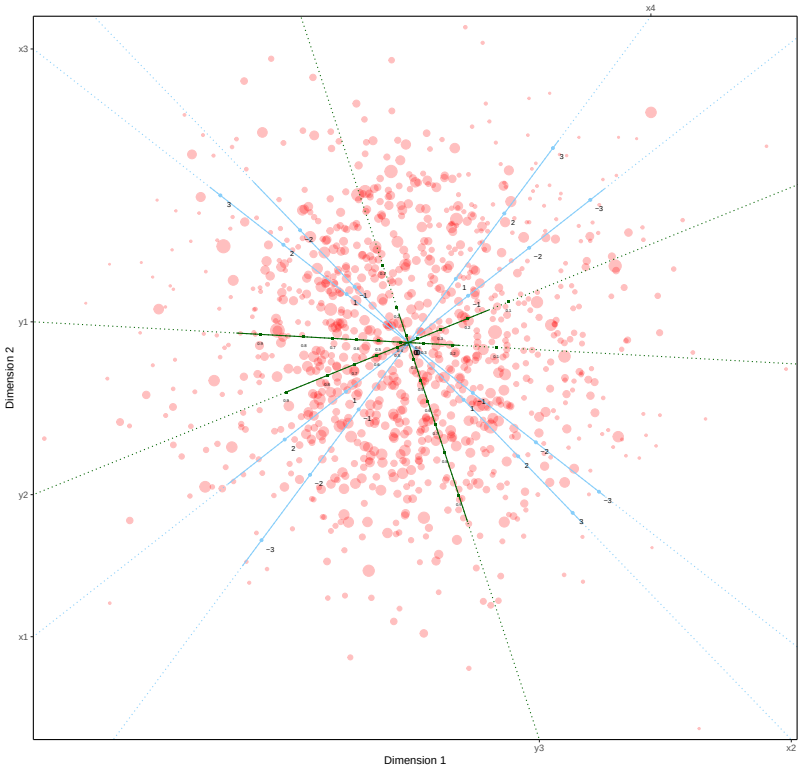


Figure 1: Triplot of Benchmark data

In the biplot, we already visualized the deviance residuals (see next Section) by the area of the points. The larger the point, the higher the contribution of that observation to the total deviance. The diagnostics corresponding to this analysis are shown in Appendix I.

Hidden quadratic effect of predictor 4

Here, we analyse the data that are generated with the quadratic effect, while our model only assumes linear relationships.

```
lrrr.out2 = lpca(Y = Y2, X = X, S = 2)
```

The fitted model has deviance 2729.11. Its estimated weight matrix is

```
round(lrrr.out2$B, digits = 2)
```

```
      [,1]  [,2]
[1,] -0.53 -0.49
[2,]  0.47 -0.48
[3,] -0.65  0.38
[4,]  0.38  0.62
```

and the estimated loading matrix is

```
round(lrrr.out2$V, digits = 2)
```

```
      [,1]  [,2]
[1,] -1.96  0.20
[2,] -1.82 -0.95
[3,]  0.79 -1.69
```

Finally, we can make a biplot.

```
plot.lpcah(lrrr.out2, ocol = "red")
```

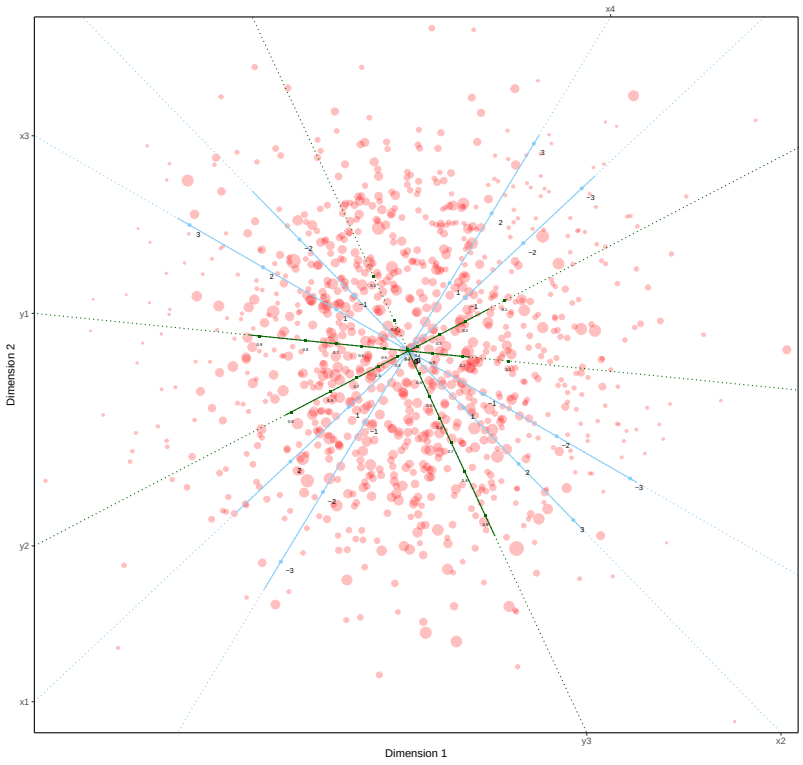


Figure 2: Triplot of Benchmark data

Non-linearity

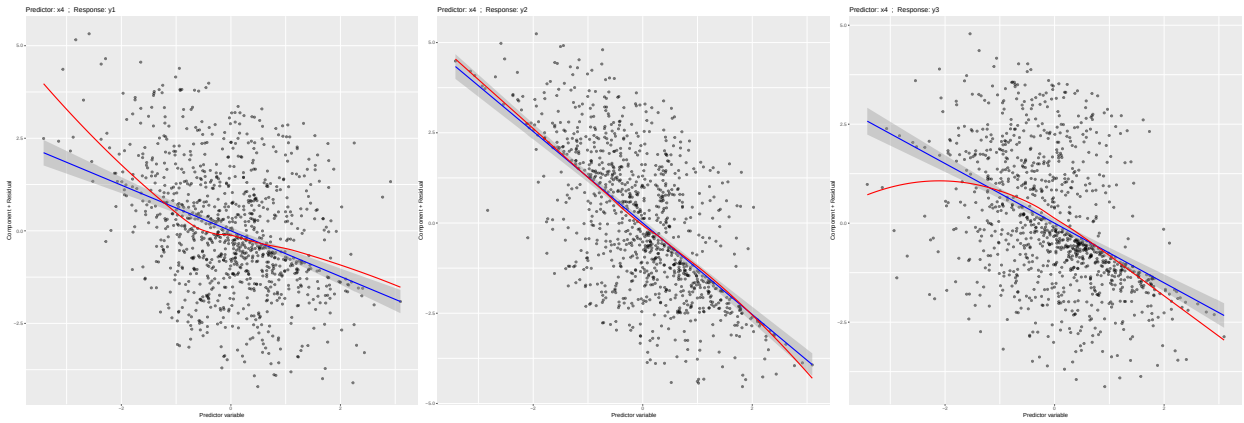
We are specifically interested in the question whether the component plus residual plots pick up the nonlinearity. Therefore, we look at the plots corresponding to the fourth predictor. We look at the plots for the first two response variables.

```
ggplot(df_partialresidual41, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out2$xnames[4], " ; ",
                     "Response:", lrrr.out2$ynames[1]),
       x = "Predictor variable",
       y = "Component + Residual")

```

```r
ggplot(df_partialresidual42, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out2$xnames[4], " ; ",
                     "Response:", lrrr.out2$ynames[2]),
       x = "Predictor variable",
       y = "Component + Residual")

ggplot(df_partialresidual43, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out2$xnames[4], " ; ",
                     "Response:", lrrr.out2$ynames[3]),
       x = "Predictor variable",
       y = "Component + Residual")
```



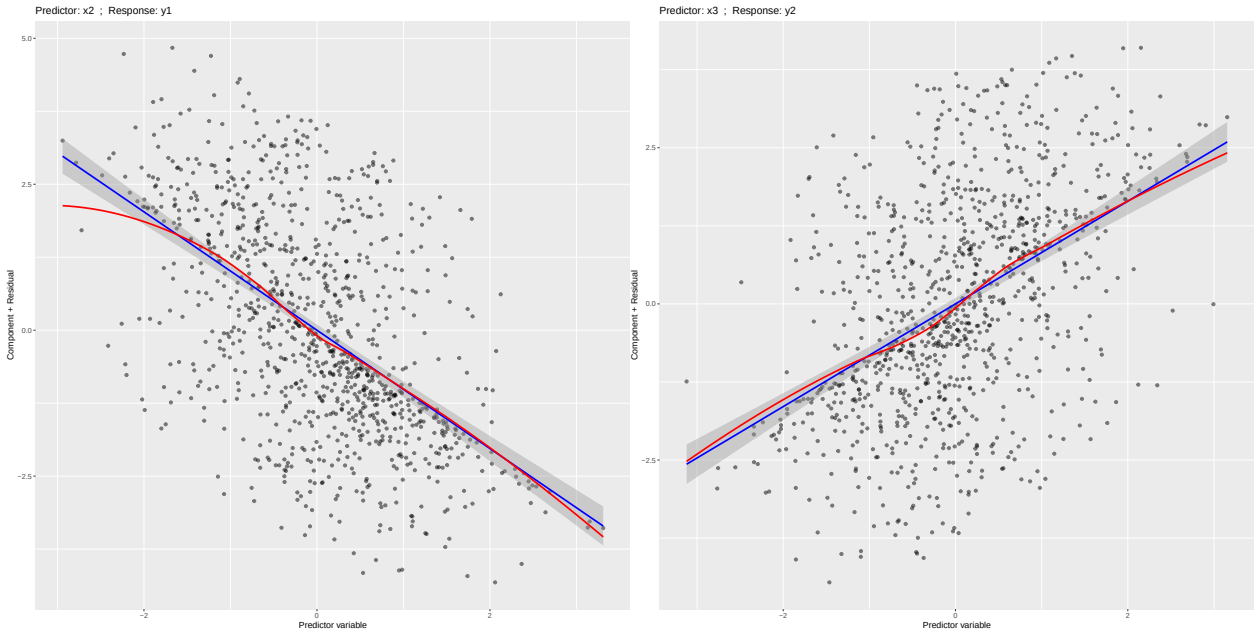
(a) First response (b) Second response (c) Third response

Figure 3: Component plus residual plots for non-linear relationships

To verify whether other predictor-response relationships are influenced by the nonlinearity of the fourth predictor we also make the component plus resdiaul plots for the ccombinations 2-1 and 3-2.

```r
ggplot(df_partialresidual21, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out2$xnames[2], " ; ",
                     "Response:", lrrr.out2$ynames[1]),
       x = "Predictor variable",
       y = "Component + Residual")

ggplot(df_partialresidual32, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out2$xnames[3], " ; ",
                     "Response:", lrrr.out2$ynames[2]),
       x = "Predictor variable",
       y = "Component + Residual")
```



(a) Second predictor - First response

(b) Third predictor - Second response

Figure 4: Component plus residual plots for linear relationships

Hidden cubic effect of predictor 4

Here, we analyse the data that are generated with the quadratic effect, while our model only assumes linear relationships.

```
lrrr.out3 = lpca(Y = Y3, X = X, S = 2)
```

The fitted model has deviance 2758.79. Its estimated weight matrix is

```
round(lrrr.out3$B, digits = 2)
```

```
      [,1]  [,2]
[1,] -0.56 -0.48
[2,]  0.38 -0.56
[3,] -0.56  0.54
[4,]  0.56  0.43
```

and the estimated loading matrix is

```
round(lrrr.out3$V, digits = 2)
```

```
      [,1]  [,2]
[1,] -1.80  0.17
[2,] -2.18 -0.59
[3,]  0.61 -1.60
```

Finally, we can make a biplot.

```
plot.lpcah(lrrr.out3, ocol = "red")
```

Figure 5: Triplot of Benchmark data

Non-linearity

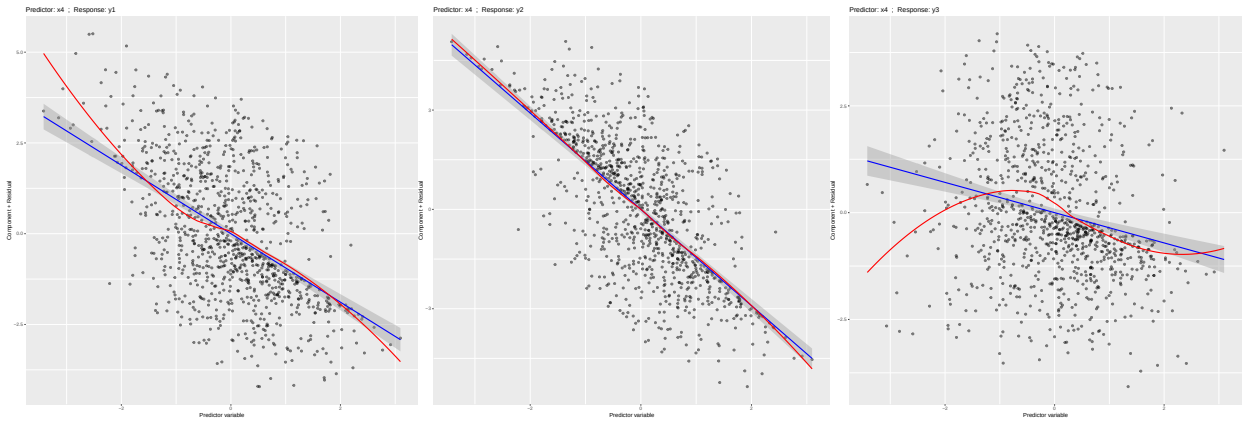
We are specifically interested in the question whether the component plus residual plots pick up the nonlinearity. Therefore, we look at the plots corresponding to the fourth predictor. We look at the plots for the first two response variables.

```
ggplot(df_partialresidual41, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out3$xnames[4], " ; ",
                     "Response:", lrrr.out3$ynames[1]),
       x = "Predictor variable",
       y = "Component + Residual")

```

```r
ggplot(df_partialresidual42, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out3$xnames[4], " ; ",
                     "Response:", lrrr.out3$ynames[2]),
       x = "Predictor variable",
       y = "Component + Residual")

ggplot(df_partialresidual43, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out3$xnames[4], " ; ",
                     "Response:", lrrr.out3$ynames[3]),
       x = "Predictor variable",
       y = "Component + Residual")
```



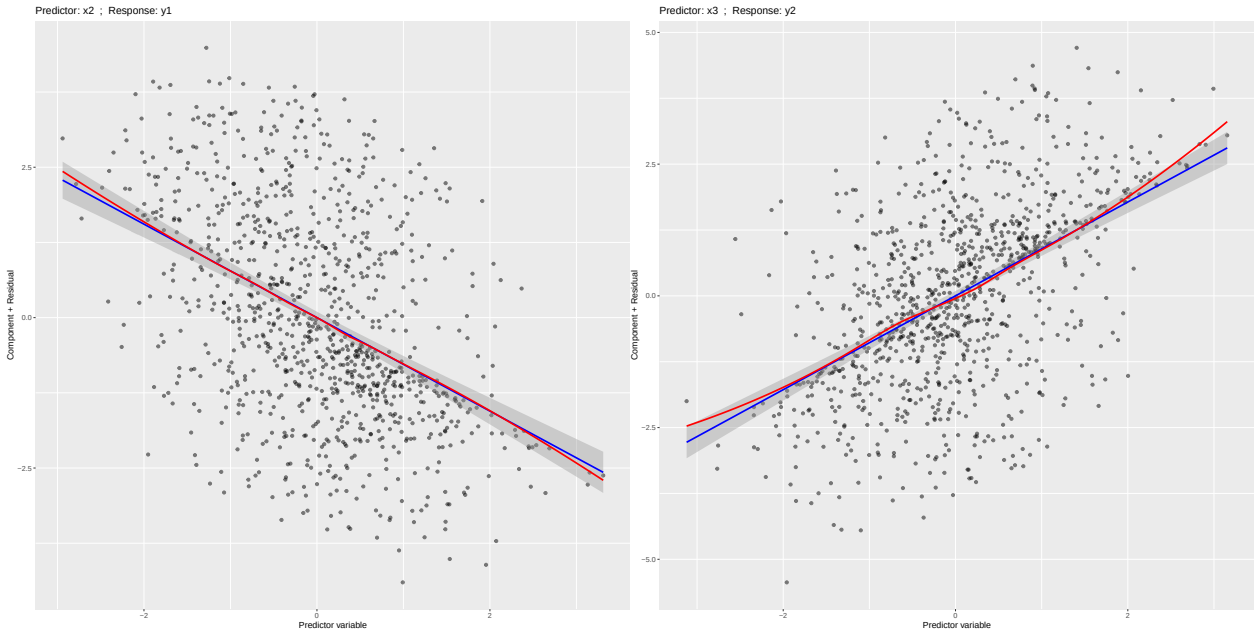
(a) First response (b) Second response (c) Third response

Figure 6: Component plus residual plots for non-linear relationships

To verify whether other predictor-response relationships are influenced by the nonlinearity of the fourth predictor we also make the component plus resdiaul plots for the ccombinations 2-1 and 3-2.

```r
ggplot(df_partialresidual21, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out2$xnames[2], " ; ",
                     "Response:", lrrr.out2$ynames[1]),
       x = "Predictor variable",
       y = "Component + Residual")

ggplot(df_partialresidual32, aes(x = Predictor, y = Partial_Residual)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, col = "blue") +
  geom_smooth(method = loess,  formula = y ~ x, se = FALSE, col = "red") +
  labs(title = paste("Predictor:", lrrr.out2$xnames[3], " ; ",
                     "Response:", lrrr.out2$ynames[2]),
       x = "Predictor variable",
       y = "Component + Residual")
```



(a) Second predictor - First response

(b) Third predictor - Second response

Figure 7: Component plus residual plots for linear relationships