Variable Selection for Health Care Demand in Germany

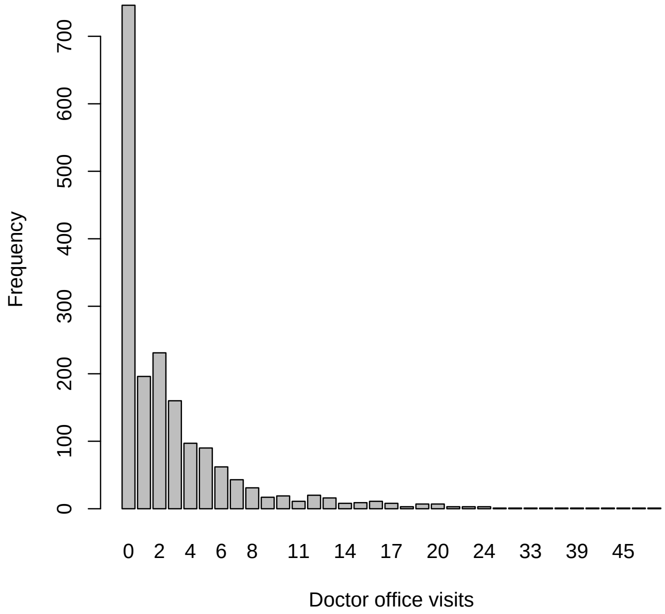
Zhu Wang
UT Health San Antonio
wangz1@uthscsa.edu

June 5, 2019

This document reproduces the data analysis presented in Wang et al. (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 et al. (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.

```
R> library("mpath")
R> library("zic")
R> library("pscl")
R> data(docvisits)
```

```
R> 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*.

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

Full ZINB model with all predictor variables.

```
R> m1 <- zeroinfl(docvisits ~ . | ., data = dat, dist = "negbin")
R> summary(m1)

Call:
zeroinfl(formula = docvisits ~ . | ., data = dat, dist = "negbin")

Pearson residuals:
   Min     1Q Median     3Q    Max 
-1.073 -0.660 -0.394  0.301  9.910 

Count model coefficients (negbin with log link):
                   Estimate Std. Error z value Pr(>|z|)    
(Intercept)         2.41222    0.34563    6.98    3e-12 ***
health             -0.16382    0.03449   -4.75    2e-06 ***
handicap            0.26691    0.19452    1.37  0.17001    
hdegree            -0.00201    0.00329   -0.61  0.54180    
married            -0.14720    0.09284   -1.59  0.11282    
```

```
schooling          -0.00458    0.01539   -0.30  0.76616
hhincome            0.00441    0.01616    0.27  0.78504
children            0.01741    0.08841    0.20  0.84385
self               -0.35994    0.15389   -2.34  0.01934 *
civil              -0.26809    0.16062   -1.67  0.09511 .
bluec               0.10345    0.08615    1.20  0.22983
employed           -0.09392    0.10723   -0.88  0.38110
public             -0.01141    0.13959   -0.08  0.93487
addon               0.36473    0.23249    1.57  0.11670
age30TRUE           0.09414    0.36278    0.26  0.79525
age35TRUE          -0.25482    0.36728   -0.69  0.48780
age40TRUE           0.05154    0.39890    0.13  0.89720
age45TRUE           0.72053    0.38568    1.87  0.06173 .
age50TRUE           0.20245    0.34105    0.59  0.55278
age55TRUE          -0.51587    0.30727   -1.68  0.09318 .
age60TRUE           0.40081    0.31340    1.28  0.20093
`age30TRUE:health` -0.01175    0.05206   -0.23  0.82146
`health:age35TRUE`  0.04319    0.05427    0.80  0.42605
`health:age40TRUE` -0.01669    0.06167   -0.27  0.78664
`health:age45TRUE` -0.10124    0.06145   -1.65  0.09946 .
`health:age50TRUE` -0.02410    0.05336   -0.45  0.65150
`health:age55TRUE`  0.13292    0.05166    2.57  0.01008 *
`health:age60TRUE` -0.09509    0.05593   -1.70  0.08909 .
Log(theta)          0.32239    0.09046    3.56  0.00037 ***

Zero-inflation model coefficients (binomial with logit link):
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -2.31059    0.97764   -2.36    0.018 *
health              0.22741    0.09983    2.28    0.023 *
handicap           -0.33423    0.75516   -0.44    0.658
hdegree            -0.00243    0.01562   -0.16    0.876
married            -0.40373    0.24700   -1.63    0.102
schooling           0.01852    0.03825    0.48    0.628
hhincome           -0.03842    0.04432   -0.87    0.386
children            0.50569    0.23533    2.15    0.032 *
self               -0.24892    0.47771   -0.52    0.602
civil               0.02095    0.38325    0.05    0.956
bluec               0.02283    0.22215    0.10    0.918
employed           -0.08448    0.29703   -0.28    0.776
public             -0.23043    0.33817   -0.68    0.496
addon               0.29983    0.52438    0.57    0.567
age30TRUE          -1.67862    1.32598   -1.27    0.206
age35TRUE           0.90000    1.44345    0.62    0.533
age40TRUE          -0.64989    1.44217   -0.45    0.652
age45TRUE           2.99929    1.20006    2.50    0.012 *
age50TRUE          -2.95569    1.70089   -1.74    0.082 .
age55TRUE           0.33612    1.80886    0.19    0.853
age60TRUE          -2.33629    2.68475   -0.87    0.384
`age30TRUE:health`  0.22794    0.16279    1.40    0.161
`health:age35TRUE` -0.11043    0.18081   -0.61    0.541
```

```
`health:age40TRUE`  0.11569    0.18628    0.62    0.535
`health:age45TRUE` -0.40960    0.16390   -2.50    0.012 *
`health:age50TRUE`  0.25083    0.22087    1.14    0.256
`health:age55TRUE`  0.10792    0.23375    0.46    0.644
`health:age60TRUE`  0.19599    0.33942    0.58    0.564
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 1.38
Number of iterations in BFGS optimization: 63
Log-likelihood: -3.63e+03 on 57 Df
```

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

```
loglik of zero-inflated model -3626
```

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

```
BIC of zero-inflated model 7679
```

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

```
AIC of zero-inflated model 7366
```

Backward stepwise variable selection with significance level alpha=0.01.

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

```
Call:
zeroinfl(formula = eval(parse(text = out)), data = data, dist = dist)

Pearson residuals:
   Min     1Q Median     3Q    Max 
-1.020 -0.646 -0.394  0.296  8.665 

Count model coefficients (negbin with log link):
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)   2.5662     0.0955   26.86  < 2e-16 ***
health       -0.2013     0.0143  -14.12  < 2e-16 ***
handicap      0.3031     0.0849    3.57  0.00036 ***
self         -0.3663     0.1178   -3.11  0.00187 ** 
civil        -0.3372     0.1043   -3.23  0.00123 ** 
Log(theta)    0.2360     0.0898    2.63  0.00858 ** 

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  -2.9838     0.3722   -8.02  1.1e-15 ***
health        0.3010     0.0461    6.53  6.4e-11 ***
age50TRUE    -1.0004     0.2602   -3.84  0.00012 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 1.266
Number of iterations in BFGS optimization: 17
Log-likelihood: -3.66e+03 on 9 Df
```

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

```
loglik of zero-inflated model with backward selection -3656
```

```
R> 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
```

Compute LASSO estimates.

```
R> 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.

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

```
$count
       (Intercept)             health           handicap 
           2.30544           -0.17379            0.15521 
           hdegree            married          schooling 
           0.00000            0.00000            0.00000 
          hhincome           children               self 
           0.00000            0.00000            0.00000 
             civil              bluec           employed 
           0.00000            0.00000            0.00000 
            public              addon          age30TRUE 
           0.05667            0.00000            0.00000 
         age35TRUE          age40TRUE          age45TRUE 
           0.00000            0.00000            0.00000 
         age50TRUE          age55TRUE          age60TRUE 
           0.03813            0.09871            0.00000 
`age30TRUE:health` `health:age35TRUE` `health:age40TRUE` 
           0.00000            0.00000            0.00000 
`health:age45TRUE` `health:age50TRUE` `health:age55TRUE` 
           0.00000            0.00000            0.00000 
`health:age60TRUE` 
           0.00000 

$zero
       (Intercept)             health           handicap 
```

```
            -2.7005              0.2520              0.0000
            hdegree             married           schooling
             0.0000              0.0000              0.0000
           hhincome            children                self
             0.0000              0.1959              0.0000
              civil               bluec            employed
             0.0000              0.0000              0.0000
             public               addon           age30TRUE
             0.0000              0.0000              0.0000
          age35TRUE           age40TRUE           age45TRUE
             0.0000              0.0000              0.0000
          age50TRUE           age55TRUE           age60TRUE
            -0.3976              0.0000              0.0000
`age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
             0.0000              0.0000              0.0000
`health:age45TRUE` `health:age50TRUE` `health:age55TRUE`
             0.0000              0.0000              0.0000
`health:age60TRUE`
             0.0000
```

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

```
theta estimate 1.365
```

Compute standard errors of coefficients and theta:

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

```
$count
(Intercept)      health    handicap      public   age50TRUE
    0.15054     0.01801     0.10395     0.09112     0.10280
  age55TRUE
    0.11074

$zero
(Intercept)      health    children   age50TRUE
    0.28716     0.03506     0.15455     0.18346

$theta
[1] 0.1292
```

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

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

```
0.048
 7351
```

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

```
0.048
 7411
```

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

```
[1] -3664
```

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

```
R> n <- dim(dat)[1]
R> K <- 10
R> set.seed(197)
R> foldid <- split(sample(1:n), rep(1:K, length = n))
R> 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)
R> cat("cross-validated loglik", max(fitcv$cv))
```

```
cross-validated loglik -367.8
```

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.

```
R> 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")
R> 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.

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

```
$count
       (Intercept)             health           handicap
            2.4857            -0.1953             0.2267
           hdegree            married          schooling
            0.0000             0.0000             0.0000
          hhincome           children               self
            0.0000             0.0000            -0.3694
             civil              bluec           employed
           -0.3314             0.0000             0.0000
            public              addon          age30TRUE
            0.0000             0.0000             0.0000
         age35TRUE          age40TRUE          age45TRUE
            0.0000             0.0000             0.0000
         age50TRUE          age55TRUE          age60TRUE 
            0.0000             0.2141             0.0000 
`age30TRUE:health` `health:age35TRUE` `health:age40TRUE` 
            0.0000             0.0000             0.0000 
`health:age45TRUE` `health:age50TRUE` `health:age55TRUE` 
            0.0000             0.0000             0.0000 
`health:age60TRUE` 
            0.0000 

$zero
       (Intercept)             health           handicap 
           -3.3545             0.3170             0.0000 
           hdegree            married          schooling 
            0.0000             0.0000             0.0000 
          hhincome           children               self 
            0.0000             0.4329             0.0000 
             civil              bluec           employed 
            0.0000             0.0000             0.0000 
            public              addon          age30TRUE 
            0.0000             0.0000             0.0000 
         age35TRUE          age40TRUE          age45TRUE 
            0.0000             0.0000             0.0000 
         age50TRUE          age55TRUE          age60TRUE 
           -0.6718             0.0000             0.0000 
`age30TRUE:health` `health:age35TRUE` `health:age40TRUE` 
            0.0000             0.0000             0.0000 
`health:age45TRUE` `health:age50TRUE` `health:age55TRUE` 
            0.0000             0.0000             0.0000 
`health:age60TRUE` 
            0.0000 
```

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

```
theta estimate 1.277
```

Compute standard errors of coefficients and theta:

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

```
$count
(Intercept)      health    handicap        self       civil 
    0.12483     0.01815     0.10380     0.12199     0.11886 
  age55TRUE 
    0.08564 

$zero
(Intercept)      health    children   age50TRUE 
    0.40861     0.04503     0.17729     0.25093 

$theta
[1] 0.1336
```

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

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

```
0.0228
  7320
```

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

```
0.0228
  7380
```

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

```
[1] -3649
```

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

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

```
cross-validated loglik -367.8
```

Compute SCAD estimates.

```
R> 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")
R> 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.

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

```
$count
      (Intercept)            health          handicap
           2.4875           -0.1952            0.2259
          hdegree           married         schooling
           0.0000            0.0000            0.0000
         hhincome          children              self
           0.0000            0.0000           -0.3692
            civil             bluec          employed
          -0.3314            0.0000            0.0000
           public             addon          age30TRUE
           0.0000            0.0000            0.0000
         age35TRUE          age40TRUE          age45TRUE
            0.0000             0.0000             0.0000
         age50TRUE          age55TRUE          age60TRUE
            0.0000             0.2142             0.0000
`age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
            0.0000             0.0000             0.0000
`health:age45TRUE` `health:age50TRUE` `health:age55TRUE`
            0.0000             0.0000             0.0000
`health:age60TRUE`
            0.0000

$zero
       (Intercept)             health           handicap
           -3.3158             0.3143             0.0000
           hdegree            married          schooling
            0.0000             0.0000             0.0000
          hhincome           children               self
            0.0000             0.4137             0.0000
             civil              bluec           employed
            0.0000             0.0000             0.0000
            public              addon          age30TRUE
            0.0000             0.0000             0.0000
         age35TRUE          age40TRUE          age45TRUE
            0.0000             0.0000             0.0000
         age50TRUE          age55TRUE          age60TRUE
           -0.6730             0.0000             0.0000
`age30TRUE:health` `health:age35TRUE` `health:age40TRUE`
            0.0000             0.0000             0.0000
`health:age45TRUE` `health:age50TRUE` `health:age55TRUE`
            0.0000             0.0000             0.0000
`health:age60TRUE`
            0.0000
```

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

```
theta estimate 1.286
```

Compute standard errors of coefficients and theta:

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

```
$count
(Intercept)      health    handicap        self       civil
    0.12478     0.01813     0.10376     0.12195     0.11884
  age55TRUE
    0.08555

$zero
(Intercept)      health    children   age50TRUE
    0.39924     0.04438     0.17474     0.24739
```

```
$theta
[1] 0.1336
```

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

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

0.0228
  7320

R> BIC(fit.scad)[minBic]

0.0228
  7380

R> logLik(fit.scad)[minBic]

[1] -3649
```

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 -367.7
```

Running time for the entire analysis.

```
R> print(proc.time() - ptm)

   user  system elapsed
766.155   0.292 775.944

R> sessionInfo()

R version 3.6.0 (2019-04-26)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.2 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
[7] base

other attached packages:
[1] pscl_1.5.2   zic_0.9.1    mpath_0.3-15

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.0         lattice_0.20-38   codetools_0.2-16
 [4] glmnet_2.0-16      foreach_1.4.4     MASS_7.3-51.1
 [7] grid_3.6.0         gtable_0.2.0      coda_0.19-2
[10] doParallel_1.0.14 bst_0.3-18        rpart_4.1-15
[13] Matrix_1.2-17      splines_3.6.0     iterators_1.0.10
[16] tools_3.6.0        survival_2.43-3   numDeriv_2016.8-1
[19] parallel_3.6.0     compiler_3.6.0    gbm_2.1.5
[22] gridExtra_2.3
```

References

Regina T Riphahn, Achim Wambach, and Andreas Million. Incentive effects in the demand for health care: a bivariate panel count data estimation. *Journal of Applied Econometrics*, 18(4):387–405, 2003.

Zhu Wang, Shuangge Ma, and Ching-Yun Wang. Variable selection for zero-inflated and overdispersed data with application to health care demand in germany. *Biometrical Journal*, 33(29):5192–208, 2015. URL http://dx.doi.org/10.1002/bimj.201400143.