

# Robust Generalized Linear Models

Zhu Wang\*

December 22, 2022

The CC-family contains functions of composite of concave and convex functions. The CC-estimators are derived from minimizing loss functions in the CC-family by the iteratively reweighted convex optimization (IRCO), an extension of the iteratively reweighted least squares (IRLS). The IRCO reduces the weight of the observation that leads to a large loss; it also provides weights to help identify outliers. In the applications of robust (penalized) generalized linear models, the IRCO becomes the iteratively reweighted GLM or IRGLM. See Wang (2020).

## Robust logistic regression

In a UK hospital, 135 expectant mothers were surveyed on the decision of breastfeeding their babies or not, along with two-level predictive factors. Description and references can be found in Heritier et al. (2009).

```
require("mpath")
data(breastfeed)
```

Remove rows with missing values.

```
breastfeed=na.omit(breastfeed)
```

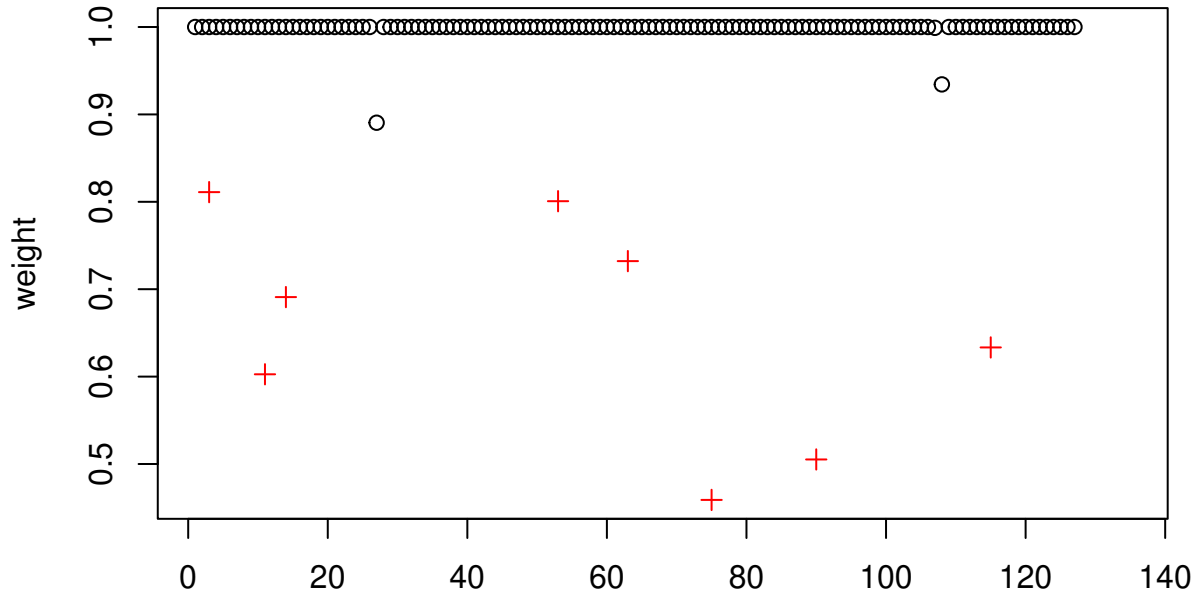
We compute binomial-induced CC-estimators, i.e., robust logistic regression, and display the robust weights for each model.

```
sval <- c(1.5, 1.5, 5, 2.5, 3.5, 2.5, 2.2, 7)
cfun <- c("hcave", "acave", "bcave", "ccave", "dcave", "gcave", "tcave", "ecave")
id <- 1:8
for(i in c(1:5,8,6,7)){
  fitnew <- irglm(breast~., data=breastfeed, s=sval[i], cfun=i, dfun=binomial(),
                 trace=FALSE)
  goodid <- sort.list(fitnew$weights_update)[id]
  plot(fitnew$weights_update, type="n", ylab="weight",
       main = eval(substitute(expression(paste(cfun, "(", sigma, "=", s, ")")),
                     list(cfun=cfun[i], s = sval[i]))))
  points(fitnew$weights_update[-goodid], ylab="weight",
         main = eval(substitute(expression(paste(cfun, "(", sigma, "=", s, ")")),
                       list(cfun=cfun[i], s = sval[i]))))
  points(sort.list(fitnew$weights_update)[id], sort(fitnew$weights_update)[id], pch=3,
         col="red")
}
```

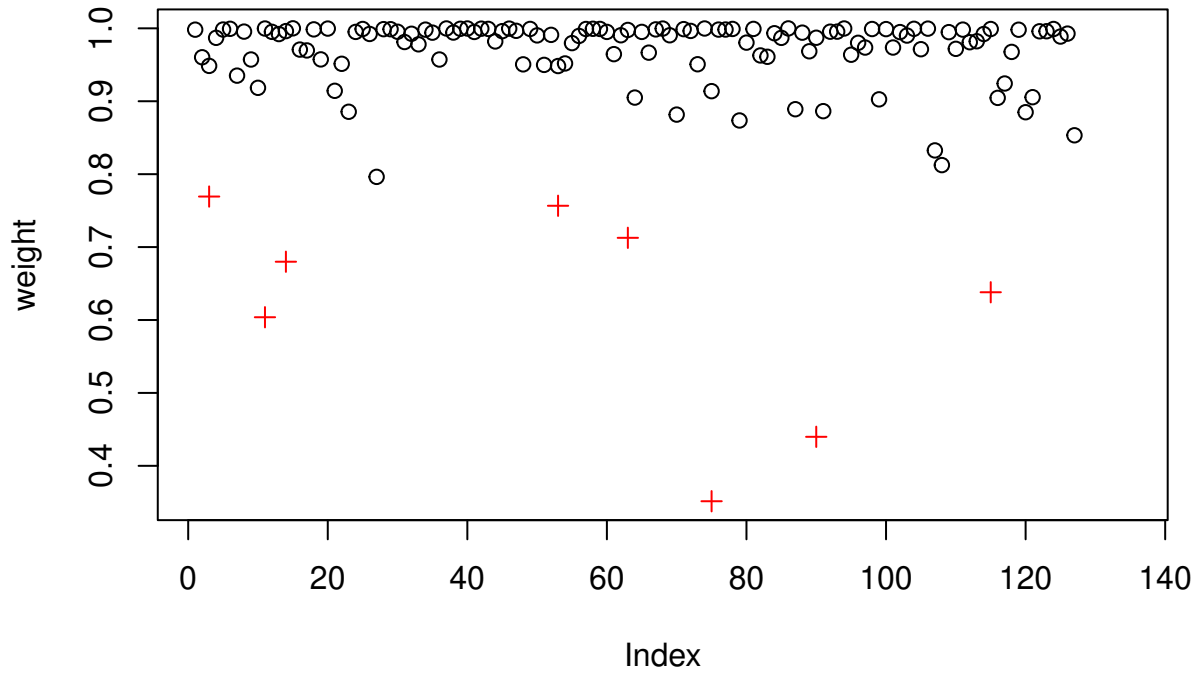
---

\*University of Tennessee Health Science Center, zwang145@uthsc.edu

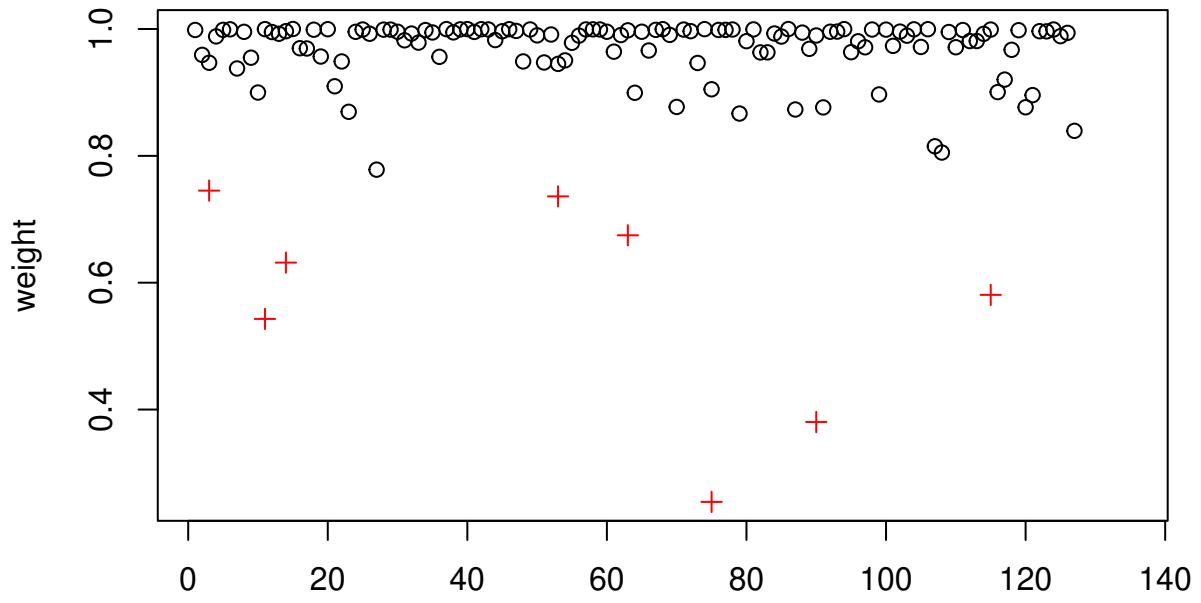
hcave( $\sigma=1.5$ )



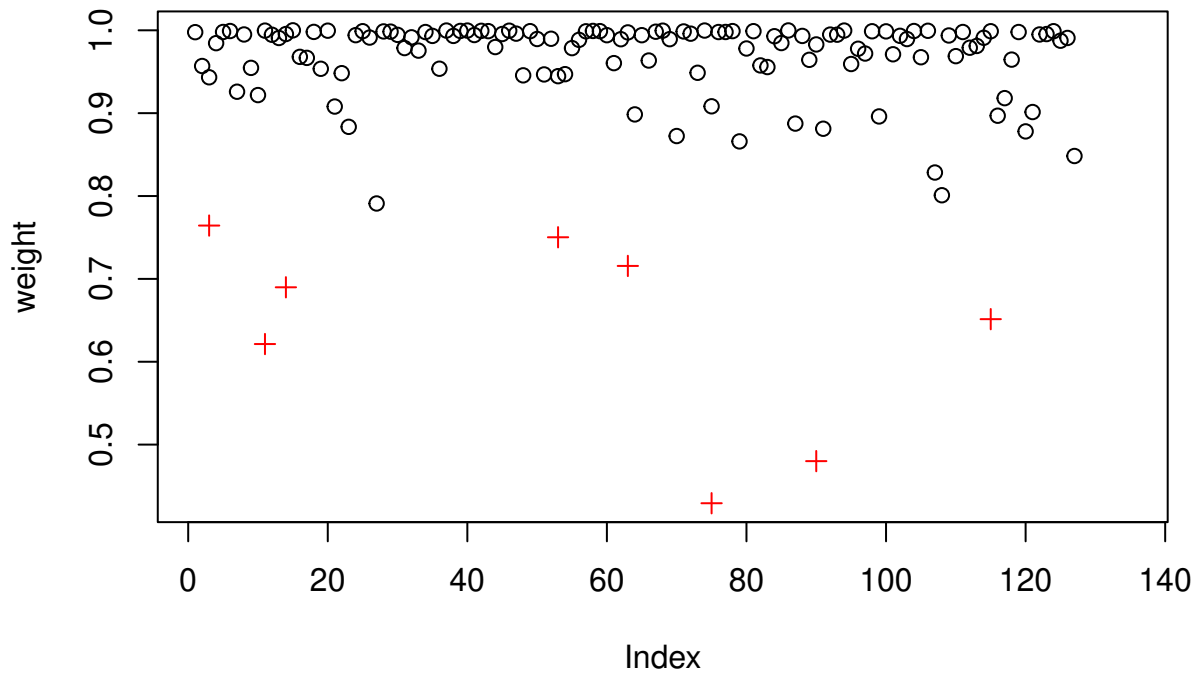
Index  
acave( $\sigma=1.5$ )



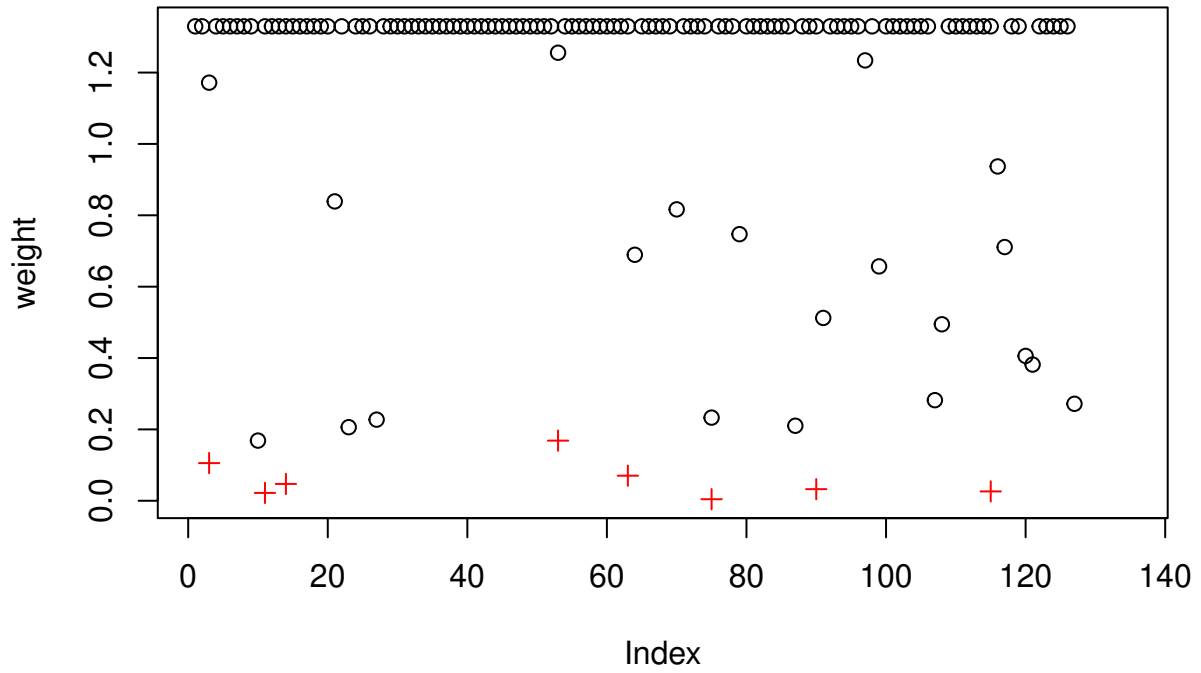
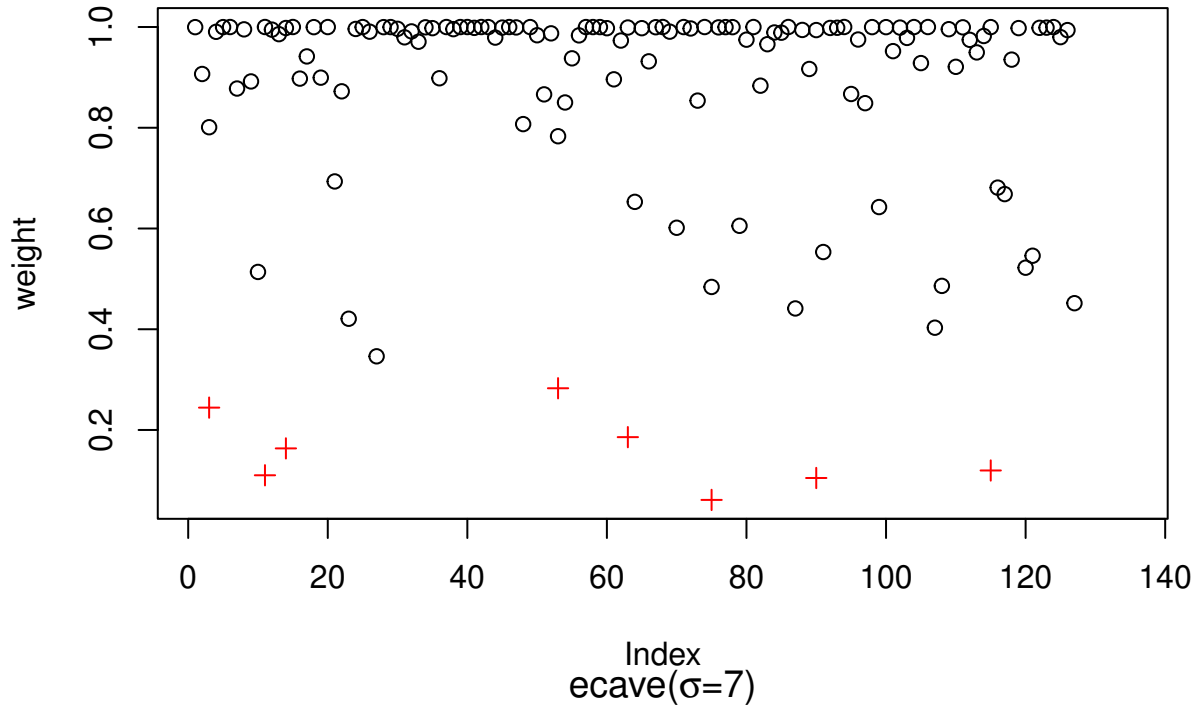
bcave( $\sigma=5$ )



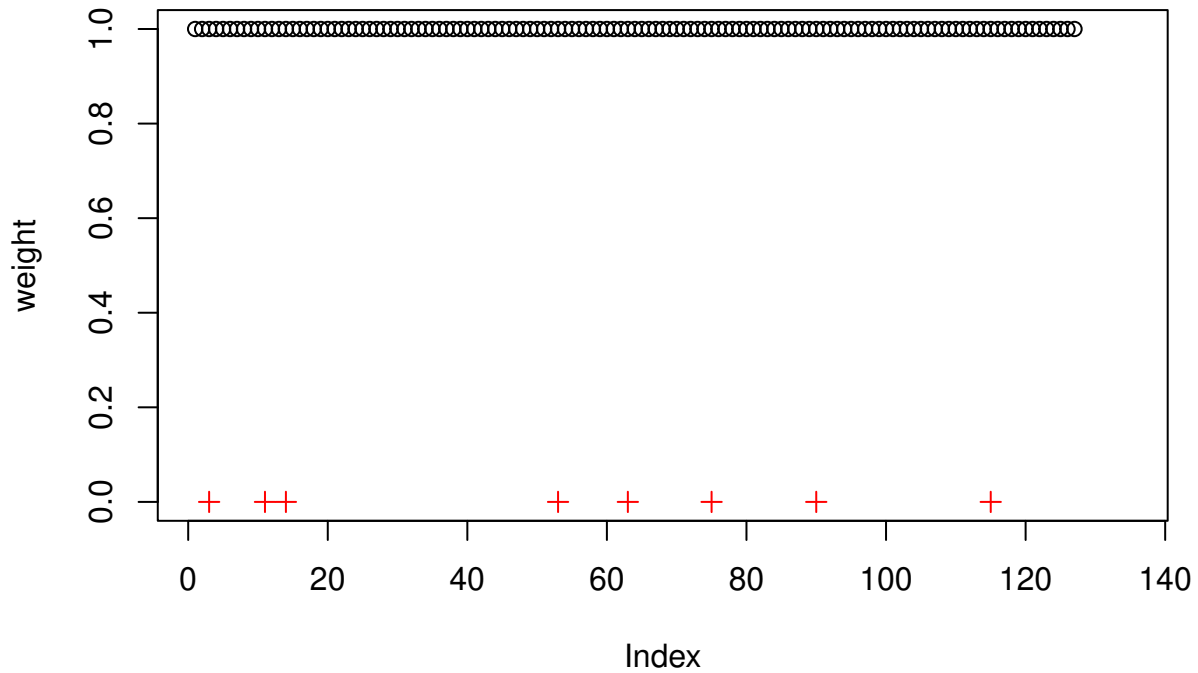
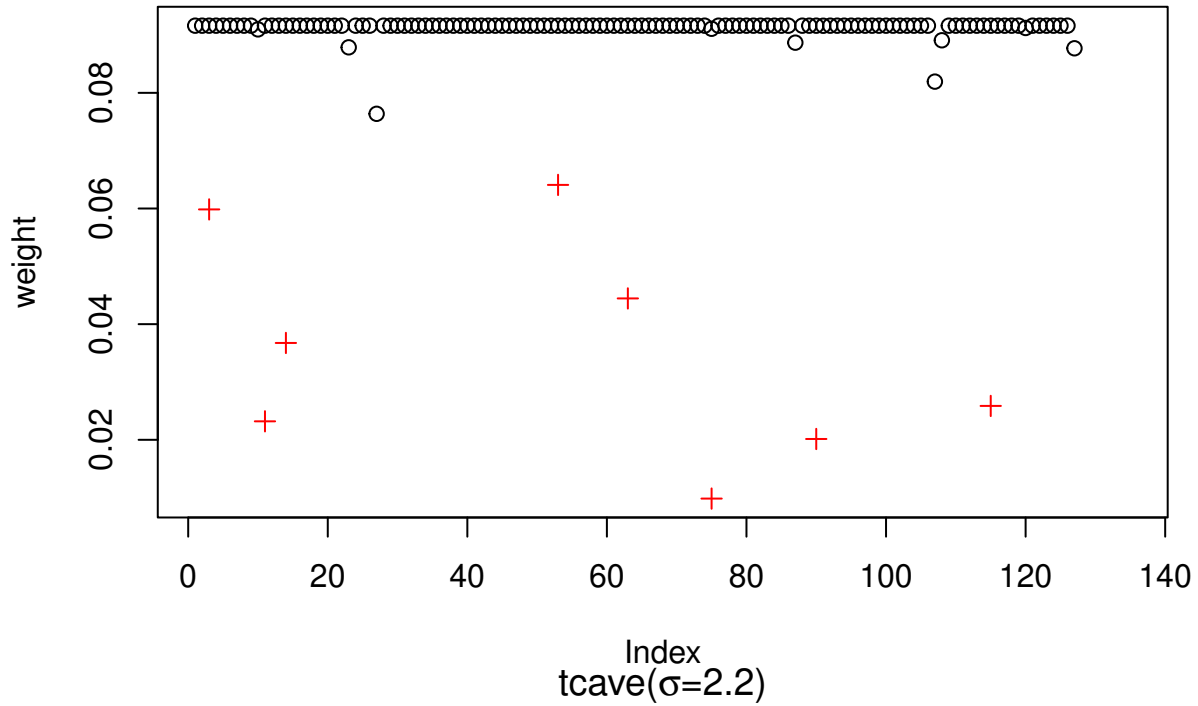
Index  
ccave( $\sigma=2.5$ )



dcave( $\sigma=3.5$ )



gcave( $\sigma=2.5$ )



Despite large estimated probability  $\geq 0.8$  of trying to breastfeed or not in a logistic regression, these individuals took the opposite decisions.

```
fit.glm <- glm(as.integer(breast)-1~., data=breastfeed, family=binomial())
id <- c(3, 11, 14, 53,63, 75,90, 115)
pred <- predict(fit.glm, type="response") ### predicted probabilities
```

```
cbind(breastfeed, pred)[id,]
```

```
##      breast pregnancy howfed howfedfr partner smokenow smokebf age educat
## 3   Bottle Beginning Breast  Breast Partner      No      No 39    16
## 12  Bottle Beginning Breast  Breast Single      No      No 29    18
## 15  Bottle Beginning Bottle  Breast Partner      No      No 33    21
## 56  Bottle      End Bottle  Breast Partner      No      No 25    16
## 66  Breast      End Bottle  Bottle Partner      Yes     Yes 27    16
## 78  Bottle Beginning Breast  Bottle Partner      No      No 28    28
## 93  Breast Beginning Bottle  Bottle Single      Yes     Yes 19    16
## 118 Bottle Beginning Breast  Breast Single      No      No 20    18
##      ethnic      pred
## 3      White 0.77424035
## 12 Non-white 0.89871707
## 15      White 0.83665383
## 56      White 0.82204044
## 66      White 0.18544970
## 78 Non-white 0.97026037
## 93      White 0.02275785
## 118 Non-white 0.87454933
```

For variable selection, we develop a usual penalized LASSO logistic regression, where the optimal penalty parameter  $\lambda$  is chosen by a 10-fold cross-validation.

```
n.cores <- 2
set.seed(195)
fitcv.glm <- cv.glmreg(as.integer(breast)-1~., data=breastfeed, penalty="enet",
                      family="binomial", type="loss", plot.it=FALSE, parallel=TRUE,
                      n.cores=n.cores, standardize=TRUE)
fit <- fitcv.glm$fit
```

The smallest CV value from penalized logistic regression

```
min(fitcv.glm$cv)
```

```
## [1] -7.815975
```

Penalized logistic regression with penalty LASSO

```
coef(fit)[,fitcv.glm$lambda.which]
```

```
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##      -2.492768348      -0.552721156      0.257557827      1.220755264
##      partnerPartner      smokenowYes      smokebfYes      age
##      0.862448995      -2.252414341      0.623626979      0.002488979
##      educat      ethnicNon-white
##      0.119557901      1.541320581
```

Compute the SCAD logistic regression, where the optimal penalty parameter  $\lambda$  is chosen by a 10-fold cross-validation. The SCAD logistic regression is more sparse than the LASSO estimator.

```
set.seed(195)
fitcv.glm <- cv.glmreg(as.integer(breast)-1~., data=breastfeed, penalty="snet",
                      family="binomial", type="loss", plot.it=FALSE, parallel=TRUE,
                      n.cores=n.cores, standardize=TRUE)
fit <- fitcv.glm$fit
```

The smallest CV value from penalized logistic regression

```
min(fitcv.glm$cv)
```

```
## [1] -7.815975
```

Penalized logistic regression with penalty SCAD

```
coef(fit)[,fitcv.glm$lambda.which]
```

```
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##      0.09874844      0.00000000      0.00000000      1.04702265
##      partnerPartner      smokenowYes      smokebfYes      age
##      0.47532632      -2.00125709      0.00000000      0.00000000
##      educat      ethnicNon-white
##      0.00000000      1.94414156
```

The  $\lambda$  value in SCAD is then utilized to compute binomial-induced SCAD CC-estimators for various concave components.

```
for(i in c(1:5,8,6,7)){
  cat("\ncfun=", cfun[i], "\n")
  fit.irglmreg <- irglmreg(breast~., data=breastfeed, s=sval[i], cfun=i, penalty="snet",
                          lambda=fitcv.glm$lambda.optim, dfun=binomial(), parallel=FALSE,
                          type.path="nonactive", standardize=TRUE)
  print(coef(fit.irglmreg))
}
```

```
##
## cfun= hcave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##      -0.20262257      0.00000000      0.00000000      1.41623162
##      partnerPartner      smokenowYes      smokebfYes      age
##      0.24121875      -2.31220066      0.00000000      0.00000000
##      educat      ethnicNon-white
##      0.02524874      2.48775264
##
```

```
## cfun= acave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##      0.323505087      0.000000000      0.000000000      1.194139787
##      partnerPartner      smokenowYes      smokebfYes      age
##      0.197984927      -2.383687907      0.000000000      0.000000000
##      educat      ethnicNon-white
##      0.008521202      2.523000073
##
```

```
## cfun= bcave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##      0.32721541      0.00000000      0.00000000      1.20998505
##      partnerPartner      smokenowYes      smokebfYes      age
##      0.12916071      -2.44460015      0.00000000      0.00000000
##      educat      ethnicNon-white
##      0.01285265      2.63969594
##
```

```
## cfun= ccave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##      0.354230145      0.000000000      0.000000000      1.177822530
##      partnerPartner      smokenowYes      smokebfYes      age
##      0.224698662      -2.376636566      0.000000000      0.000000000
```

```

##          educat      ethnicNon-white
##    0.006084647      2.483034951
##
## cfun= dcave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##    2.71145780      0.00000000      0.12034058      0.02718298
##    partnerPartner      smokenowYes      smokebfYes      age
##    0.00000000      -3.89297984      0.00000000      0.00000000
##          educat      ethnicNon-white
##    0.00000000      1.15836471
##
## cfun= ecave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##    3.26756873      0.00000000      0.00000000      0.05330153
##    partnerPartner      smokenowYes      smokebfYes      age
##    0.00000000      -4.24833062      0.00000000      0.00000000
##          educat      ethnicNon-white
##    0.00000000      2.44864324
##
## cfun= gcave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##    -0.70232531      0.00000000      0.00000000      1.75532959
##    partnerPartner      smokenowYes      smokebfYes      age
##    0.00000000      -2.68637620      0.00000000      0.00000000
##          educat      ethnicNon-white
##    0.06456509      3.24729928
##
## cfun= tcave
##      (Intercept) pregnancyBeginning      howfedBreast      howfedfrBreast
##    -2.2679907      0.00000000      0.00000000      1.2678561
##    partnerPartner      smokenowYes      smokebfYes      age
##    0.00000000      -2.4827295      0.00000000      0.00000000
##          educat      ethnicNon-white
##    0.1648061      3.5883907

```

## Robust Poisson regression

A cohort of 3066 Americans over the age of 50 were studied on health care utilization, doctor office visits Heritier et al. (2009). The survey also contained 24 predictors in demographic, health needs and economic access. We compute Poisson-induced CC-estimators, i.e., robust Poisson regressions. The seven smallest weights occur to the subjects with 200, 208, 224, 260, 300, 365 and 750 doctor visits in two years.

```

data(docvisits)
sval <- c(10, 10, 45, 20, 5, 5, 280, 200)
cfun <- c("hcave", "acave", "bcave", "ccave", "dcave", "gcave", "tcave", "ecave")
id <- 1:7
for(i in c(1:5,8,6,7)){
  fitnew <- irglm(visits~age+factor(gender)+factor(race)+factor(hispan)
    +factor(marital)+factor(arthri)+factor(cancer)
    +factor(hipress)+factor(diabet)+factor(lung)+factor(hearth)
    +factor(stroke)+factor(psych)+factor(iadla)+factor(adlwa)
    +edyears+feduc+meduc+log(income+1)+factor(insur),
    data=docvisits,cfun=i,s=sval[i],dfun=poisson(),trace=FALSE)
}

```

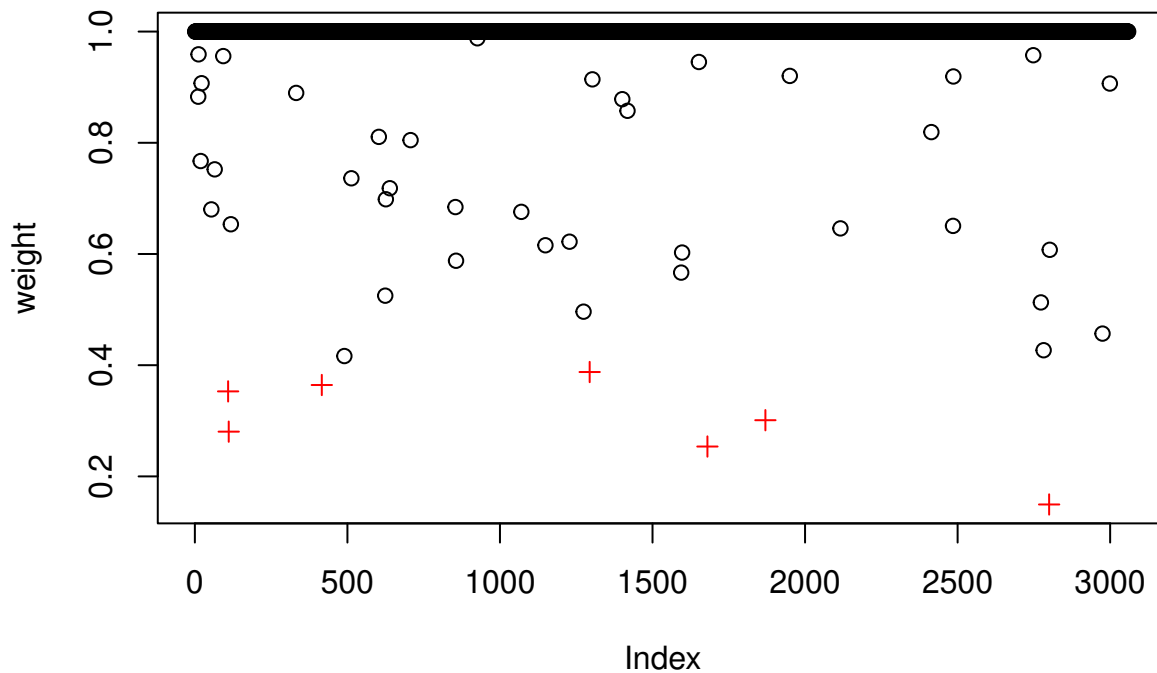


```

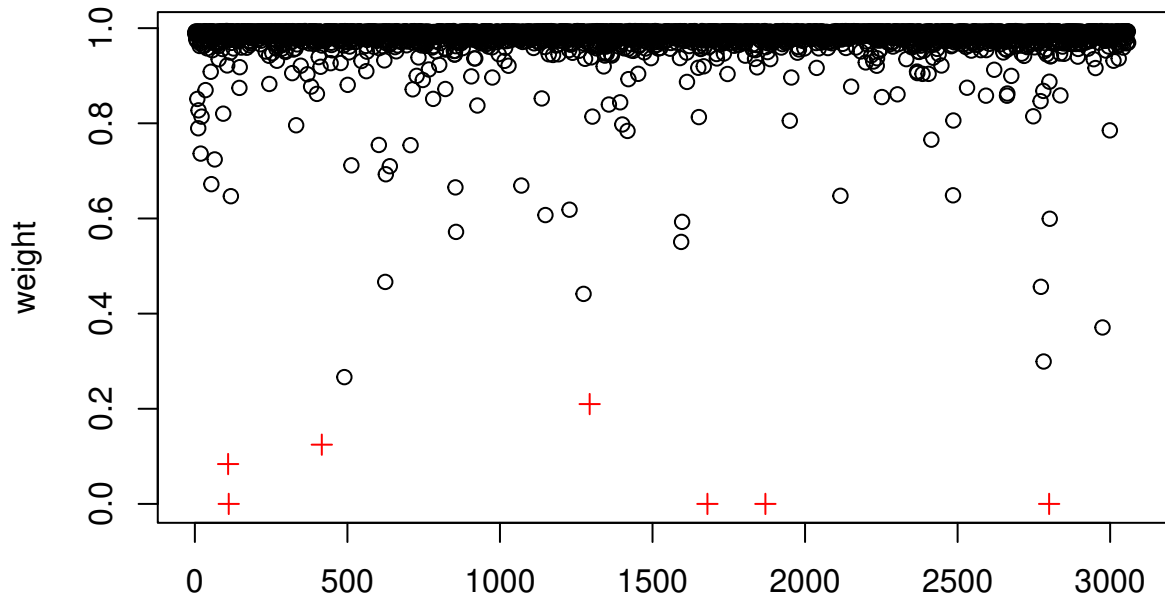
goodid <- sort.list(fitnew$weights_update)[id]
plot(fitnew$weights_update, type="n", ylab="weight",
     main = eval(substitute(expression(paste(cfun, "(", sigma, "=", s, ")"))),
                 list(cfun=cfun[i], s = sval[i])))
points(fitnew$weights_update[-goodid], ylab="weight",
       main = eval(substitute(expression(paste(cfun, "(", sigma, "=", s, ")"))),
                   list(cfun=cfun[i], s = sval[i])))
if(i > 4){
  ### deal with overlapped points: obs 109, 111
  x <- sort.list(fitnew$weights_update)[id]
  y <- sort(fitnew$weights_update)[id]
  xnew <- sort(x)
  ynew <- y[sort.list(x)]
  points(xnew[1]-10, ynew[1], pch=3, col="red")
  points(xnew[2]+10, ynew[2], pch=3, col="red")
  points(xnew[3:7], ynew[3:7], pch=3, col="red")
}
else points(sort.list(fitnew$weights_update)[id], sort(fitnew$weights_update)[id],
           pch=3, col="red")
}

```

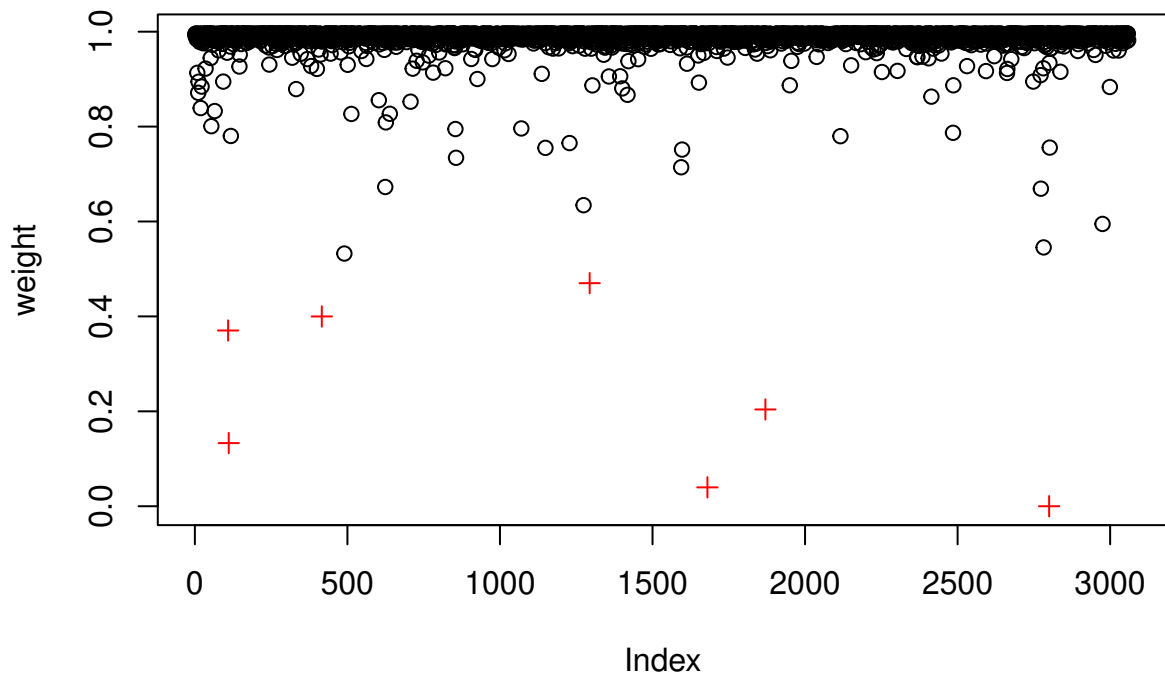
hcave( $\sigma=10$ )



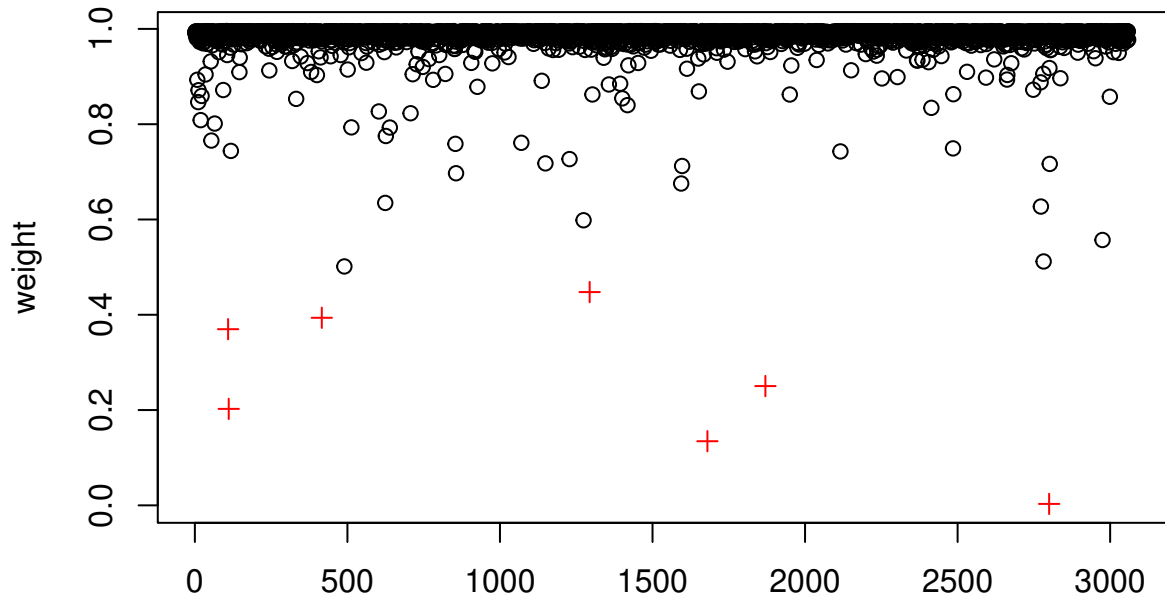
acave( $\sigma=10$ )



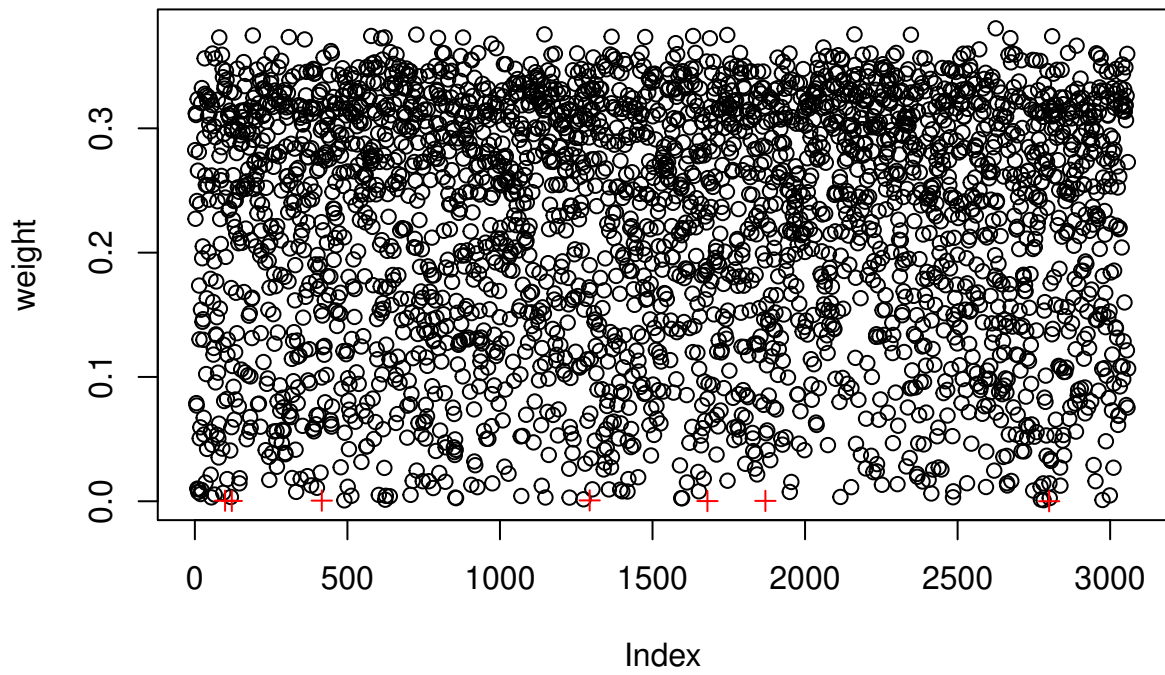
bcave( $\sigma=45$ )



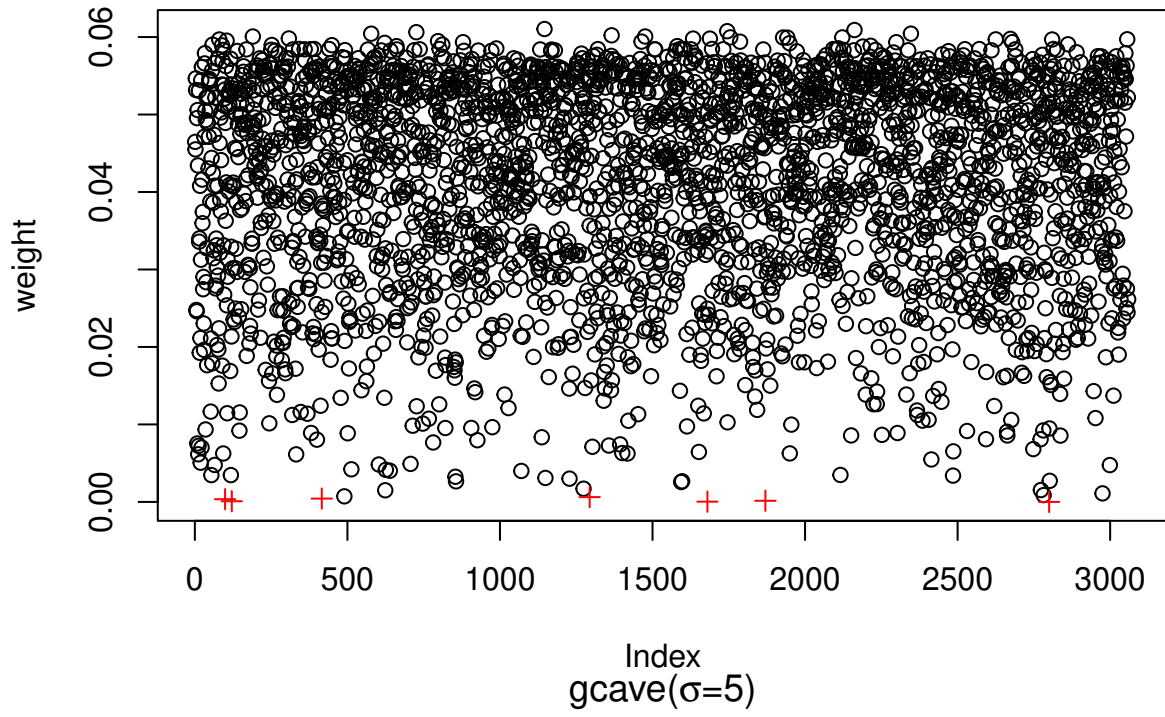
ccave( $\sigma=20$ )



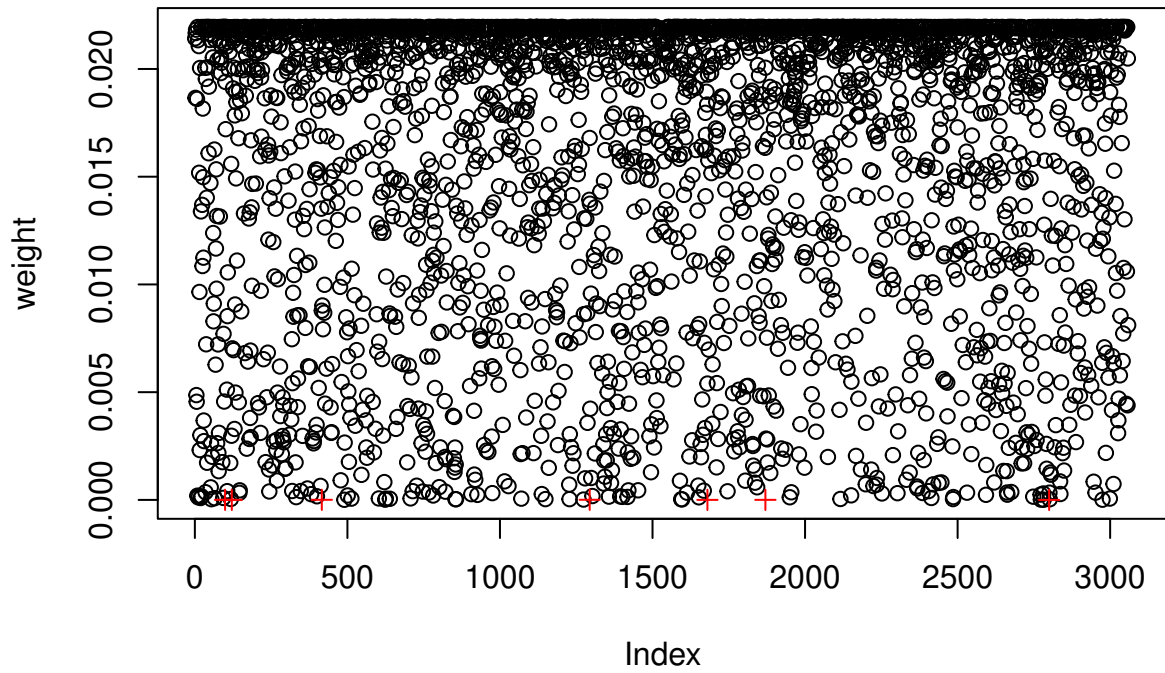
Index  
dcave( $\sigma=5$ )



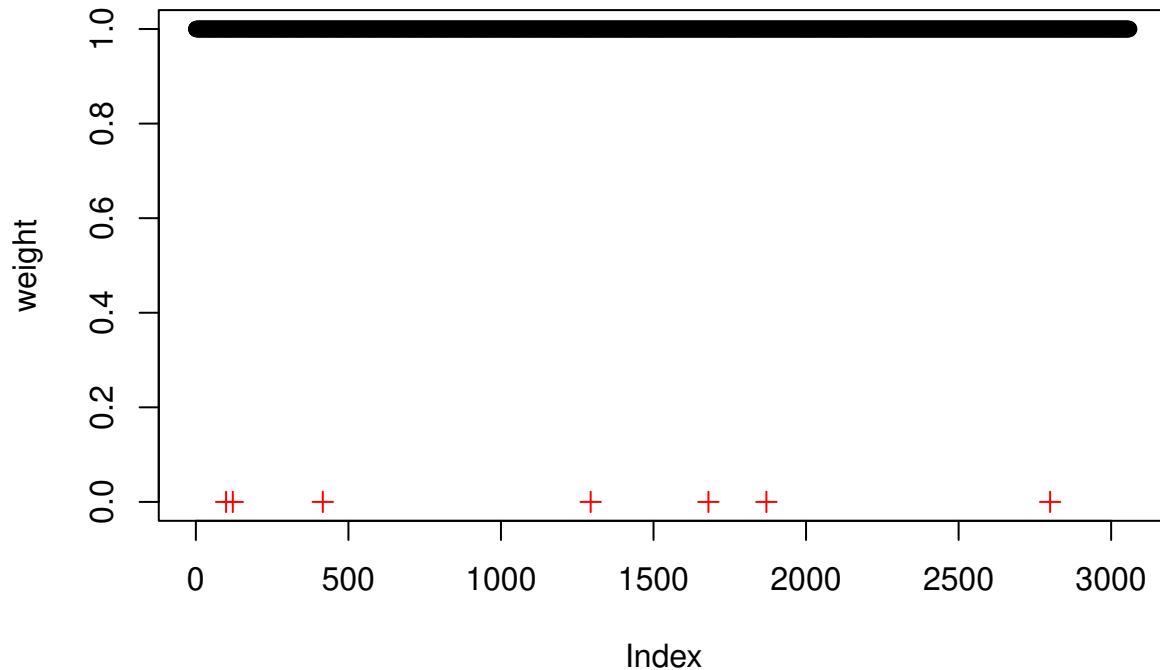
ecave( $\sigma=200$ )



gcave( $\sigma=5$ )



tcave( $\sigma=280$ )



Outliers of office visits

```
newid <- sort(sort.list(fitnew$weights_update)[id])
docvisits$visits[newid]
```

```
## [1] 224 300 208 200 365 260 750
```

Penalized Poisson regression with LASSO penalty. The tuning parameter  $\lambda$  value is chosen by cross-validation.

```
set.seed(195)
fitcv.glm <- cv.glmreg(visits~age+factor(gender)+factor(race)+factor(hispan)
  +factor(marital)+factor(arthri)+factor(cancer)
  +factor(hipress)+factor(diabet)+factor(lung)+factor(hearth)
  +factor(stroke)+factor(psych)+factor(iadla)+factor(adlwa)
  +edyears+feduc+meduc+log(income+1)+factor(insur),
  data=docvisits,family="poisson", penalty="enet", type="loss",
  plot.it=FALSE, parallel=TRUE, n.cores=n.cores, standardize=TRUE)
fit <- fitcv.glm$fit
```

The smallest CV value from penalized Poisson regression

```
min(fitcv.glm$cv)
```

```
## [1] -2894.529
```

Penalized Poisson regression with penalty LASSO

```
coef(fit)[,fitcv.glm$lambda.which]
```

```
##      (Intercept)          age factor(gender)1  factor(race)1
##      1.958525362    -0.006577549    0.058079225    -0.016603770
## factor(hispan)1 factor(marital)1 factor(arthri)1 factor(cancer)1
##      0.005810267    0.000000000    0.087497840    0.115530929
## factor(hipress)1 factor(diabet)1  factor(lung)1  factor(hearth)1
```

```
##      0.139162108      0.272014643      0.071727840      0.264918089
## factor(stroke)1 factor(psych)1 factor(iadla)1 factor(iadla)2
##      0.069391756      0.220222838      0.068125763      0.043447857
## factor(iadla)3 factor(adlwa)1 factor(adlwa)2 factor(adlwa)3
##      0.068932457      0.312374326      0.616403088      0.545308893
##      edyears      feduc      meduc log(income + 1)
##      0.009212421      -0.023840699      0.000000000      0.058544656
## factor(insur)1
##      0.072032891
```

Penalized Poisson regression with SCAD penalty. The tuning parameter  $\lambda$  value is chosen by cross-validation.

```
set.seed(195)
fitcv.glm <- cv.glmreg(visits~age+factor(gender)+factor(race)+factor(hispan)
+factor(marital)+factor(arthri)+factor(cancer)
+factor(hipress)+factor(diabet)+factor(lung)+factor(hearth)
+factor(stroke)+factor(psych)+factor(iadla)+factor(adlwa)
+edyears+feduc+meduc+log(income+1)+factor(insur),
data=docvisits, family="poisson", penalty="snet", type="loss",
plot.it=FALSE, parallel=TRUE, n.cores=n.cores, standardize=TRUE)
fit <- fitcv.glm$fit
```

The smallest CV value from penalized Poisson regression

```
min(fitcv.glm$cv)
```

```
## [1] -2894.529
```

Penalized Poisson regression with penalty SCAD

```
coef(fit)[,fitcv.glm$lambda.which]
```

```
##      (Intercept)      age factor(gender)1 factor(race)1
##      1.858069526      -0.003786934      0.000000000      0.000000000
## factor(hispan)1 factor(marital)1 factor(arthri)1 factor(cancer)1
##      0.000000000      0.000000000      0.029059308      0.067717808
## factor(hipress)1 factor(diabet)1 factor(lung)1 factor(hearth)1
##      0.120234297      0.296950385      0.000000000      0.291999839
## factor(stroke)1 factor(psych)1 factor(iadla)1 factor(iadla)2
##      0.001233415      0.252816298      0.000000000      0.000000000
## factor(iadla)3 factor(adlwa)1 factor(adlwa)2 factor(adlwa)3
##      0.000000000      0.368482223      0.681643963      0.641110056
##      edyears      feduc      meduc log(income + 1)
##      0.004754850      0.000000000      0.000000000      0.040644933
## factor(insur)1
##      0.017165257
```

The  $\lambda$  value in SCAD is then utilized to compute robust Poisson SCAD CC-estimators for various concave components.

```
for(i in c(1:5,8,6,7)){
  cat("\nCFUN=", cfun[i], "\n")
  fit.irglmreg <- irglmreg(visits~age+factor(gender)+factor(race)+factor(hispan)
+factor(marital)+factor(arthri)+factor(cancer)
+factor(hipress)+factor(diabet)+factor(lung)+factor(hearth)
+factor(stroke)+factor(psych)+factor(iadla)+factor(adlwa)
+edyears+feduc+meduc+log(income+1)+factor(insur),
data=docvisits, s=sval[i], cfun=i, penalty="snet",
```

```

lambda=fitcv.glm$lambda.optim, dfun=poisson(), parallel=FALSE,
type.path="nonactive", standardize=TRUE)
print(coef(fit.irglmreg))
}

```

```

##
## cfun= hcave
##      (Intercept)          age  factor(gender)1  factor(race)1
##      1.98635667      0.00000000      0.00000000      0.00000000
##  factor(hispan)1 factor(marital)1  factor(arthri)1  factor(cancer)1
##      0.00000000      0.00000000      0.03808422      0.03052763
##  factor(hipress)1  factor(diabet)1  factor(lung)1  factor(hearth)1
##      0.10857169      0.22472945      0.01353496      0.32244446
##  factor(stroke)1  factor(psych)1  factor(iadla)1  factor(iadla)2
##      0.04586816      0.26510976      0.00000000      0.00000000
##  factor(iadla)3  factor(adlwa)1  factor(adlwa)2  factor(adlwa)3
##      0.00000000      0.25088902      0.43649856      0.53724280
##      edyears          feduc          meduc  log(income + 1)
##      0.00000000      0.00000000      0.00000000      0.00000000
##  factor(insur)1
##      0.00000000
##
## cfun= acave
##      (Intercept)          age  factor(gender)1  factor(race)1
##      1.98189657      0.00000000      0.00000000      0.00000000
##  factor(hispan)1 factor(marital)1  factor(arthri)1  factor(cancer)1
##      0.00000000      0.00000000      0.05203613      0.03289539
##  factor(hipress)1  factor(diabet)1  factor(lung)1  factor(hearth)1
##      0.08280116      0.19573875      0.02543506      0.33252481
##  factor(stroke)1  factor(psych)1  factor(iadla)1  factor(iadla)2
##      0.06802881      0.28272718      0.00000000      0.00000000
##  factor(iadla)3  factor(adlwa)1  factor(adlwa)2  factor(adlwa)3
##      0.00000000      0.14459462      0.36550972      0.48845297
##      edyears          feduc          meduc  log(income + 1)
##      0.00000000      0.00000000      0.00000000      0.00000000
##  factor(insur)1
##      0.00000000
##
## cfun= bcave
##      (Intercept)          age  factor(gender)1  factor(race)1
##      1.978829e+00  -5.216532e-05      0.000000e+00      0.000000e+00
##  factor(hispan)1 factor(marital)1  factor(arthri)1  factor(cancer)1
##      0.000000e+00      0.000000e+00      3.802145e-02      1.872132e-02
##  factor(hipress)1  factor(diabet)1  factor(lung)1  factor(hearth)1
##      1.222130e-01      1.954667e-01      2.796854e-02      3.279856e-01
##  factor(stroke)1  factor(psych)1  factor(iadla)1  factor(iadla)2
##      7.044509e-02      2.903266e-01      0.000000e+00      0.000000e+00
##  factor(iadla)3  factor(adlwa)1  factor(adlwa)2  factor(adlwa)3
##      0.000000e+00      2.729839e-01      3.864905e-01      5.060047e-01
##      edyears          feduc          meduc  log(income + 1)
##      0.000000e+00      0.000000e+00      0.000000e+00      0.000000e+00
##  factor(insur)1
##      0.000000e+00
##

```

```

## cfun= ccave
##      (Intercept)                age  factor(gender)1  factor(race)1
##      1.976567e+00  -4.375527e-05  0.000000e+00  0.000000e+00
##  factor(hispan)1  factor(marital)1  factor(arthri)1  factor(cancer)1
##      0.000000e+00  0.000000e+00  3.463168e-02  1.715012e-02
##  factor(hipress)1  factor(diabet)1  factor(lung)1  factor(hearth)1
##      1.290332e-01  1.881076e-01  2.466065e-02  3.285646e-01
##  factor(stroke)1  factor(psych)1  factor(iadla)1  factor(iadla)2
##      6.225529e-02  2.828722e-01  0.000000e+00  0.000000e+00
##  factor(iadla)3  factor(adlwa)1  factor(adlwa)2  factor(adlwa)3
##      0.000000e+00  2.736037e-01  3.977317e-01  5.175466e-01
##      edyears                feduc                meduc  log(income + 1)
##      0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##  factor(insur)1
##      0.000000e+00
##
## cfun= dcave
##      (Intercept)                age  factor(gender)1  factor(race)1
##      1.82551207  0.00000000  0.00000000  0.00000000
##  factor(hispan)1  factor(marital)1  factor(arthri)1  factor(cancer)1
##      0.00000000  0.00000000  0.02685254  0.00000000
##  factor(hipress)1  factor(diabet)1  factor(lung)1  factor(hearth)1
##      0.05332835  0.02872780  0.00000000  0.35662745
##  factor(stroke)1  factor(psych)1  factor(iadla)1  factor(iadla)2
##      0.00000000  0.03407841  0.00000000  0.00000000
##  factor(iadla)3  factor(adlwa)1  factor(adlwa)2  factor(adlwa)3
##      0.00000000  0.00000000  0.00000000  0.60245837
##      edyears                feduc                meduc  log(income + 1)
##      0.00000000  0.00000000  0.00000000  0.00000000
##  factor(insur)1
##      0.00000000
##
## cfun= ecave
##      (Intercept)                age  factor(gender)1  factor(race)1
##      1.881486009  0.000000000  0.000000000  0.000000000
##  factor(hispan)1  factor(marital)1  factor(arthri)1  factor(cancer)1
##      0.000000000  0.000000000  0.034132803  0.006426225
##  factor(hipress)1  factor(diabet)1  factor(lung)1  factor(hearth)1
##      0.066216272  0.067616141  0.000000000  0.346514697
##  factor(stroke)1  factor(psych)1  factor(iadla)1  factor(iadla)2
##      0.000000000  0.080270550  0.000000000  0.000000000
##  factor(iadla)3  factor(adlwa)1  factor(adlwa)2  factor(adlwa)3
##      0.000000000  0.046662355  0.359270807  0.592414299
##      edyears                feduc                meduc  log(income + 1)
##      0.000000000  0.000000000  0.000000000  0.000000000
##  factor(insur)1
##      0.000000000
##
## cfun= gcave
##      (Intercept)                age  factor(gender)1  factor(race)1
##      1.78471352  0.000000000  0.000000000  0.000000000
##  factor(hispan)1  factor(marital)1  factor(arthri)1  factor(cancer)1
##      0.000000000  0.000000000  0.02635143  0.000000000
##  factor(hipress)1  factor(diabet)1  factor(lung)1  factor(hearth)1

```



```

##      0.06526733      0.01393760      0.00000000      0.34003347
## factor(stroke)1 factor(psych)1 factor(iadla)1 factor(iadla)2
##      0.00000000      0.02149100      0.00000000      0.00000000
## factor(iadla)3 factor(adlwa)1 factor(adlwa)2 factor(adlwa)3
##      0.00000000      0.00000000      0.00000000      0.65357731
##      edyears      feduc      meduc log(income + 1)
##      0.00000000      0.00000000      0.00000000      0.00000000
## factor(insur)1
##      0.00000000
##
## cfun= tcave
##      (Intercept)      age factor(gender)1 factor(race)1
##      1.969315265      0.000000000      0.000000000      0.000000000
## factor(hispan)1 factor(marital)1 factor(arthri)1 factor(cancer)1
##      0.000000000      0.000000000      0.064288784      0.030497991
## factor(hipress)1 factor(diabet)1 factor(lung)1 factor(hearth)1
##      0.076827493      0.244339304      0.030909720      0.325954228
## factor(stroke)1 factor(psych)1 factor(iadla)1 factor(iadla)2
##      0.130262217      0.306230170      0.001956634      0.000000000
## factor(iadla)3 factor(adlwa)1 factor(adlwa)2 factor(adlwa)3
##      0.000000000      0.200614987      0.372065923      0.460492140
##      edyears      feduc      meduc log(income + 1)
##      0.001441250      0.000000000      0.000000000      0.000000000
## factor(insur)1
##      0.000000000

```

## References

- Heritier, Stephane, Eva Cantoni, Samuel Copt, and Maria-Pia Victoria-Feser. 2009. *Robust Methods in Biostatistics*. Vol. 825. John Wiley & Sons.
- Wang, Zhu. 2020. "Unified Robust Estimation." *arXiv e-Prints*, October, arXiv:2010.02848. <https://arxiv.org/abs/2010.02848>.