PML 8

2023-05-20

Opgave 1

Vi kigger på *phoneme* datasættet og vil så på hvordan xgboost og glmnet virker.

1)

Jeg indlæsser datasættet.

```
# Indlæs datasættet
phoneme <- read.csv("phoneme.data.txt")

# Subset kun de rækker, hvor g er "aa" eller "ao"
phoneme <- subset(phoneme, g=="aa" | g=="ao")

# Opret ph2 data frame med de ønskede kolonner
ph2 <- data.frame(X = I(as.matrix(phoneme[, 2:257])),
                  g = factor(phoneme$g, levels = c("aa", "ao")))

# Definer antallet af rækker som N
N <- nrow(ph2)

# Opret træningssæt og testsæt
train <- sample(1:N, 1000)
test <- setdiff(1:N, train)
```

2)

Vi vil nu kigge på xgboost.

```
library(xgboost)

ph2.test <- ph2[test, ]
ph2.train <- ph2[train, ]
y_train <- as.numeric(ph2.train$g) - 1  # Konverter faktoren til 0/1 variabel
bst <- xgboost(data = as.matrix(ph2.train$X), label = y_train,
               max.depth = 2, eta = 2, nthread = 2,
               nrounds = 100, objective = "binary:logistic")
```

```
## [1]  train-logloss:0.513311
## [2]  train-logloss:0.782934
## [3]  train-logloss:0.935249
```

```
## [4]  train-logloss:4.321315
## [5]  train-logloss:6.063531
## [6]  train-logloss:18.294065
## [7]  train-logloss:6.084318
## [8]  train-logloss:6.352397
## [9]  train-logloss:6.373291
## [10] train-logloss:5.674926
## [11] train-logloss:16.137843
## [12] train-logloss:7.770450
## [13] train-logloss:8.252671
## [14] train-logloss:8.252671
## [15] train-logloss:8.252671
## [16] train-logloss:8.252671
## [17] train-logloss:8.252671
## [18] train-logloss:8.252671
## [19] train-logloss:8.252671
## [20] train-logloss:8.252671
## [21] train-logloss:8.252671
## [22] train-logloss:8.252671
## [23] train-logloss:8.252671
## [24] train-logloss:8.252671
## [25] train-logloss:8.252671
## [26] train-logloss:8.252671
## [27] train-logloss:8.252671
## [28] train-logloss:8.252671
## [29] train-logloss:8.252671
## [30] train-logloss:8.252671
## [31] train-logloss:8.252671
## [32] train-logloss:8.252671
## [33] train-logloss:8.252671
## [34] train-logloss:8.252671
## [35] train-logloss:8.252671
## [36] train-logloss:8.252671
## [37] train-logloss:8.252671
## [38] train-logloss:8.252671
## [39] train-logloss:8.252671
## [40] train-logloss:8.252671
## [41] train-logloss:8.252671
## [42] train-logloss:8.252671
## [43] train-logloss:8.252671
## [44] train-logloss:8.252671
## [45] train-logloss:8.252671
## [46] train-logloss:8.252671
## [47] train-logloss:8.252671
## [48] train-logloss:8.252671
## [49] train-logloss:8.252671
## [50] train-logloss:8.252671
## [51] train-logloss:8.252671
## [52] train-logloss:8.252671
## [53] train-logloss:8.252671
## [54] train-logloss:8.252671
## [55] train-logloss:8.252671
## [56] train-logloss:8.252671
## [57] train-logloss:8.252671
```

```
## [58] train-logloss:8.252671
## [59] train-logloss:8.252671
## [60] train-logloss:8.252671
## [61] train-logloss:8.252671
## [62] train-logloss:8.252671
## [63] train-logloss:8.252671
## [64] train-logloss:8.252671
## [65] train-logloss:8.252671
## [66] train-logloss:8.252671
## [67] train-logloss:8.252671
## [68] train-logloss:8.252671
## [69] train-logloss:8.252671
## [70] train-logloss:8.252671
## [71] train-logloss:8.252671
## [72] train-logloss:8.252671
## [73] train-logloss:8.252671
## [74] train-logloss:8.252671
## [75] train-logloss:8.252671
## [76] train-logloss:8.252671
## [77] train-logloss:8.252671
## [78] train-logloss:8.252671
## [79] train-logloss:8.252671
## [80] train-logloss:8.252671
## [81] train-logloss:8.252671
## [82] train-logloss:8.252671
## [83] train-logloss:8.252671
## [84] train-logloss:8.252671
## [85] train-logloss:8.252671
## [86] train-logloss:8.252671
## [87] train-logloss:8.252671
## [88] train-logloss:8.252671
## [89] train-logloss:8.252671
## [90] train-logloss:8.252671
## [91] train-logloss:8.252671
## [92] train-logloss:8.252671
## [93] train-logloss:8.252671
## [94] train-logloss:8.252671
## [95] train-logloss:8.252671
## [96] train-logloss:8.252671
## [97] train-logloss:8.252671
## [98] train-logloss:8.252671
## [99] train-logloss:8.252671
## [100]    train-logloss:8.252671
```

Koden gør følgende:

data: Inputdataen, der forventes som en matrix. Vi bruger as.matrix(ph2.train$X) til at konvertere dataen til en matrix.

label: Vektor af målværdierne. Vi bruger y_train, som er den konverterede 0/1-variabel baseret på ph2.train$g.

max.depth: Maksimal dybde af hver træ-node. I dette tilfælde er den sat til 2.

eta: Læringshastigheden (også kendt som "learning rate"). Den styrer, hvor hurtigt modellen tilpasser sig dataene.

nthread: Antal tråde til at køre xgboost-træningen. Dette kan indstilles baseret på dine systemressourcer.

nrounds: Antal træningsrunder (også kendt som "boosting iterations"). Jo højere værdi, desto flere træningsrunder udføres, hvilket kan forbedre præstationen. Du kan øge denne værdi til f.eks. 100 eller mere.

objective: Den objektive funktion, der bruges under træningen. I dette tilfælde bruger vi "binary:logistic" for binær logistisk regression.

Det anbefales at justere nrounds-parameteren til en højere værdi (f.eks. 100 eller mere), da 2 runder sandsynligvis ikke vil være tilstrækkeligt til at opnå gode resultater.

3)

```
# Foretag forudsigelser på testsættet
xgb_pred <- predict(bst, as.matrix(ph2.test$X))

# Konverter sandsynligheder til klasseetiketter (0 eller 1)
xgb_pred_labels <- ifelse(xgb_pred > 0.5, 1, 0)

# Opret forvirringsmatrix
confusion_matrix <- table(xgb_pred_labels, ph2.test$g)

# Vis forvirringsmatrix
confusion_matrix
```

```
##
## xgb_pred_labels  aa  ao
##               0 157  42
##               1 131 387
```

I dette eksempel antages det, at en sandsynlighed større end 0,5 betyder klasse 1, og en sandsynlighed mindre end eller lig med 0,5 betyder klasse 0. Vi kan justere tærskelværdien efter behov.

Når forvirringsmatricen er oprettet, kan vi analysere resultaterne og evaluere modellens præstation. For eksempel kan vi beregne nøjagtighed, præcision, recall osv. baseret på forvirringsmatricen.

Vi kan eksperimentere med forskellige parametre i xgboost-kaldet for at se, hvordan de påvirker modellens præstation. Specielt kan eta-parameteren (læringshastigheden) have en stor indflydelse. En høj eta-værdi kan føre til hurtigere indlæring, men det kan også øge risikoen for overfit.

4)

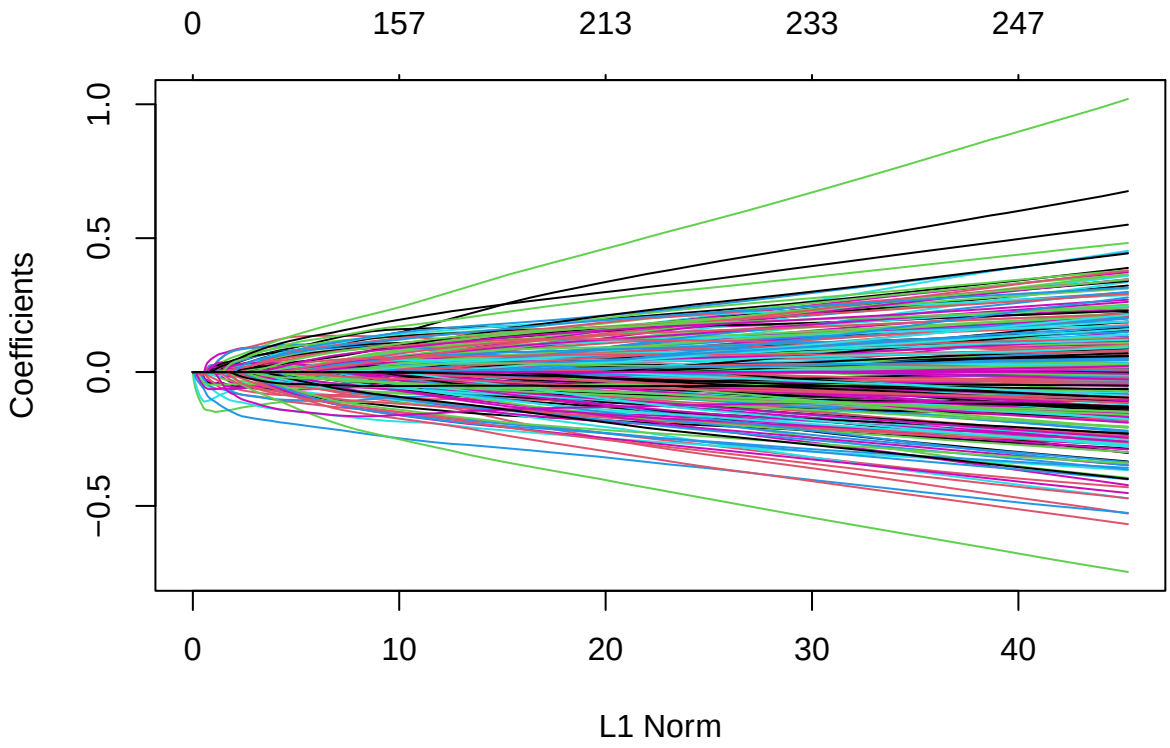
Vi vil nu se på glmnet.

```
library(glmnet)
```

```
## Indlæser krævet pakke: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
gnet <- glmnet(ph2.train$X, ph2.train$g, family = "binomial")
plot(gnet)
```



Grafen har lambda-værdien (logaritmisk skala) på x-aksen og koefficientværdierne på y-aksen. Hver linje i grafen repræsenterer en variabel og dens koefficientværdi som funktion af lambda.

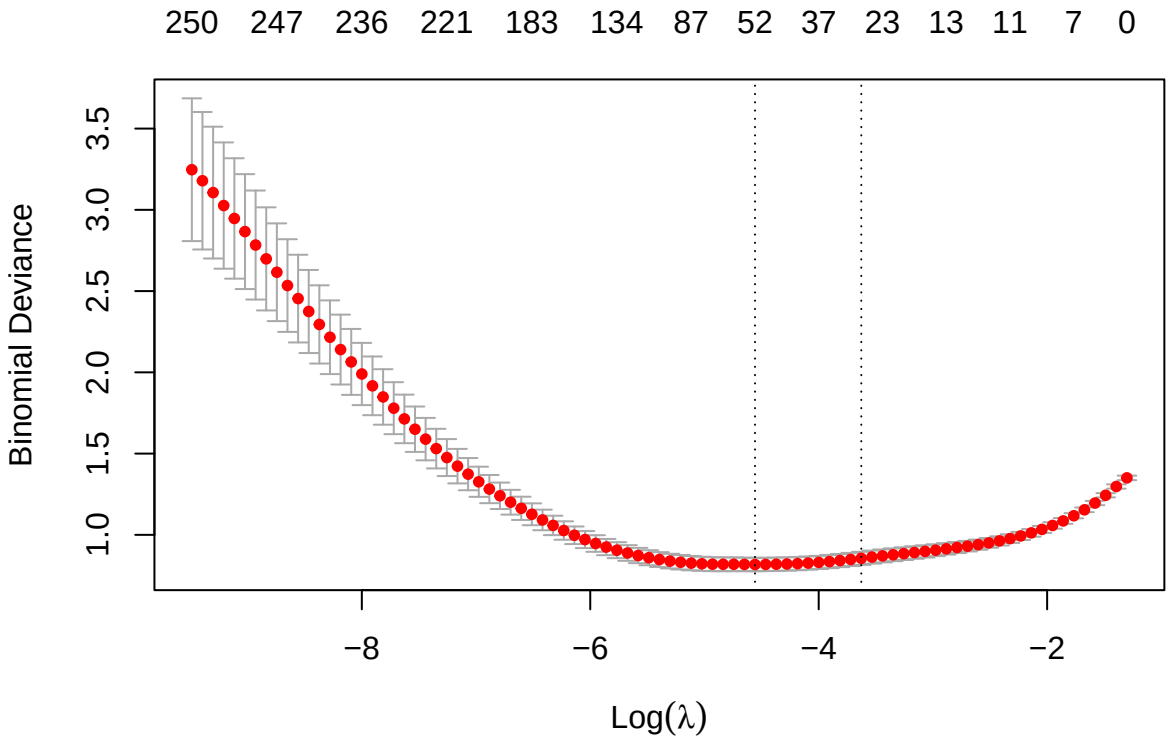
Regularisering er en metode til at kontrollere modellens kompleksitet og undgå overfitting ved at tilføje en "straf" til koefficienterne. I glmnet bruger man en form for regularisering kaldet Lasso-penalty, hvilket betyder, at lambda-værdien styrer graden af straf (regularisering) på koefficienterne. Jo højere lambda-værdi, desto mere straf på koefficienterne, hvilket fører til en mere restriktiv model.

Grafen viser, hvordan koefficienterne for hver variabel ændrer sig, når lambda-værdien ændrer sig. Vi kan observere, hvilke variabler der har nonzero koefficienter ved forskellige lambda-værdier og dermed få en idé om, hvilke variabler der har størst indflydelse på modellen. Grafen kan også hjælpe med at identificere en passende værdi for lambda, der balancerer modelens præcision og kompleksitet.

5)

Vi laver nu CV:

```
CV1 <- cv.glmnet(ph2.train$X, ph2.train$g, family = "binomial")
plot(CV1)
```

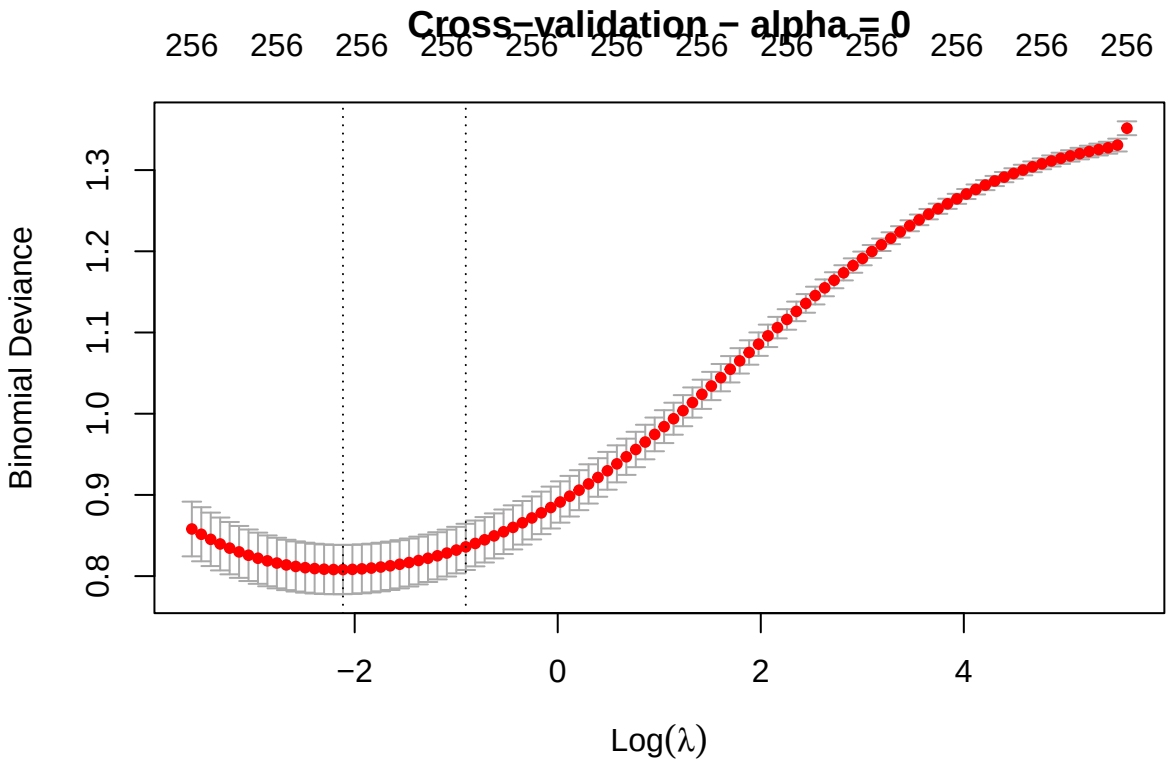


6)

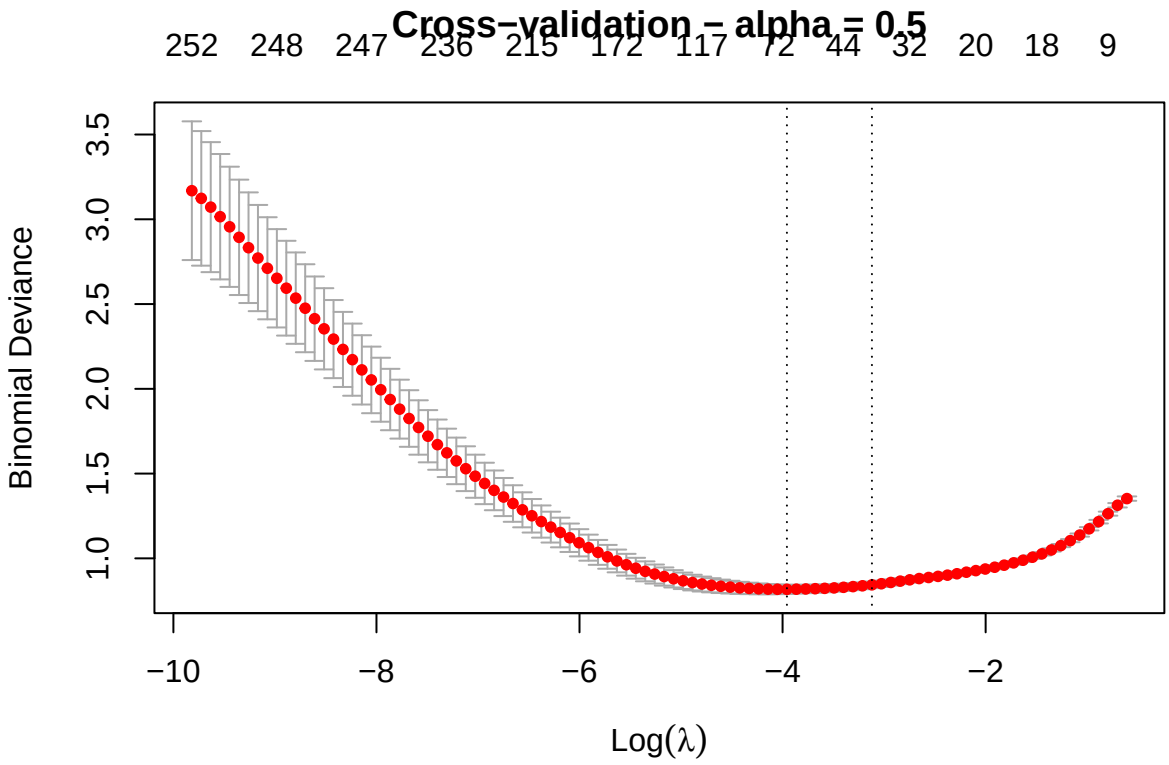
Vi laver igen CV, men for $\alpha = 0$ og $\alpha = 0.5$:

```
CV_alpha0 <- cv.glmnet(ph2.train$X, ph2.train$g, family = "binomial", alpha = 0)
CV_alpha0.5 <- cv.glmnet(ph2.train$X, ph2.train$g, family = "binomial", alpha = 0.5)

plot(CV_alpha0, main = "Cross-validation - alpha = 0")
```




```
plot(CV_alpha0.5, main = "Cross-validation - alpha = 0.5")
```



7)

I figurerne genereret af cv.glmnet-funktionen i glmnet-pakken vil vi bemærke to lodrette linjer, der markerer positionen af "lambda.min" og "lambda.1se". Disse linjer er nyttige for at vælge den optimale lambda-værdi baseret på krydsvalidering.

1. "lambda.min": Dette er lambda-værdien, der er forbundet med den mindste krydsvalideringsfejl. Det er den lambda-værdi, der typisk vælges som den optimale værdi. Den repræsenterer den mest restriktive lambda-værdi, hvor modellen stadig har en god præstation.
2. "lambda.1se": Dette er lambda-værdien, der vælges ved hjælp af en en standardfejlsregel (one-standard-error rule). Denne regel forsøger at vælge en mindre restriktiv lambda-værdi, der er inden for en standardfejl af lambda-værdien, der giver den laveste krydsvalideringsfejl. "lambda.1se" værdien er ofte lidt mindre restriktiv end "lambda.min" og kan være mere passende, hvis vi ønsker en mindre kompleks model med en acceptabel præstation.

De lodrette linjer for "lambda.min" og "lambda.1se" er nyttige for at identificere, hvor de tilsvarende lambda-værdier ligger på grafen og hjælper os med at vælge den optimale lambda-værdi baseret på vores præferencer og behov. Det er vigtigt at bemærke, at valget af lambda-værdien afhænger af vores specifikke situation og præference for modelens præcision og kompleksitet.

8)

Vi fitter for de optimale værdier af lambda.min og lambda.1se. Tærskelværdien er sat til 50%

```r
# Træn glmnet-modellen med krydsvalidering og få lambda-værdierne
cv_fit <- cv.glmnet(ph2.train$X, ph2.train$g, family = "binomial")
lambda_min <- cv_fit$lambda.min
lambda_1se <- cv_fit$lambda.1se

# Få koefficienterne for lambda.min og lambda.1se
coef_min <- coef(cv_fit, s = "lambda.min")
coef_1se <- coef(cv_fit, s = "lambda.1se")

# Forudsig på testsættet ved hjælp af lambda.min
pred_min <- predict(cv_fit, newx = ph2.test$X, s = lambda_min, type = "response")
pred_labels_min <- ifelse(pred_min > 0.5, 1, 0)

# Forudsig på testsættet ved hjælp af lambda.1se
pred_1se <- predict(cv_fit, newx = ph2.test$X, s = lambda_1se, type = "response")
pred_labels_1se <- ifelse(pred_1se > 0.5, 1, 0)

# Opret forvirringsmatricer for lambda.min og lambda.1se
confusion_matrix_min <- table(pred_labels_min, ph2.test$g)
confusion_matrix_1se <- table(pred_labels_1se, ph2.test$g)

# Vis forvirringsmatricer
confusion_matrix_min
```

```
##
## pred_labels_min  aa  ao
##               0 214  48
##               1  74 381
```

```r
confusion_matrix_1se
```

```
##
## pred_labels_1se  aa  ao
##               0 206  44
##               1  82 385
```

9)

Koden, der er angivet, forsøger at parametrisere koefficienterne β_i i glmnet-modellen ved hjælp af successive differenser.

```r
library(MASS)

B <- cbind(1, contr.sdif(256))
XB <- ph2.train$X %*% B
CVsdif <- cv.glmnet(XB, ph2.train$g, family = "binomial")
```

```
## Warning: from glmnet C++ code (error code -100); Convergence for 100th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

1. library(MASS): Denne linje indlæser MASS-pakken, der indeholder funktionen contr.sdif, som bruges til at generere successive differenser.

2. B <- cbind(1, contr.sdif(256)): Her oprettes en designmatrix B ved at binde sammen en kolonne med etere for konstantleddet (1) og successive differenser genereret ved hjælp af contr.sdif(256). Antallet 256 angiver antallet af faktorniveauer eller antallet af kategorier i den faktor, der repræsenteres af de successive differenser. Resultatet er en matrix B med 256 kolonner, hvor hver kolonne repræsenterer successive differenser for den tilsvarende faktorværdi.

3. XB <- ph2.trainXX med B for at opnå XB, som er det resulterende produkt af krydsningen mellem de beskrivende variable og successive differenser. Dette giver en ny designmatrix, der parametriserer koefficienterne ved hjælp af successive differenser.

4. CVsdif <- cv.glmnet(XB, ph2.train$g, family = "binomial") : $Herudf\varnothing reskrydsvalideringmedglmnet-modellenvedhj\_lpafXBsominputdataogph2.traing$ som responsvariabel. Familien er angivet som "binomial" for binær logistisk regression. cv.glmnet udfører krydsvalidering for at vælge den optimale lambda-værdi baseret på krydsvalideringsfejlen.

For at forstå betydningen af successive differenser kan vi se på et eksempel:

```
solve(cbind(1, contr.sdif(10)))
```

```
##         1    2    3    4    5    6    7    8    9  10
##       0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1 0.1
## 2-1  -1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 0.0
## 3-2   0.0 -1.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0 0.0
## 4-3   0.0  0.0 -1.0  1.0  0.0  0.0  0.0  0.0  0.0 0.0
## 5-4   0.0  0.0  0.0 -1.0  1.0  0.0  0.0  0.0  0.0 0.0
## 6-5   0.0  0.0  0.0  0.0 -1.0  1.0  0.0  0.0  0.0 0.0
## 7-6   0.0  0.0  0.0  0.0  0.0 -1.0  1.0  0.0  0.0 0.0
## 8-7   0.0  0.0  0.0  0.0  0.0  0.0 -1.0  1.0  0.0 0.0
## 9-8   0.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0  1.0 0.0
## 10-9  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0 1.0
```

Her oprettes en designmatrix ved at binde sammen en kolonne med etere for konstantleddet (1) og successive differenser genereret af contr.sdif(10). solve-funktionen anvendes til at beregne inversen af denne designmatrix. Resultatet er en matrix, der viser, hvordan koefficienterne β_i parametriseres ved hjælp af successive differenser.

Ved at anvende successive differenser på designmatricen i glmnet-modellen, søger man at opnå en mere struktureret og "pæn" funktion for β_i, hvor koefficienterne antages at være mere sammenhængende og mindre vilkårlige i forhold til faktorværdierne.

10)

Når man kører coef(CVsdif) på den krydsvaliderede glmnet-model med successive differenser, vil vi få koefficienterne for hver lambda-værdi. Bemærk, at i glmnet-modellen med successive differenser vil den første koefficient (intercept) blive udeladt fra resultatet, da den repræsenteres af konstantleddet 1 i designmatricen.

```
coef_values <- coef(CVsdif)
coef_values
```

```
## 257 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  1.2775624780
##             -0.0006531309
## 2-1          .
## 3-2          .
## 4-3          .
## 5-4          .
## 6-5          .
## 7-6          .
## 8-7         -0.0028426224
## 9-8          .
## 10-9         .
## 11-10        .
## 12-11        .
## 13-12        .
## 14-13        .
## 15-14        .
## 16-15        .
## 17-16        .
## 18-17        .
## 19-18       -0.0066507170
## 20-19        .
## 21-20        .
## 22-21        .
## 23-22        .
## 24-23        .
## 25-24        .
## 26-25        .
## 27-26        .
## 28-27        .
## 29-28        .
## 30-29        .
## 31-30        .
## 32-31        .
## 33-32        .
## 34-33        .
## 35-34       -0.0338098579
## 36-35       -0.0108163221
## 37-36        .
## 38-37        .
## 39-38        .
## 40-39        .
## 41-40        .
## 42-41        .
## 43-42        .
## 44-43        .
## 45-44        .
## 46-45        .
## 47-46        .
## 48-47        .
## 49-48        .
## 50-49        .
## 51-50        .
```

```
## 52-51        .
## 53-52        .
## 54-53        .
## 55-54        .
## 56-55        .
## 57-56        .
## 58-57        .
## 59-58        .
## 60-59        .
## 61-60        .
## 62-61        .
## 63-62        0.0207840734
## 64-63        0.0078828579
## 65-64        .
## 66-65        .
## 67-66        0.0015487146
## 68-67        .
## 69-68        .
## 70-69        .
## 71-70        .
## 72-71        .
## 73-72        .
## 74-73        .
## 75-74        .
## 76-75        .
## 77-76        .
## 78-77        .
## 79-78        .
## 80-79        .
## 81-80        .
## 82-81        .
## 83-82        .
## 84-83        .
## 85-84        .
## 86-85        .
## 87-86        .
## 88-87        .
## 89-88        .
## 90-89        .
## 91-90        .
## 92-91        .
## 93-92        .
## 94-93        .
## 95-94        .
## 96-95        .
## 97-96        .
## 98-97        .
## 99-98        .
## 100-99       .
## 101-100      .
## 102-101      .
## 103-102      .
## 104-103      .
## 105-104      .
```

```
## 106-105      .
## 107-106      .
## 108-107      .
## 109-108      .
## 110-109      .
## 111-110      .
## 112-111      .
## 113-112      .
## 114-113      .
## 115-114      .
## 116-115      .
## 117-116      .
## 118-117      .
## 119-118      .
## 120-119      .
## 121-120      .
## 122-121      .
## 123-122      .
## 124-123      .
## 125-124      .
## 126-125      .
## 127-126      .
## 128-127      .
## 129-128      .
## 130-129      .
## 131-130      .
## 132-131      .
## 133-132      .
## 134-133      .
## 135-134      .
## 136-135      .
## 137-136      .
## 138-137      .
## 139-138      .
## 140-139      .
## 141-140      .
## 142-141      .
## 143-142      .
## 144-143      .
## 145-144      .
## 146-145      .
## 147-146      .
## 148-147      .
## 149-148      .
## 150-149      .
## 151-150      .
## 152-151      .
## 153-152      .
## 154-153      .
## 155-154      .
## 156-155      .
## 157-156      .
## 158-157      .
## 159-158      .
```

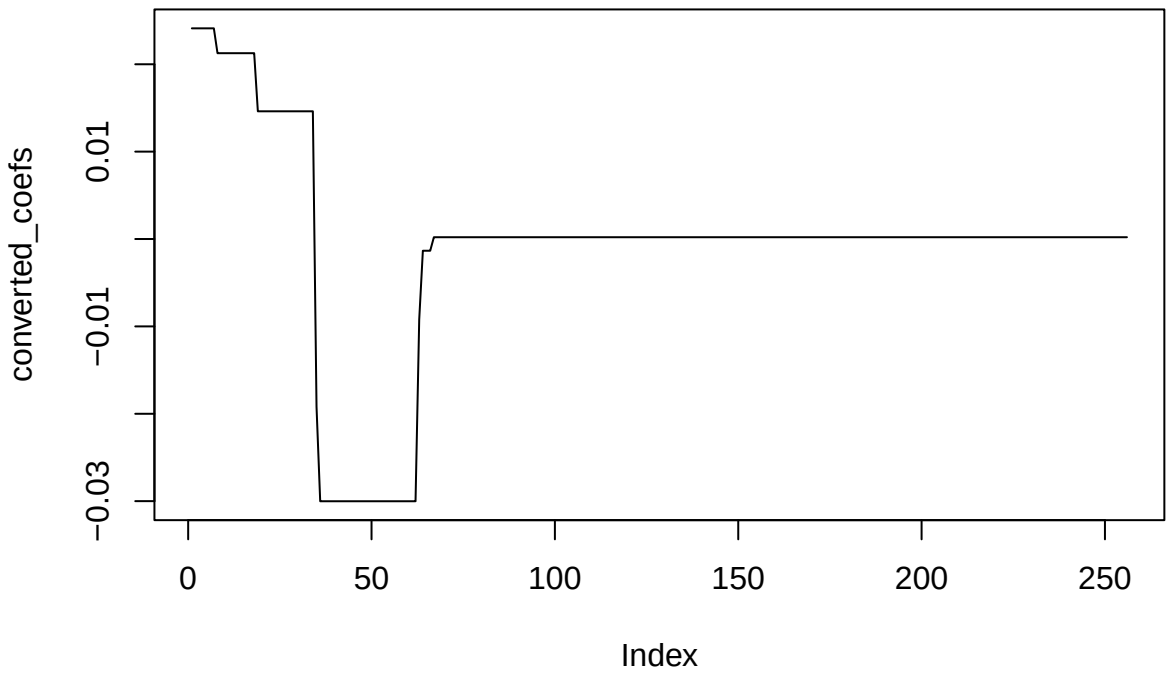
```
## 160-159      .
## 161-160      .
## 162-161      .
## 163-162      .
## 164-163      .
## 165-164      .
## 166-165      .
## 167-166      .
## 168-167      .
## 169-168      .
## 170-169      .
## 171-170      .
## 172-171      .
## 173-172      .
## 174-173      .
## 175-174      .
## 176-175      .
## 177-176      .
## 178-177      .
## 179-178      .
## 180-179      .
## 181-180      .
## 182-181      .
## 183-182      .
## 184-183      .
## 185-184      .
## 186-185      .
## 187-186      .
## 188-187      .
## 189-188      .
## 190-189      .
## 191-190      .
## 192-191      .
## 193-192      .
## 194-193      .
## 195-194      .
## 196-195      .
## 197-196      .
## 198-197      .
## 199-198      .
## 200-199      .
## 201-200      .
## 202-201      .
## 203-202      .
## 204-203      .
## 205-204      .
## 206-205      .
## 207-206      .
## 208-207      .
## 209-208      .
## 210-209      .
## 211-210      .
## 212-211      .
## 213-212      .
```

```
## 214-213      .
## 215-214      .
## 216-215      .
## 217-216      .
## 218-217      .
## 219-218      .
## 220-219      .
## 221-220      .
## 222-221      .
## 223-222      .
## 224-223      .
## 225-224      .
## 226-225      .
## 227-226      .
## 228-227      .
## 229-228      .
## 230-229      .
## 231-230      .
## 232-231      .
## 233-232      .
## 234-233      .
## 235-234      .
## 236-235      .
## 237-236      .
## 238-237      .
## 239-238      .
## 240-239      .
## 241-240      .
## 242-241      .
## 243-242      .
## 244-243      .
## 245-244      .
## 246-245      .
## 247-246      .
## 248-247      .
## 249-248      .
## 250-249      .
## 251-250      .
## 252-251      .
## 253-252      .
## 254-253      .
## 255-254      .
## 256-255      .
```

Når vi har kørt koden, vil vi se en liste over koefficientværdier for hver lambda-værdi. Hver koefficient repræsenterer koefficienten for den tilsvarende successive difference i den parametriserede model.

For at konvertere koefficienterne tilbage til de oprindelige β_i-værdier, kan vi multiplicere dem med designmatricen B og plotte resultaterne.

```
converted_coefs <- B %*% coef_values[-1]  # Multiplicer koefficienter med designmatricen, undladelse af
plot(converted_coefs, type = "l")
```



Når vi kører koden, vil vi få et plot, der viser de konverterede koefficientværdier som en funktion af faktornummeret (i). Plotet viser, hvordan koefficienterne ændrer sig som en "pæn" funktion af successive differenser.

Bemærk, at i dette eksempel er der brugt coef_values[-1] for at undlade den første koefficient (intercept), da denne værdi allerede er repræsenteret af konstantleddet 1 i designmatricen. Ved at udelade den første koefficient kan plottet visualisere ændringerne i de øvrige koefficienter som funktion af faktornummeret.

11)

For at udføre den tilsvarende konstruktion med L2-penalty (ridge-penalty) kan vi bruge den oprindelige designmatrix (ph2.train$X) i stedet for at anvende successive differenser. Dette vil resultere i en model, hvor koefficienterne ikke er parametriseret ved hjælp af successive differenser, men i stedet er direkte forbundet med de oprindelige variabler. Her er et eksempel på, hvordan vi kan gøre det:

```
CVridge <- cv.glmnet(ph2.train$X, ph2.train$g, family = "binomial", alpha = 0)
coef_values_ridge <- coef(CVridge)
coef_values_ridge
```

```
## 257 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  6.543932e+00
## x.1          2.740931e-03
## x.2          2.382118e-02
## x.3          1.931278e-02
```

```
## x.4         -2.614125e-02
## x.5         -7.625554e-04
## x.6          2.191512e-02
## x.7          7.423459e-03
## x.8         -9.530601e-03
## x.9         -9.479469e-03
## x.10         7.781921e-03
## x.11         1.516419e-02
## x.12         2.837893e-03
## x.13         6.697834e-04
## x.14         1.818886e-02
## x.15         2.077200e-02
## x.16         2.182500e-02
## x.17         1.865182e-02
## x.18         1.113798e-02
## x.19        -4.149444e-03
## x.20         4.590627e-03
## x.21         1.212418e-02
## x.22        -3.818072e-03
## x.23        -1.770387e-02
## x.24        -1.390998e-02
## x.25         5.831167e-03
## x.26         6.736536e-03
## x.27         8.756575e-03
## x.28         1.278208e-02
## x.29         1.891695e-02
## x.30         8.959556e-03
## x.31         2.010734e-02
## x.32         1.883800e-02
## x.33         5.866235e-03
## x.34         6.833112e-03
## x.35        -3.171618e-03
## x.36        -1.473186e-02
## x.37        -2.415449e-02
## x.38        -2.913252e-02
## x.39        -3.099537e-02
## x.40        -3.331767e-02
## x.41        -2.342220e-02
## x.42        -3.050583e-02
## x.43        -3.242767e-02
## x.44        -2.436313e-02
## x.45        -2.594529e-02
## x.46        -2.640739e-02
## x.47        -3.510637e-02
## x.48        -3.301922e-02
## x.49        -2.331668e-02
## x.50        -2.387395e-02
## x.51        -2.028028e-02
## x.52        -2.433555e-02
## x.53        -2.825084e-02
## x.54        -2.465294e-02
## x.55        -2.165491e-02
## x.56        -2.011891e-02
## x.57        -2.613607e-02
```

```
## x.58       -2.500294e-02
## x.59       -1.736898e-02
## x.60       -1.979536e-02
## x.61       -2.479840e-02
## x.62       -1.764655e-02
## x.63       -1.602303e-02
## x.64       -1.036676e-02
## x.65       -1.059425e-02
## x.66       -1.580388e-02
## x.67       -3.739128e-03
## x.68        3.519243e-03
## x.69       -6.291401e-03
## x.70        9.437344e-04
## x.71        7.684033e-03
## x.72        6.098549e-03
## x.73       -2.050168e-03
## x.74        3.398101e-03
## x.75        1.043302e-02
## x.76        3.329000e-03
## x.77        2.647749e-03
## x.78       -2.892223e-03
## x.79        1.166636e-03
## x.80        1.151614e-02
## x.81        8.525290e-03
## x.82       -2.565882e-03
## x.83       -1.602644e-04
## x.84        8.549903e-03
## x.85       -3.623292e-03
## x.86       -5.065450e-03
## x.87        3.213986e-03
## x.88       -4.441803e-03
## x.89       -3.290730e-03
## x.90        3.903989e-03
## x.91        3.071146e-03
## x.92       -1.160384e-02
## x.93       -1.053910e-02
## x.94        7.465678e-03
## x.95        1.128685e-03
## x.96       -2.727192e-03
## x.97       -1.336440e-03
## x.98        8.933214e-04
## x.99       -6.987881e-03
## x.100       6.357174e-04
## x.101       2.002855e-03
## x.102      -3.423751e-04
## x.103      -4.932073e-03
## x.104       7.841047e-03
## x.105      -1.425609e-03
## x.106      -1.126213e-03
## x.107       4.386874e-03
## x.108      -1.100374e-03
## x.109       2.328291e-03
## x.110       4.084567e-04
## x.111       2.019297e-03
```

```
## x.112       -3.280575e-03
## x.113       -5.596098e-03
## x.114        7.349456e-03
## x.115        4.665324e-03
## x.116       -9.876975e-04
## x.117       -1.795581e-03
## x.118        4.380255e-03
## x.119       -2.112508e-03
## x.120        1.551091e-03
## x.121       -4.924074e-03
## x.122        1.421004e-03
## x.123        2.789181e-03
## x.124        1.385862e-03
## x.125        7.468999e-03
## x.126       -3.333380e-03
## x.127        6.419469e-03
## x.128        3.678163e-03
## x.129        4.554871e-03
## x.130        5.981637e-03
## x.131        3.665756e-04
## x.132        1.475730e-03
## x.133       -1.799534e-03
## x.134        9.392817e-04
## x.135       -1.347335e-03
## x.136        3.415400e-03
## x.137        1.243493e-03
## x.138        1.613013e-03
## x.139       -1.301226e-03
## x.140        2.592058e-03
## x.141       -9.364435e-03
## x.142       -4.640969e-04
## x.143        5.928321e-03
## x.144        2.771734e-03
## x.145       -8.412035e-03
## x.146       -1.652388e-03
## x.147        7.738537e-03
## x.148        5.493846e-03
## x.149        3.872995e-03
## x.150        4.078875e-03
## x.151       -1.078026e-03
## x.152       -7.711679e-04
## x.153        9.539535e-04
## x.154        8.533094e-03
## x.155        1.061228e-03
## x.156        4.331343e-03
## x.157        1.185402e-02
## x.158        1.117535e-02
## x.159        9.572271e-03
## x.160        5.929314e-03
## x.161        3.616596e-03
## x.162        8.099023e-04
## x.163        2.873316e-03
## x.164        3.896951e-03
## x.165        5.624686e-04
```

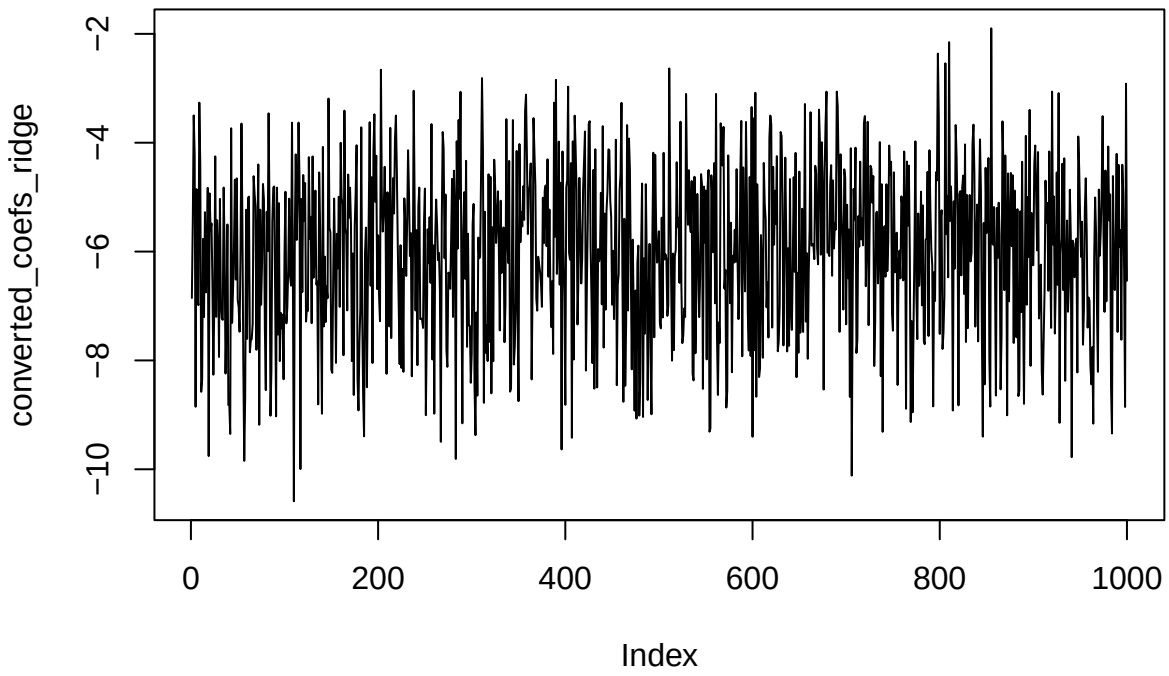
```
## x.166        1.120608e-03
## x.167       -9.464393e-03
## x.168       -8.189731e-04
## x.169        4.145061e-03
## x.170       -5.140169e-04
## x.171       -1.544889e-03
## x.172       -3.971690e-03
## x.173       -4.261880e-03
## x.174        2.758560e-03
## x.175       -2.400144e-03
## x.176        2.722958e-03
## x.177        2.738644e-04
## x.178       -3.401406e-03
## x.179        3.795219e-03
## x.180       -2.615497e-04
## x.181       -3.869516e-03
## x.182        2.621239e-03
## x.183       -4.117983e-03
## x.184       -7.376661e-03
## x.185        7.638761e-03
## x.186        5.349828e-03
## x.187       -8.655146e-03
## x.188        1.167807e-03
## x.189       -4.080964e-03
## x.190        1.006629e-02
## x.191        4.880172e-03
## x.192        4.759539e-03
## x.193       -4.513845e-03
## x.194        5.653077e-03
## x.195        4.918910e-03
## x.196        1.014392e-03
## x.197        8.604020e-03
## x.198       -3.402845e-03
## x.199        2.203936e-03
## x.200        1.333296e-03
## x.201       -2.620371e-03
## x.202       -9.083254e-03
## x.203        2.629995e-03
## x.204       -7.795469e-03
## x.205        4.983153e-04
## x.206       -8.891944e-03
## x.207       -4.074864e-03
## x.208       -2.265333e-03
## x.209       -1.442379e-03
## x.210       -2.369515e-03
## x.211        1.002054e-04
## x.212        1.286722e-03
## x.213       -4.130775e-04
## x.214        3.074931e-03
## x.215        7.854722e-03
## x.216        1.283885e-02
## x.217        8.857493e-04
## x.218        4.788914e-03
## x.219        6.629622e-03
```

```
## x.220        4.374328e-03
## x.221        6.130950e-03
## x.222       -2.938643e-03
## x.223       -4.945381e-03
## x.224       -7.151659e-03
## x.225       -4.851612e-03
## x.226       -2.252198e-03
## x.227       -1.614977e-02
## x.228       -7.865568e-03
## x.229        9.552640e-03
## x.230        2.871087e-04
## x.231       -2.028535e-02
## x.232       -6.107827e-04
## x.233       -3.158261e-04
## x.234        4.740759e-03
## x.235       -3.403186e-03
## x.236       -9.149732e-04
## x.237        4.484917e-03
## x.238       -6.041954e-03
## x.239       -1.639071e-03
## x.240       -5.350754e-03
## x.241        8.064663e-05
## x.242       -6.187145e-03
## x.243       -3.177525e-03
## x.244       -5.405796e-03
## x.245       -5.860030e-03
## x.246       -8.308311e-03
## x.247       -4.556613e-03
## x.248       -3.827283e-03
## x.249       -2.384729e-03
## x.250       -1.027359e-02
## x.251       -6.644008e-03
## x.252        4.740614e-03
## x.253        5.747192e-03
## x.254       -1.791917e-03
## x.255       -3.574901e-03
## x.256       -9.104757e-04
```

Ved at køre koden vil vi få koefficienterne for hver lambda-værdi for glmnet-modellen med L2-penalty. I modsætning til modellen med successive differenser vil alle koefficienter være til stede i resultatet, herunder interceptet (den første koefficient).

For at visualisere de oprindelige koefficienter som en funktion af variabelindekset (i), kan vi bruge designmatricen (ph2.train$X) og de konverterede koefficienter. Her er et eksempel på, hvordan vi kan gøre det:

```
converted_coefs_ridge <- ph2.train$X %*% coef_values_ridge[-1]  # Multiplicer koefficienter med designm
plot(converted_coefs_ridge, type = "l")
```



Når vi kører koden, vil vi få et plot, der viser de konverterede koefficientværdier (uden intercept) som en funktion af variabelindekset. Dette plot vil illustrere, hvordan koefficienterne ændrer sig i den model, der er opnået ved anvendelse af L2-penalty.

Forskellen mellem L1-penalty (Lasso) og L2-penalty (ridge) er, at L1-penalty har tendens til at skabe en mere sparsom model ved at sætte mange koefficienter til nøjagtigt nul, hvilket giver en form for variabeludvælgelse. På den anden side reducerer L2-penalty koefficientværdierne, men eliminerer dem normalt ikke helt. Dette betyder, at L2-penalty bevarer alle variabler, men med mindre betydning.

Plotet med L2-penalty kan vise en mere jævn ændring i koefficienterne i forhold til variabelindekset sammenlignet med L1-penalty, hvor der er sprang i koefficientværdierne på grund af nulning af variabler. Derudover kan L2-penalty føre til mere stabil og mindre følsomme koefficienter sammenlignet med L1-penalty. Vi ser også at der virker til at være mere "støj" i plottet, da alle β-værdierne bevares.