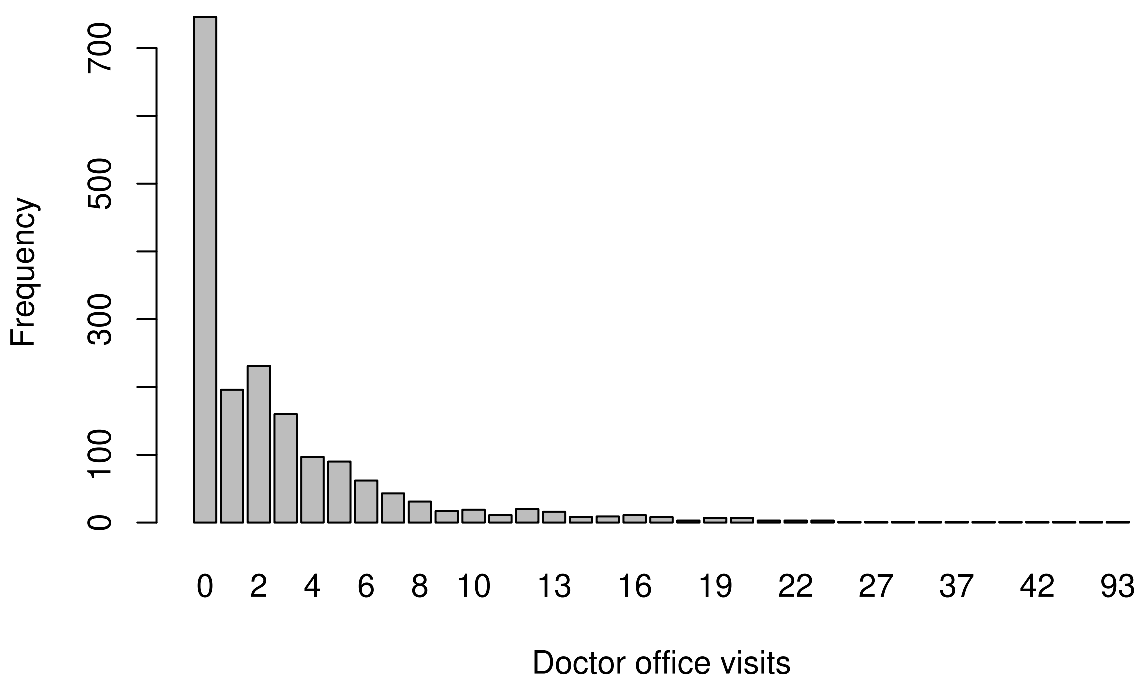
Variable Selection for Zero-inflated and Overdispersed Data with Application to Health Care Demand in Germany

Zhu Wang*

12/21/2022

This document reproduces the data analysis presented in Wang, Ma, and Wang (2015). In an effort to optimize the computing algorithms, the penalized regression can be slightly different. For a description of the theory behind application illustrated here we refer to the original manuscript. Riphahn, Wambach, and Million (2003) utilized a part of the German Socioeconomic Panel (GSOEP) data set to analyze the number of doctor visits. The original data have twelve annual waves from 1984 to 1995 for a representative sample of German households, which provide broad information on the health care utilization, current employment status, and the insurance arrangements under which subjects are protected. The data set contains number of doctor office visits for 1,812 West German men aged 25 to 65 years in the last three months of 1994. As shown in the figure, many doctor office visits are zeros, which can be difficult to fit with a Poisson or negative binomial model. Therefore, zero-inflated negative binomial (ZINB) model is considered.

```
require("mpath")
require("zic")
data(docvisits)
barplot(with(docvisits,table(docvisits)),ylab="Frequency",xlab="Doctor office visits")
```



We include the linear spline variables *age30* to *age60* and their interaction terms with the health satisfaction *health*.

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

```r
dt <- docvisits[,-(2:3)]
tmp <- model.matrix(~age30*health+age35*health+age40*health+age45*health+age50*health
                    +age55*health+age60*health, data=dt)[,-(1:9)]
dat <- cbind(dt, tmp)
```

Full ZINB model with all predictor variables.

```r
require("pscl")
m1 <- zeroinfl(docvisits~.|., data=dat, dist="negbin")
summary(m1)
```

```
##
## Call:
## zeroinfl(formula = docvisits ~ . | ., data = dat, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.0733 -0.6596 -0.3944  0.3006  9.9103
##
## Count model coefficients (negbin with log link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           2.412223   0.345625   6.979 2.97e-12 ***
## health               -0.163824   0.034487  -4.750 2.03e-06 ***
## handicap              0.266916   0.194518   1.372 0.170003
## hdegree              -0.002009   0.003292  -0.610 0.541794
## married              -0.147205   0.092835  -1.586 0.112815
## schooling            -0.004578   0.015392  -0.297 0.766150
## hhincome              0.004407   0.016158   0.273 0.785061
## children              0.017414   0.088406   0.197 0.843848
## self                 -0.359935   0.153892  -2.339 0.019341 *
## civil                -0.268081   0.160621  -1.669 0.095110 .
## bluec                 0.103446   0.086148   1.201 0.229830
## employed             -0.093915   0.107228  -0.876 0.381114
## public               -0.011404   0.139589  -0.082 0.934889
## addon                 0.364728   0.232492   1.569 0.116700
## age30TRUE             0.094128   0.362776   0.259 0.795277
## age35TRUE            -0.254808   0.367280  -0.694 0.487827
## age40TRUE             0.051516   0.398899   0.129 0.897242
## age45TRUE             0.720536   0.385682   1.868 0.061733 .
## age50TRUE             0.202441   0.341046   0.594 0.552786
## age55TRUE            -0.515859   0.307269  -1.679 0.093182 .
## age60TRUE             0.400798   0.313401   1.279 0.200944
## `age30TRUE:health`   -0.011746   0.052057  -0.226 0.821486
## `health:age35TRUE`    0.043191   0.054266   0.796 0.426078
## `health:age40TRUE`   -0.016689   0.061665  -0.271 0.786669
## `health:age45TRUE`   -0.101236   0.061449  -1.647 0.099458 .
## `health:age50TRUE`   -0.024100   0.053362  -0.452 0.651527
## `health:age55TRUE`    0.132920   0.051658   2.573 0.010080 *
## `health:age60TRUE`   -0.095085   0.055927  -1.700 0.089100 .
## Log(theta)            0.322396   0.090459   3.564 0.000365 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.310575   0.977635  -2.363   0.0181 *
```

```
## health               0.227409   0.099828   2.278   0.0227 *
## handicap            -0.334183   0.755147  -0.443   0.6581
## hdegree             -0.002431   0.015623  -0.156   0.8763
## married             -0.403730   0.246993  -1.635   0.1021
## schooling            0.018518   0.038249   0.484   0.6283
## hhincome            -0.038419   0.044317  -0.867   0.3860
## children             0.505679   0.235329   2.149   0.0316 *
## self                -0.248887   0.477695  -0.521   0.6024
## civil                0.020958   0.383246   0.055   0.9564
## bluec                0.022825   0.222147   0.103   0.9182
## employed            -0.084464   0.297028  -0.284   0.7761
## public              -0.230413   0.338168  -0.681   0.4956
## addon                0.299854   0.524368   0.572   0.5674
## age30TRUE           -1.678718   1.325983  -1.266   0.2055
## age35TRUE            0.900184   1.443436   0.624   0.5329
## age40TRUE           -0.650137   1.442160  -0.451   0.6521
## age45TRUE            2.999399   1.200079   2.499   0.0124 *
## age50TRUE           -2.955447   1.700700  -1.738   0.0822 .
## age55TRUE            0.335945   1.808686   0.186   0.8526
## age60TRUE           -2.336454   2.684804  -0.870   0.3842
## `age30TRUE:health`   0.227952   0.162793   1.400   0.1614
## `health:age35TRUE`  -0.110450   0.180807  -0.611   0.5413
## `health:age40TRUE`   0.115722   0.186281   0.621   0.5345
## `health:age45TRUE`  -0.409613   0.163907  -2.499   0.0125 *
## `health:age50TRUE`   0.250811   0.220847   1.136   0.2561
## `health:age55TRUE`   0.107929   0.233733   0.462   0.6443
## `health:age60TRUE`   0.196009   0.339419   0.577   0.5636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 1.3804
## Number of iterations in BFGS optimization: 65
## Log-likelihood: -3626 on 57 Df
```

```r
cat("loglik of zero-inflated model", logLik(m1))
```

```
## loglik of zero-inflated model -3625.926
```

```r
cat("BIC of zero-inflated model", AIC(m1, k=log(dim(dat)[1])))
```

```
## BIC of zero-inflated model 7679.476
```

```r
cat("AIC of zero-inflated model", AIC(m1))
```

```
## AIC of zero-inflated model 7365.851
```

Backward stepwise variable selection with significance level alpha=0.01.

```r
fitbe <- be.zeroinfl(m1, data=dat, dist="negbin", alpha=0.01,  trace=FALSE)
summary(fitbe)
```

```
##
## Call:
## zeroinfl(formula = eval(parse(text = out)), data = data, dist = dist)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -1.0201 -0.6459 -0.3942  0.2961  8.6647
##
## Count model coefficients (negbin with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.56617    0.09553  26.861  < 2e-16 ***
## health      -0.20133    0.01426 -14.117  < 2e-16 ***
## handicap     0.30305    0.08489   3.570 0.000357 ***
## self        -0.36627    0.11778  -3.110 0.001871 **
## civil       -0.33717    0.10434  -3.231 0.001232 **
## Log(theta)   0.23605    0.08981   2.628 0.008581 **
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.98383    0.37217  -8.017 1.08e-15 ***
## health       0.30104    0.04608   6.533 6.44e-11 ***
## age50TRUE   -1.00040    0.26019  -3.845 0.000121 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 1.2662
## Number of iterations in BFGS optimization: 19
## Log-likelihood: -3656 on 9 Df
```

```
cat("loglik of zero-inflated model with backward selection",logLik(fitbe))
```

```
## loglik of zero-inflated model with backward selection -3656.257
```

```
cat("BIC of zero-inflated model with backward selection", AIC(fitbe,k=log(dim(dat)[1])))
```

```
## BIC of zero-inflated model with backward selection 7380.034
```

Compute LASSO estimates.

```
fit.lasso <- zipath(docvisits~.|.,data = dat, family = "negbin", nlambda=100,
                    lambda.zero.min.ratio=0.001, maxit.em=300, maxit.theta=25,
                    theta.fixed=FALSE, trace=FALSE, penalty="enet", rescale=FALSE)
```

Estimated coefficient parameters with smallest BIC value.

```
minBic <- which.min(BIC(fit.lasso))
coef(fit.lasso, minBic)
```

```
## $count
##         (Intercept)              health            handicap             hdegree
##          2.30544059         -0.17379228          0.15521112          0.00000000
##             married           schooling            hhincome            children
##          0.00000000          0.00000000          0.00000000          0.00000000
##                self               civil               bluec            employed
##          0.00000000          0.00000000          0.00000000          0.00000000
##              public               addon           age30TRUE           age35TRUE
##          0.05667312          0.00000000          0.00000000          0.00000000
##           age40TRUE           age45TRUE           age50TRUE           age55TRUE
##          0.00000000          0.00000000          0.03813017          0.09871357
##           age60TRUE `age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
##          0.00000000          0.00000000          0.00000000          0.00000000
## `health:age45TRUE` `health:age50TRUE` `health:age55TRUE` `health:age60TRUE`
##          0.00000000          0.00000000          0.00000000          0.00000000
```

```
##
## $zero
##        (Intercept)              health            handicap             hdegree
##         -2.7004768           0.2519975           0.0000000           0.0000000
##            married           schooling             hhincome            children
##          0.0000000           0.0000000           0.0000000           0.1958848
##               self               civil               bluec            employed
##          0.0000000           0.0000000           0.0000000           0.0000000
##             public               addon           age30TRUE           age35TRUE
##          0.0000000           0.0000000           0.0000000           0.0000000
##          age40TRUE           age45TRUE           age50TRUE           age55TRUE
##          0.0000000           0.0000000          -0.3975528           0.0000000
##          age60TRUE `age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
##          0.0000000           0.0000000           0.0000000           0.0000000
## `health:age45TRUE` `health:age50TRUE` `health:age55TRUE` `health:age60TRUE`
##          0.0000000           0.0000000           0.0000000           0.0000000
```

```
cat("theta estimate", fit.lasso$theta[minBic])
```

```
## theta estimate 1.364578
```

Compute standard errors of coefficients and theta:

```
se(fit.lasso, minBic, log=FALSE)
```

```
## $count
## (Intercept)      health    handicap      public   age50TRUE   age55TRUE
##  0.15054073  0.01800520  0.10395229  0.09111704  0.10279568  0.11073563
##
## $zero
## (Intercept)      health    children   age50TRUE
##  0.28716188  0.03505522  0.15455293  0.18345847
##
## $theta
## [1] 0.1292102
```

Compute AIC, BIC, log-likelihood values of the selected model.

```
AIC(fit.lasso)[minBic]
```

```
##    0.048
## 7350.972
```

```
BIC(fit.lasso)[minBic]
```

```
##    0.048
## 7411.496
```

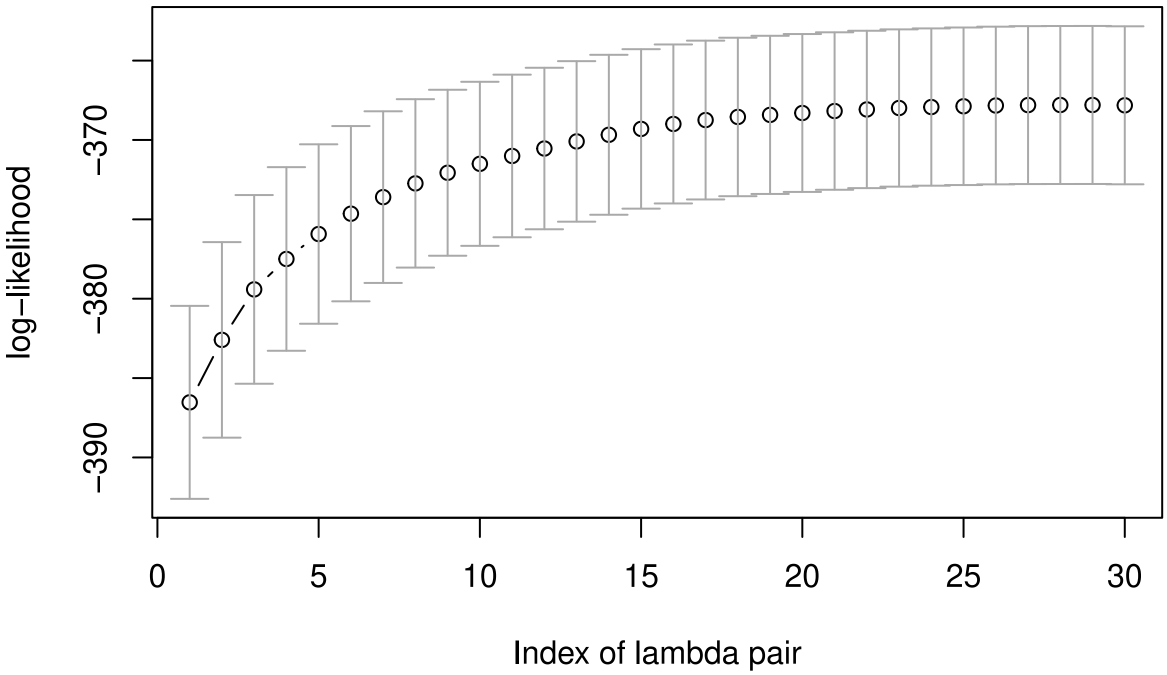
```
logLik(fit.lasso)[minBic]
```

```
## [1] -3664.486
```

Compute log-likelihood value via 10-fold cross-validation.

```
n <- dim(dat)[1]
K <- 10
set.seed(197)
foldid <- split(sample(1:n), rep(1:K, length = n))
fitcv <- cv.zipath(docvisits ~ . | ., data = dat, family = "negbin", nlambda=100,
                   lambda.count=fit.lasso$lambda.count[1:30],
```

```
                   lambda.zero= fit.lasso$lambda.zero[1:30],
                   maxit.em=300, maxit.theta=1, theta.fixed=FALSE,
                   penalty="enet", rescale=FALSE, foldid=foldid)
```




```
cat("cross-validated loglik", max(fitcv$cv))
```

```
## cross-validated loglik -367.7953
```

Compute MCP estimates. We compute solution paths for the first 30 pairs of shrinkage parameters (the EM algorithm can be slow), and then evaluate results as for the LASSO estimates. For cross-validation, set maximum number of iterations in estimating scaling parameter 1 (maxit.theta=1) to reduce computation costs.

```
tmp <- zipath(docvisits~.|.,data = dat, family = "negbin", gamma.count=2.7,
              gamma.zero=2.7, lambda.zero.min.ratio= 0.1, maxit=1, maxit.em=1,
              maxit.theta=2, theta.fixed=FALSE, penalty="mnet")
fit.mcp <- zipath(docvisits~.|.,data = dat, family = "negbin", gamma.count=2.7,
                  gamma.zero=2.7, lambda.count=tmp$lambda.count[1:30],
                  lambda.zero= tmp$lambda.zero[1:30], maxit.em=300, maxit.theta=25,
                  theta.fixed=FALSE, penalty="mnet")
```

Estimated coefficient parameters with smallest BIC value.

```
minBic <- which.min(BIC(fit.mcp))
coef(fit.mcp, minBic)
```

```
## $count
##       (Intercept)          health         handicap          hdegree
##         2.4857054      -0.1952631        0.2267380        0.0000000
##           married       schooling          hhincome         children
##         0.0000000       0.0000000        0.0000000        0.0000000
##              self           civil            bluec         employed
##        -0.3694420      -0.3314318        0.0000000        0.0000000
##            public           addon        age30TRUE        age35TRUE
```

```
##          0.0000000          0.0000000          0.0000000          0.0000000
##          age40TRUE          age45TRUE          age50TRUE          age55TRUE
##          0.0000000          0.0000000          0.0000000          0.2140696
##          age60TRUE `age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
## `health:age45TRUE` `health:age50TRUE` `health:age55TRUE` `health:age60TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
##
## $zero
##        (Intercept)             health           handicap            hdegree
##         -3.3545126          0.3169537          0.0000000          0.0000000
##            married          schooling           hhincome           children
##          0.0000000          0.0000000          0.0000000          0.4328507
##               self              civil              bluec           employed
##          0.0000000          0.0000000          0.0000000          0.0000000
##             public              addon          age30TRUE          age35TRUE
##          0.0000000          0.0000000          0.0000000          0.0000000
##          age40TRUE          age45TRUE          age50TRUE          age55TRUE
##          0.0000000          0.0000000         -0.6718362          0.0000000
##          age60TRUE `age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
## `health:age45TRUE` `health:age50TRUE` `health:age55TRUE` `health:age60TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
```

```
cat("theta estimate", fit.mcp$theta[minBic])
```

```
## theta estimate 1.276819
```

Compute standard errors of coefficients and theta:

```
se(fit.mcp, minBic, log=FALSE)
```

```
## $count
## (Intercept)      health    handicap        self       civil   age55TRUE
##  0.12483200  0.01814956  0.10380487  0.12199363  0.11886408  0.08564174
##
## $zero
## (Intercept)      health    children   age50TRUE
##   0.4086110   0.0450272   0.1772860   0.2509333
##
## $theta
## [1] 0.1336077
```

Compute AIC, BIC, log-likelihood values of the selected model.

```
AIC(fit.mcp)[minBic]
```

```
##   0.0228
## 7319.663
```

```
BIC(fit.mcp)[minBic]
```

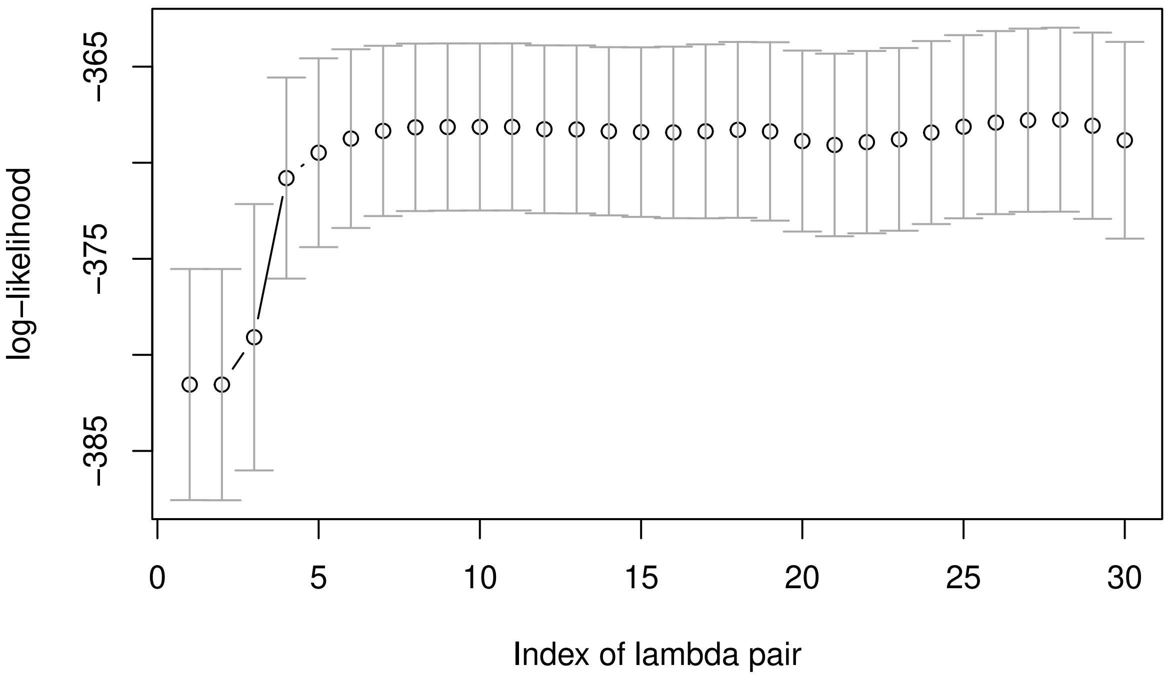
```
##   0.0228
## 7380.187
```

```
logLik(fit.mcp)[minBic]
```

```
## [1] -3648.831
```

Compute log-likelihood value via 10-fold cross-validation.

```
fitcv <- cv.zipath(docvisits ~ . | ., data = dat, family = "negbin", gamma.count=2.7,
                   gamma.zero=2.7, lambda.count=tmp$lambda.count[1:30],
                   lambda.zero= tmp$lambda.zero[1:30], maxit.em=300, maxit.theta=1,
                   theta.fixed=FALSE, penalty="mnet", rescale=FALSE, foldid=foldid)
```




```
cat("cross-validated loglik", max(fitcv$cv))
```

```
## cross-validated loglik -367.7593
```

Compute SCAD estimates.

```
tmp <- zipath(docvisits~.|.,data = dat, family = "negbin", gamma.count=2.5,
              gamma.zero=2.5, lambda.zero.min.ratio= 0.01, maxit=1, maxit.em=1,
              maxit.theta=2, theta.fixed=FALSE, penalty="snet")
fit.scad <- zipath(docvisits~.|.,data = dat, family = "negbin", gamma.count=2.5,
                   gamma.zero=2.5, lambda.count=tmp$lambda.count[1:30],
                   lambda.zero= tmp$lambda.zero[1:30], maxit.em=300, maxit.theta=25,
                   theta.fixed=FALSE, penalty="snet")
```

Estimated coefficient parameters with smallest BIC value.

```
minBic <- which.min(BIC(fit.scad))
coef(fit.scad, minBic)
```

```
## $count
##      (Intercept)          health        handicap         hdegree
##        2.4875256      -0.1951818       0.2258675       0.0000000
##          married       schooling         hhincome        children
##        0.0000000       0.0000000       0.0000000       0.0000000
##             self           civil           bluec        employed
##       -0.3692021      -0.3314020       0.0000000       0.0000000
##           public           addon       age30TRUE       age35TRUE
##        0.0000000       0.0000000       0.0000000       0.0000000
```

```
##          age40TRUE          age45TRUE          age50TRUE          age55TRUE
##          0.0000000          0.0000000          0.0000000          0.2142137
##          age60TRUE `age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
## `health:age45TRUE` `health:age50TRUE` `health:age55TRUE` `health:age60TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
##
## $zero
##        (Intercept)             health           handicap            hdegree
##         -3.3157884          0.3142947          0.0000000          0.0000000
##            married          schooling           hhincome           children
##          0.0000000          0.0000000          0.0000000          0.4137072
##               self              civil              bluec           employed
##          0.0000000          0.0000000          0.0000000          0.0000000
##             public              addon          age30TRUE          age35TRUE
##          0.0000000          0.0000000          0.0000000          0.0000000
##          age40TRUE          age45TRUE          age50TRUE          age55TRUE
##          0.0000000          0.0000000         -0.6729928          0.0000000
##          age60TRUE `age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
## `health:age45TRUE` `health:age50TRUE` `health:age55TRUE` `health:age60TRUE`
##          0.0000000          0.0000000          0.0000000          0.0000000
```

```
cat("theta estimate", fit.scad$theta[minBic])
```

```
## theta estimate 1.285932
```

Compute standard errors of coefficients and theta:

```
se(fit.scad, minBic, log=FALSE)
```

```
## $count
## (Intercept)      health    handicap        self       civil   age55TRUE
##  0.12477762  0.01812803  0.10376182  0.12195491  0.11884231  0.08555418
##
## $zero
## (Intercept)      health    children   age50TRUE
##  0.39924146  0.04438007  0.17474182  0.24738755
##
## $theta
## [1] 0.1336176
```

Compute AIC, BIC, log-likelihood values of the selected model.

```
AIC(fit.scad)[minBic]
```

```
##   0.0228
## 7319.682
```

```
BIC(fit.scad)[minBic]
```

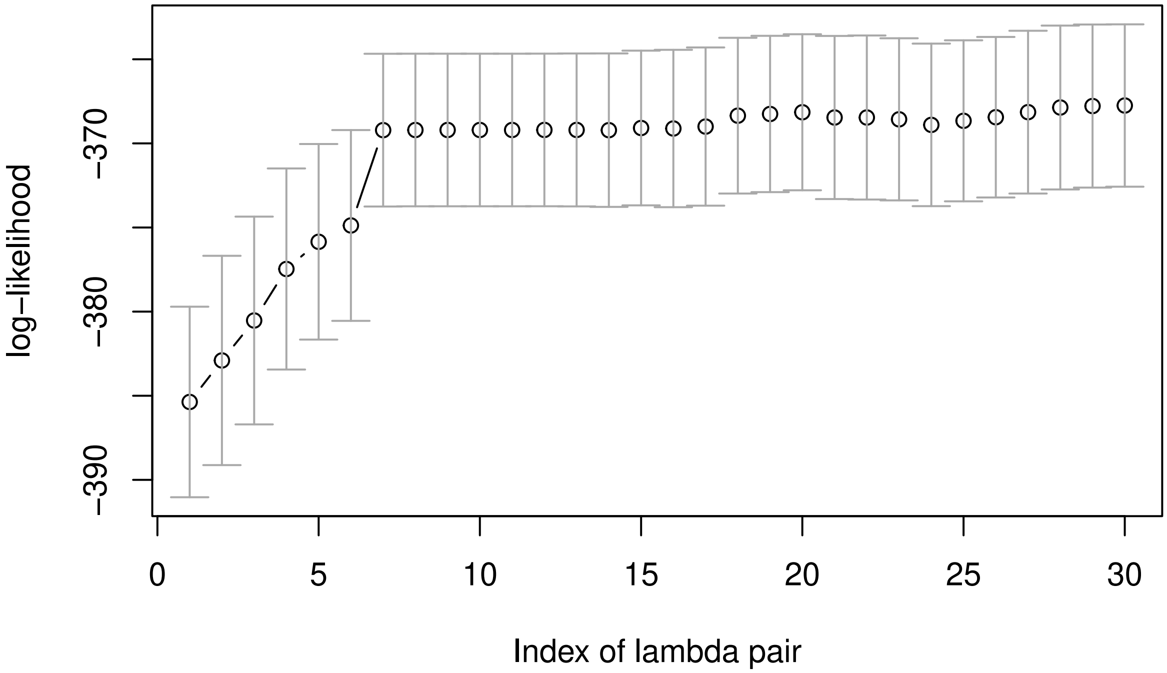
```
##   0.0228
## 7380.206
```

```
logLik(fit.scad)[minBic]
```

```
## [1] -3648.841
```

Compute log-likelihood value via 10-fold cross-validation.

```r
fitcv <- cv.zipath(docvisits ~ . | ., data = dat, family = "negbin", gamma.count=2.5,
                   gamma.zero=2.5, lambda.count=tmp$lambda.count[1:30],
                   lambda.zero= tmp$lambda.zero[1:30], maxit.em=300, maxit.theta=1,
                   theta.fixed=FALSE, penalty="snet", rescale=FALSE, foldid=foldid)
```




```r
cat("cross-validated loglik", max(fitcv$cv))
```

```
## cross-validated loglik -367.7493
```

```r
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] pscl_1.5.5     zic_0.9.1      mpath_0.4-2.23 pamr_1.56.1    survival_3.4-0
## [6] cluster_2.1.4  glmnet_4.1-6   Matrix_1.5-3
```

```
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.9          compiler_4.1.3      highr_0.9
##  [4] iterators_1.0.14    tools_4.1.3         rpart_4.1.19
##  [7] digest_0.6.31       evaluate_0.19       lifecycle_1.0.3
## [10] lattice_0.20-45     rlang_1.0.6         foreach_1.5.2
## [13] cli_3.5.0           yaml_2.3.6          parallel_4.1.3
## [16] gbm_2.1.8.1         xfun_0.35           fastmap_1.1.0
## [19] coda_0.19-4         stringr_1.5.0       knitr_1.41
## [22] vctrs_0.5.1         grid_4.1.3          bst_0.3-24
## [25] glue_1.6.2          WeightSVM_1.7-11    rmarkdown_2.19
## [28] magrittr_2.0.3      codetools_0.2-18    htmltools_0.5.4
## [31] splines_4.1.3       MASS_7.3-58.1       shape_1.4.6
## [34] numDeriv_2016.8-1.1 stringi_1.7.8       doParallel_1.0.17
```

References

Riphahn, Regina T, Achim Wambach, and Andreas Million. 2003. "Incentive Effects in the Demand for Health Care: A Bivariate Panel Count Data Estimation." *Journal of Applied Econometrics* 18 (4): 387–405.

Wang, Zhu, Shuangge Ma, and Ching-Yun Wang. 2015. "Variable Selection for Zero-Inflated and Overdispersed Data with Application to Health Care Demand in Germany." *Biometrical Journal* 33(29): 5192–5208. http://dx.doi.org/10.1002/bimj.201400143.