Sparse Modelling with Categorical Predictors

A Brief Overview

Shrayan Roy

Indian Statistical Institute, Delhi Center

20/05/2023

Sparse Modeling :

- It refers to a modeling approach that aims to identify and utilize a small number of important features or variables in a dataset, while ignoring or discarding the less important ones.
- Here, "sparse" refers to the idea of using a small number of important features or variables in a dataset, while ignoring or discarding the less important ones.
- Some frequently used Sparse Modeling approaches are - Lasso Regression, Ridge Regression, Elastic Net Regression.
- All these three types also falls under the category of Penalized Regression, as well as Constrained Regression.
- They help in selecting the important variables by reducing the coefficient of not so important (in a relative sense) to zero.
- In case of Lasso, the coefficient of that variables in the model become exactly equal to zero.
- Thus, using the above approaches we basically select coefficients, rather than variables.
- If it is a metric variable, it simply means selecting that variable only. But this becomes problematic for categorical explanatory variable.

Sparse Modeling(Contd.):

- Let us illustrate through example - if a categorical variable have four categories. Then, in a **linear regression model** three dummy variables are needed for that categorical variable. It may happen that using lasso, one dummy is selected. Also, the selection depends on the **reference category chosen**.

- So, we need to select coefficient in a group wise manner. For that, another sparse modeling approach is available called **Group-Lasso**. This method has a separate objective. But, we can use this for our purpose by considering group of coefficients as the collection of coefficient of the dummy variables corresponding to categorical variables.

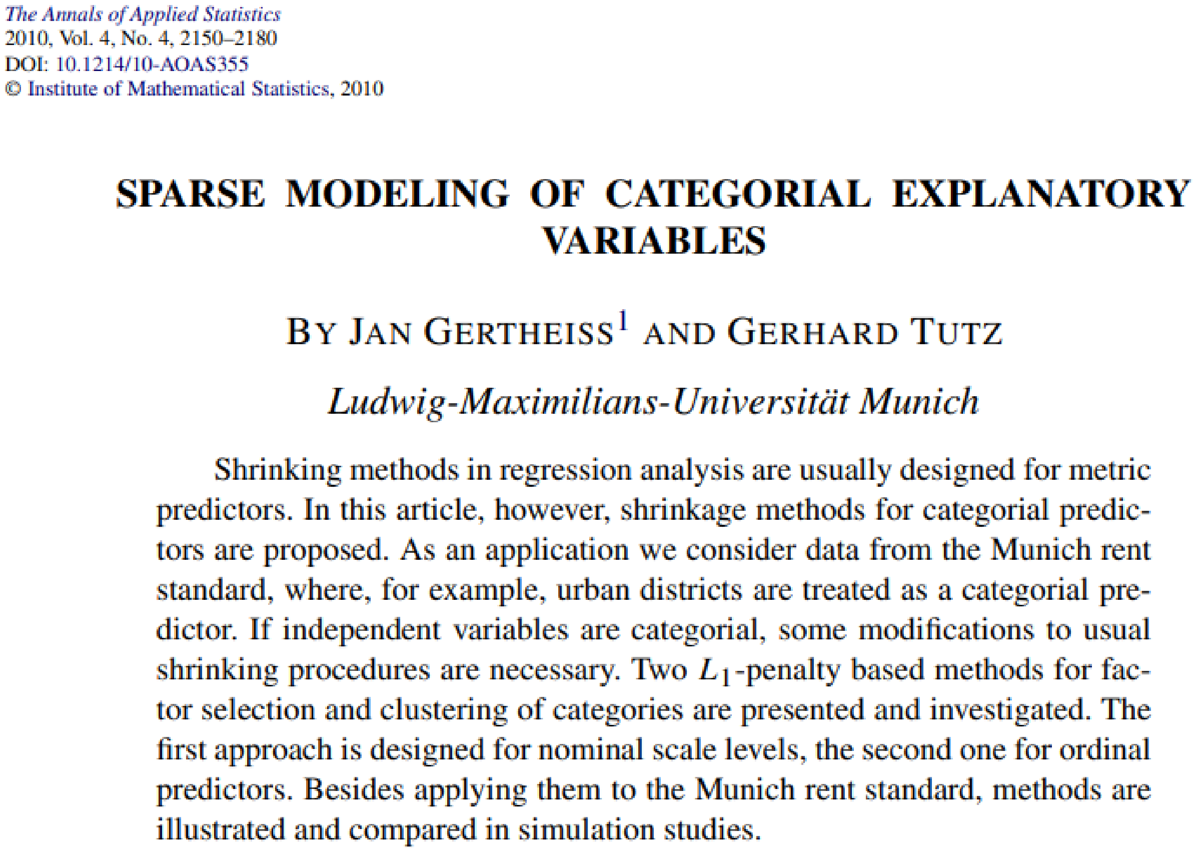
- But, another important question when we have categorical explanatory variable is that -

  Which levels within a categorical explanatory variable are similar with respect to response ?

- This type of questions are particularly important, when the categorical variables have very large number of levels. For example - In Bio-statistics such situation arises very frequently. But, Group-Lasso doesn't take care of this issue.

- Possible solutions may be to use Variable-Fusion and Fused Lasso.But, we need to tackle the situation differently depending upon Nominal or Ordinal nature of the variable.

- These issues and there possible solutions are addressed in the following paper :



The Annals of Applied Statistics
2010, Vol. 4, No. 4, 2150–2180
DOI: 10.1214/10-AOAS355

SPARSE MODELING OF CATEGORIAL EXPLANATORY VARIABLES

BY JAN GERTHEISS[1] AND GERHARD TUTZ

Ludwig-Maximilians-Universität Munich

Shrinking methods in regression analysis are usually designed for metric predictors. In this article, however, shrinkage methods for categorial predictors are proposed. As an application we consider data from the Munich rent standard, where, for example, urban districts are treated as a categorial predictor. If independent variables are categorial, some modifications to usual shrinking procedures are necessary. Two L_1-penalty based methods for factor selection and clustering of categories are presented and investigated. The first approach is designed for nominal scale levels, the second one for ordinal predictors. Besides applying them to the Munich rent standard, methods are illustrated and compared in simulation studies.

Proposed Solution :

In the following we consider the penalized least squares criterion -

$$Q_p(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda J(\beta)$$

Where, $\mathbf{X}$ is the design matrix. p denotes the number of variables. β is the parameter vector. $\mathbf{y}$ is the vector of observed values of response. Here, response is assumed to be continuous. $J(\beta)$ is the penalty term. The estimate of β is obtained by -

$$\hat{\beta} = \underset{\beta}{argmin}\ Q_p(\beta)$$

- Now, the question is what should be the penalty $J(\beta)$, so that we will get the benefit of both group lasso and variable-fusion ?
- To put things simple, we will assume that our linear regression model consists of only one predictor, which is categorical.
- Corresponding to the categorical predictor, we have k dummy variables $x_1, x_2, \dots, x_k$.

Choice of penalty $J(\beta)$:

For Nominal Scale :

$$J(\beta) = \sum_{i>j} w_{ij}|\beta_i - \beta_j|$$

with weights w_{ij} and β_i denoting the coefficient of dummy x_i. Since the ordering of $x_0, \ldots, x_k$ is arbitrary, not only differences $\beta_i - \beta_{i-1}$ (as in original fusion methodology), but all differences $\beta_i - \beta_{i-1}$ are considered. Since $i = 0$ is chosen as reference, $\beta_0 = 0$ is fixed.

For Ordinal Scale :

$$J(\beta) = \sum_{i=1}^{k} w_i|\beta_i - \beta_{i-1}|$$

Since, in the case of ordered categories the ordering of dummy coefficients is meaningful, consecutive differences are considered.

Computational Issues :

- To find the actual solution, constrained minimization is done instead of penalized minimization.
- For estimation purpose original parameters are transformed into $\theta_{ij} = \beta_i - \beta_j$. So, we have $k(k-1)$ number of parameters. (**For nominal scale variables**)
- But, one has to take care of the restrictions $\theta_{ij} = \theta_{i0} - \theta_{j0}\ \forall i.\ j$ for estimation purpose.
- For practical estimation, parameters θ_{ij} are additionally split into positive and negative parts, that is, $\theta_{ij} = \theta_{ij}^{+} - \theta_{ij}^{-}$
- **Quadratic programming** is used for minimization purpose. But, there can be some numerical problems with this.
- That's why an approximate solution can be found by -

$$\hat{\theta}_{\gamma,\lambda} = \underset{\theta}{argmin}\ \{(\mathbf{y} - \mathbf{Z}\theta)^T(\mathbf{y} - \mathbf{Z}\theta) + \gamma \sum_{i>j>0} (\theta_{ij} - \theta_{i0} + \theta_{j0})^2 + \lambda \sum_{i>j} |\theta_{ij}|\}$$

- Where, Z is so that $\mathbf{Z}\theta = \mathbf{X}\beta$.

Computational Issues(contd.):

- One can reformulate the problem as a lasso problem. If matrix A represents restrictions $\theta_{ij} = \theta_{i0} - \theta_{j0}$ in terms of $\mathbf{A}\theta = 0$. Then, with augmented data $\tilde{Z} = (Z^T, \sqrt{\gamma}A^T)^T$ and $\tilde{y} = (y^T, 0)^T$, we have -

$$\hat{\theta}_{\gamma,\lambda} = \underset{\theta}{argmin} \, \{(\mathbf{\tilde{y} - \tilde{Z}\theta)^T(\tilde{y} - \tilde{Z}\theta)} + \lambda \sum_{i>j} |\theta_{ij}|\}$$

- By doing this, we can compute the whole path of $\hat{\theta}_{\gamma,\lambda}$.

- In case of ordinal predictors also, we can reformulate the problem in a similar manner, by defining $\delta_i = \beta_i - \beta_{i-1} \; \forall \; i$. Here also we can form a similar lasso problem.

Multiple Input :

- In general, we can have more than one predictor. Then, our penalization criterion will be different.
- Since, our concern is about categorical predictors only. We will consider a model consisting of p categorical predictors $x_1, x_2, ..x_p$ with levels $0, \ldots, k_l$ for variable x_l ($l = 1, \ldots, p$, and fixed p).
- The corresponding penalty term would be $J(\beta) = \sum_{l=1}^{p} J_l(\beta_l)$ with

  $J(\beta_l) = \sum_{i>j} w_{ij}^{(l)} |\beta_{li} - \beta_{lj}|$ or, $J(\beta_l) = \sum_{i=1}^{k} w_i^{(l)} |\beta_i - \beta_{i-1}|$

  depending upon the scale level of predictor x_l. The first expression refers to nominal covariates, the second to ordinal ones.

Choice of Weights :

- In many situations weights $w_{ij}^{(l)} \neq 1$ are preferred over the simple weights $w_{ij}^{(l)} = 1$.
- The higher the weights, the higher penalization is made, the more quickly corresponding coefficient will become zero.
- For nominal variables Bondell and Reich (2009) suggested the weights -

$$w_{ij}^{(l)} = (k_l + 1)^{-1} \sqrt{\frac{n_i^{(l)} + n_j^{(l)}}{n}}$$

Where, $n_i^{(l)}$ and $n_j^{(l)}$ respectively denote the number of observations corresponding to level i,j of x_l and k_l is the number of levels of variable.

- There is also adaptive version of weights, where weights contain an additional factor $|\beta_{li}^{LS} - \beta_{lj}^{LS}|^{-1}$. Basically, if ordinary least square estimates corresponding to two dummy variables are very close. Then, these adaptive version of weights put higher weights i.e. higher penalties.

Refitting Procedures :

- The most attractive features of the methods described above are variable selection and clustering.
- However, due to penalization,estimates are obviously biased.
- In regression analysis in general — we are also interested in parameter estimation and prediction accuracy.
- In order to reduce the bias, refitting procedures have been proposed by several authors.
- In refitting approach, we refit the the model using the variables having non-zero coefficients with fused levels.

Application :

- We will apply the above discussed method on a dataset.
- Implementation is available in **R**.
- But, It is more general. Infact, the available implementation uses **Smurf Algorithm** .
- Which is discussed in the paper Devriendt, S., K. Antonio, T. Reynkens, and R. Verbelen. 2021 : Sparse regression with Multi-type Regularized Feature modeling.
- This paper extends the application to glm class of families as well.
- We will discuss about it briefly.

Multi-type Lasso Regularization :

- Consider a response y and the corresponding model matrix $\mathbf{X}$.
- The objective function for a regularized generalized linear model with a multi-type penalty is -

$$O(\boldsymbol{\beta}; \mathbf{X}, \mathbf{y}) = f(\beta; \mathbf{X}, \mathbf{y}) + \lambda \sum_{j=0}^{J} g_j(\beta_j)$$

- Where $f(\cdot)$ is minus the log-likelihood function divided by the sample size, $g_j(.)$ a convex function for all $j \in \{0, \ldots, J\}$ and β_j represents a subset of the full parameter vector β such that $(\beta_0, \beta_1, \ldots, \beta_J) = \boldsymbol{\beta}$, with β_0 the intercept.
- As the intercept is usually not regularized, we set $g_0(\cdot) = 0$.The penalty function $g_j(.)$ serve as a measure to avoid overfitting the data, while the tuning parameter λ controls the strength of the penalty.
- This paper discusses about different types of penalties, such as - Lasso, Group Lasso,Fused Lasso, Generalized Fused Lasso. And a method to apply these penalties by combining them.

Application on Data :

- We will apply the above discussed methods on "rent" dataset. It is available Here.Also, this data is available in "catdata" package in R.

- Let's get some insights about the data.

```
data(rent,package = "catdata")

dim(rent) # Number of Rows and Columns
```

```
## [1] 2053   13
```

```
str(rent) # About the variables
```

```
## 'data.frame':    2053 obs. of  13 variables:
##  $ rent     : num  741 716 528 554 698 ...
##  $ rentm    : num  10.9 11.01 8.38 8.52 6.98 ...
##  $ size     : int  68 65 63 65 100 81 55 79 52 77 ...
##  $ rooms    : int  2 2 3 3 4 4 2 3 1 3 ...
##  $ year     : num  1918 1995 1918 1983 1995 ...
##  $ area     : int  2 2 2 16 16 16 6 6 6 6 ...
##  $ good     : int  1 1 1 0 1 0 0 0 0 0
```

Data Preparation :

```
sum(is.na(rent))  #so, no na values
```

```
## [1] 0
```

```
# Urban district in Munich
rent$area <- as.factor(rent$area)

# Decade of construction
rent$year <- as.factor(floor(rent$year / 10) * 10)

# Number of rooms
rent$rooms <- as.factor(rent$rooms)

#Let's make a house quality variable
rent$quality <- as.factor(rent$good + 2*rent$best)
levels(rent$quality) <- c("Fair","Good","Excellent")
```

- Here, we want our predictors to be categorical, so we will make "size" also, categorical variable.

Data Preparation (Contd.):

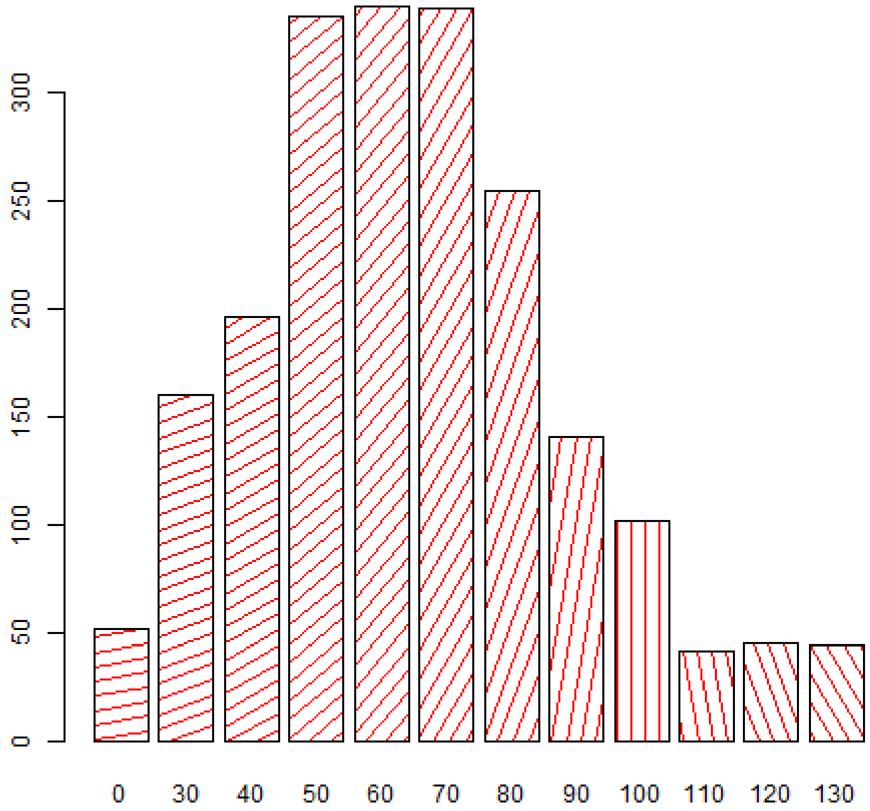
```
summary(rent$size)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    17.0    53.0    67.0    69.6    83.0   185.0
```

```
#But we need to also look that, at end parts, we have very few observations !
# Floor space divided in categories (0, 30), [30, 40), ...,  [120, 130),[130, Inf)

sizeClasses <- c(0, seq(30, 130, 10))
rent$size <- as.factor(sizeClasses[findInterval(rent$size, sizeClasses)])
```

Data Preparation (Contd.):

```
barplot(table(rent$size),angle= seq(10,120,length = 12),density = 10,col = "red")
```



What if we fit model using all variables ?

```
summary(lm(rentm ~.,data = rentData))
```

```
## 
## Call:
## lm(formula = rentm ~ ., data = rentData)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -6.5768 -1.2261 -0.0106  1.2664  7.4314 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      12.55465    0.44095  28.472  < 2e-16 ***
## size30           -1.75504    0.31960  -5.491 4.50e-08 ***
## size40           -2.93844    0.35074  -8.378  < 2e-16 ***
## size50           -3.30923    0.36710  -9.015  < 2e-16 ***
```

Variable Selection Using Lasso :

```
X.mat <- model.matrix(rentm ~ . ,data = rentData)[,-1]   #otherwise it is taking size0 into
lasso.model <- glmnet::glmnet(X.mat,rentData$rentm,alpha = 1,family = gaussian,lambda = 0.1)

#The variables for which, we have zero coefficient !
colnames(X.mat)[as.vector(coef(lasso.model) == 0)]
```

```
##  [1] "size50"     "size60"     "size90"     "size100"    "size110"   
##  [6] "size120"    "size130"    "rooms3"     "year1920"   "year1940"  
## [11] "year1960"   "year1970"   "year1980"   "area3"      "area5"     
## [16] "area6"      "area7"      "area8"      "area9"      "area10"    
## [21] "area11"     "area12"     "area13"     "area14"     "area16"    
## [26] "area18"     "area19"     "area20"     "area21"     "area22"    
## [31] "area23"     "area24"     "area25"     "warmno"     "kitchenyes"
```

- The coefficients of all dummies corresponding to a variable are not zero together !

What if we use Stepwise Criteria ?

```r
#stepwise selection can be done !
library(MASS)
stepAIC(lm(rentm ~. ,data =  rentData))
```

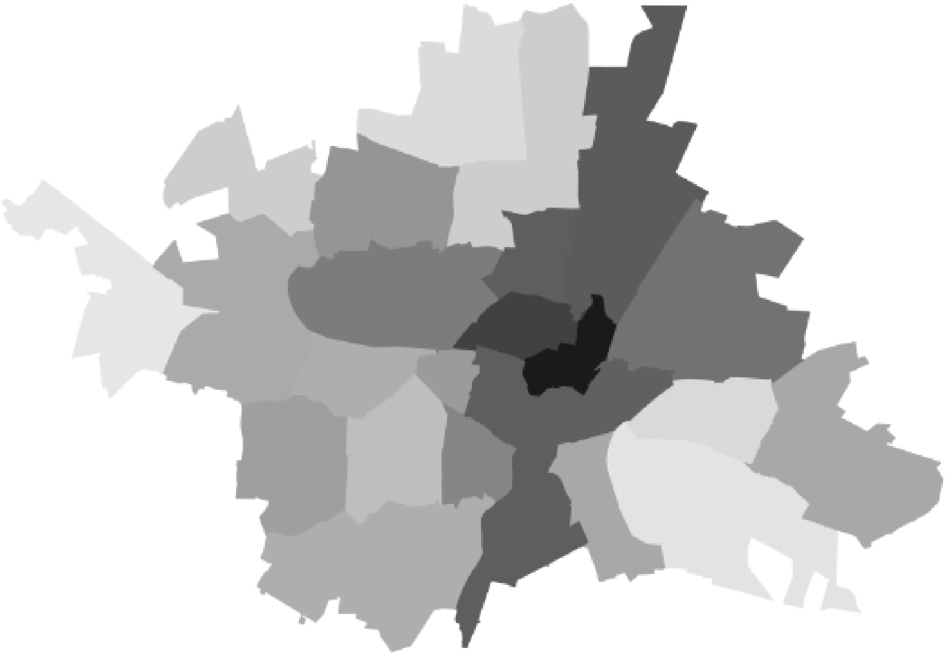
```
## Start:  AIC=2824.85
## rentm ~ size + rooms + year + area + warm + central + tiles + 
##     bathextra + kitchen + quality
## 
##             Df Sum of Sq    RSS    AIC
## - rooms      5     27.39 7716.0 2822.2
## <none>                   7688.6 2824.8
## - bathextra  1     38.39 7727.0 2833.1
## - tiles      1     91.91 7780.5 2847.2
## - quality    2    113.44 7802.0 2850.9
## - area      24    335.68 8024.3 2864.6
## - kitchen    1    180.33 7868.9 2870.4
```

- But, we are having the same problem of so many variables !

What about Level Fusion ?

- It could be the case that, with respect to response, some of the levels of a predictor are similar. Then, we can fuse those levels.
- Which will reduce the number of levels of the categorical variable. i.e. number of dummy variables !

```
#Another problem !
district.cof <- coef(lm(rentm ~ 0 + area,data = rentData))
```



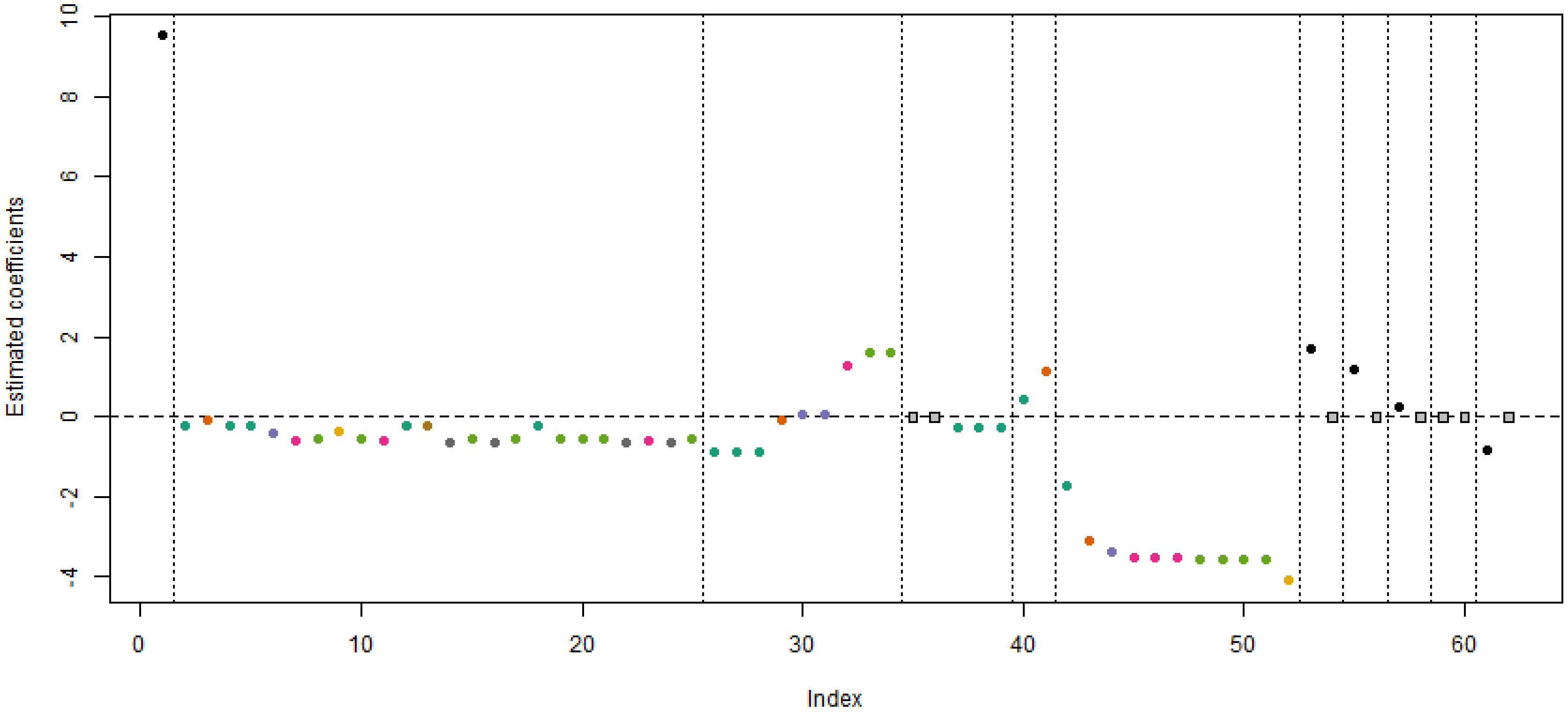
Using Proposed Method :

```r
library(smurf)  #To Implement Smurf Algo

my.formula <- rentm ~ p(area, pen = "gflasso") +
                p(year, pen = "flasso") + p(rooms, pen = "flasso") +
              p(quality, pen = "flasso") + p(size, pen = "flasso") +
                p(warm, pen = "lasso") + p(central, pen = "lasso") +
              p(tiles, pen = "lasso") + p(bathextra, pen = "lasso") +
                p(kitchen, pen = "lasso")

munich.fit <- glmsmurf(formula = my.formula, family = gaussian(), data = rentData,
                       pen.weights = "glm.stand", lambda = 0.015)
```

- Thus, we fitted the model with $\lambda = 0.015$. $glm.stand$ means that we have used standardized adaptive penalty weights based on an initial GLM fit.
- Note that, there is a CV based approach to find **Optimal** λ.

Using Proposed Method (Contd.) :

```
plot(munich.fit)
```



Using Proposed Method (Contd.) :

```
summary(munich.fit)
```

```
##
## Call:  glmsmurf(formula = my.formula, family = gaussian(), data = rentData,
##     lambda = 0.015, pen.weights = "glm.stand")
##
## Deviance residuals of estimated model:
##    Min. 1st Qu.  Median 3rd Qu.    Max.
## -6.5187 -1.2710 -0.0446  1.2723  7.8608
##
## Deviance residuals of re-estimated model:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -6.72931 -1.20614 -0.00988  1.27658  7.31087
##
##
## Coefficients:
```

Fused Levels of Categorical Variables :

Area

```
## [1] "area14" "area16" "area22" "area24"
## [1] "area7"  "area11" "area23"
## [1] "area8"  "area10" "area15" "area17" "area19" "area20" "area21" "area25"
## [1] "area6"
## [1] "area9"
## [1] "area13"
## [1] "area2"  "area4"  "area5"  "area12" "area18"
## [1] "area3"
```

Year of Construction

```
## [1] "year1920" "year1930" "year1940"
## [1] "year1950"
## [1] "year1960" "year1970"
## [1] "year1980"
## [1] "year1990" "year2000"
```

Fused Levels of Categorical Variables (Contd.):

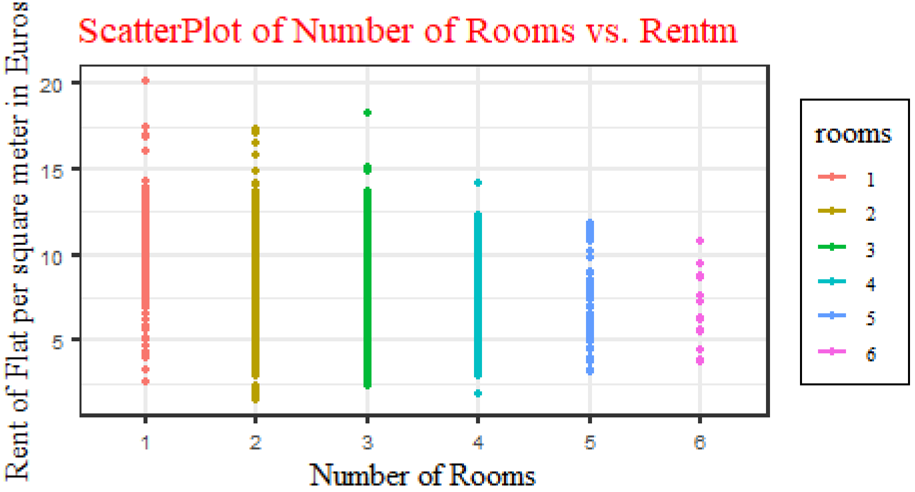
Size

```
## [1] "size130"
## [1] "size90"  "size100" "size110" "size120"
## [1] "size60" "size70" "size80"
## [1] "size50"
## [1] "size40"
## [1] "size30"
```

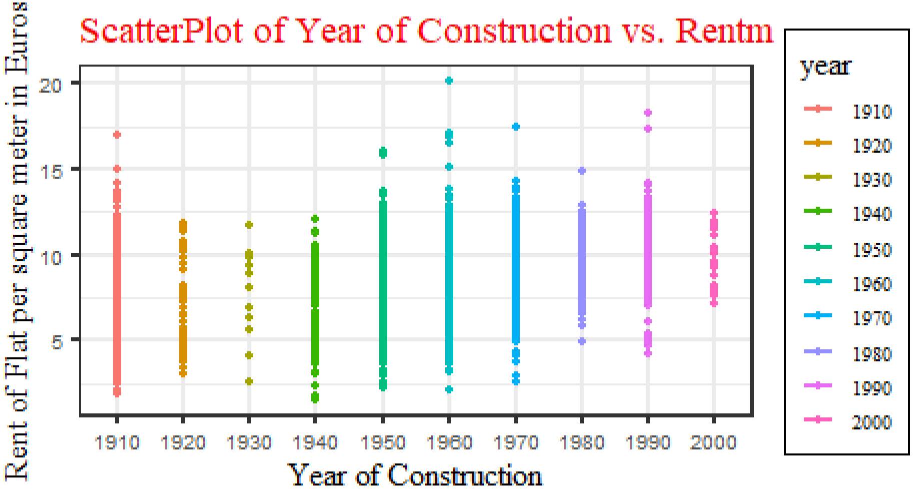
Number of Rooms

```
## [1] "rooms4" "rooms5" "rooms6"
## [1] "rooms2" "rooms3"
```

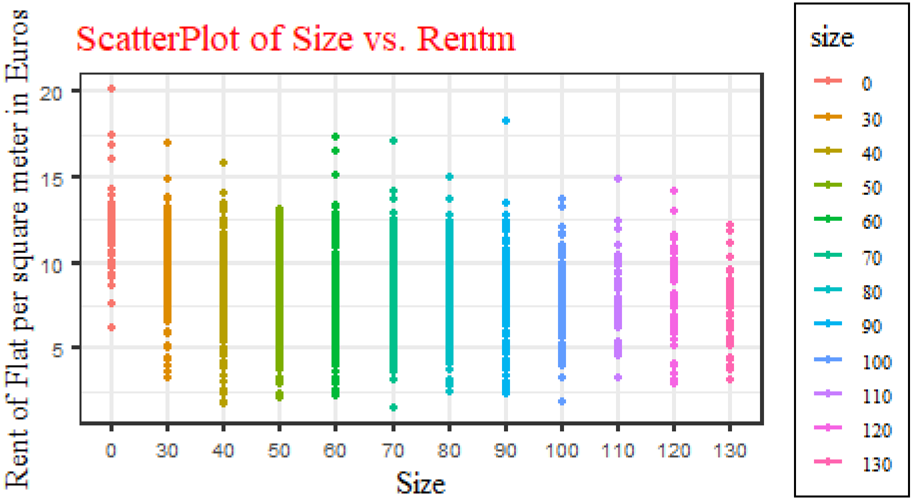
- This fusions are very much clear from EDA also !



ScatterPlot of Number of Rooms vs. Rentm



ScatterPlot of Year of Construction vs. Rentm



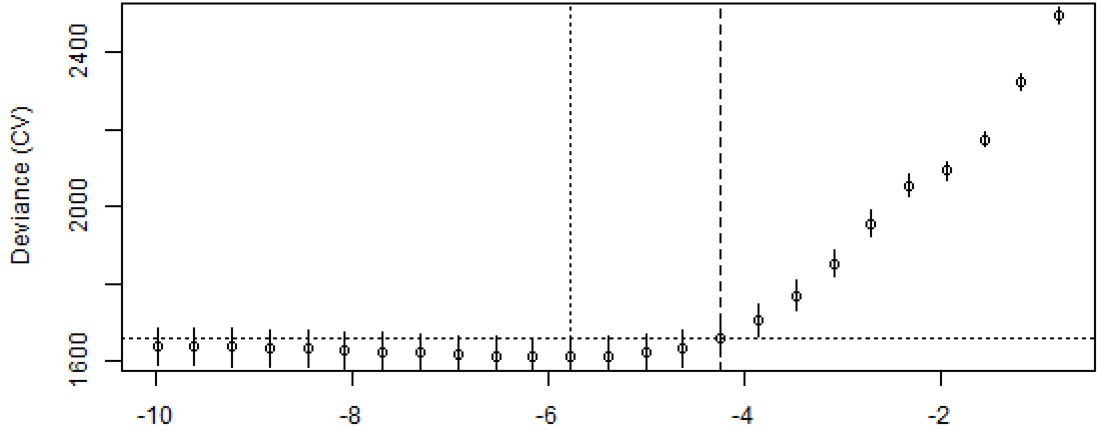
ScatterPlot of Size vs. Rentm

Finding Optimal Lambda :

```
munich.fit.cv <- glmsmurf(formula = my.formula, family = gaussian(), data = rentData,
                          pen.weights = "glm.stand", lambda = "cv1se.dev",
                          control = list(lambda.length = 25L))

munich.fit.cv$lambda
```

```
## [1] 0.01449709
```

```
plot_lambda(munich.fit.cv)
```



What if the Response is also Categorical $(m > 2)$?

- If response y is categorical variable having $m(> 2)$ number of categories. then, in general we use Multinomial Logistic Regression. In that case, to find estimates of the parameter maximize the log-likelihood or equivalently minimize negative log-likelihood.
- If $l(\beta|\mathbf{y};\mathbf{X})$ is the log-likelihood under the multinomial logistic model with parameter vector β as mentioned before. Then proceeding in a similar manner we have -

$$\hat{\beta} = \underset{\beta}{argmin} \left\{-l(\beta|\mathbf{y};\mathbf{X}) + \sum_{l=1}^{p} J_l(\beta_l)\right\}$$

Sparse Contingency Table :

- In the context of contingency tables, "sparse" refers to the situation where the frequency counts of one or more categories in the table are very small or zero.
- A contingency table is said to be sparse if the observed frequencies of some categories are much lower than expected based on the marginal totals.
- These type of contingency table occurs very frequently.
- The zeros in Sparse Contingency table occurs due to one of the two reasons -

  1. Random Zeros

  2. Structural Zeros

- If we have a 2×2 contingency table, then we cannot perform test of independence (Chi-Square Test) for Sparse Contingency Table.
- One solution could be to combine two or more than two levels of the categorical variables.
- But, if the categorical variables have large number of levels. Then, this becomes subjective thing to do.

References :

- JAN GERTHEISS1 AND GERHARD TUTZ : SPARSE MODELING OF CATEGORIAL EXPLANATORY VARIABLES
- Sander,Katrien,2,Tom,and Roel: Sparse regression with Multi-type Regularized Feature modeling
- Smurf Package

Thank You!