Chapter 5

Subset Selection in Poisson Regression

5.1. Introduction

In the previous chapters, we discussed variable selection in multiple linear regression. If the response is count, then the usual multiple linear regression is not appropriate. This is so, because estimated value of response from a linear regression model can be negative which is not allowed for a count response. If the distribution of response is a member of exponential family, generalized linear model is usually advocated. In this chapter, we consider the Poisson response model and develop a subset selection procedure.

Suppose response variable represents a count data of some relatively rare event, such as defects in a unit of manufactured product, bugs in software etc. In this situation, Poisson regression model is suitable for developing the relationship between the observed count and potentially useful regressors or regressor variables. Subset selection problem is of interest for linear regression model, giving the best subset from the set of regressors. Similarly, for Poisson regression model, one would like to obtain the best subset of regressors. In literature, very few subset selection methods in generalized linear regression models are available like Group least absolute shrinkage and selection operator (Meier, Sara and Buhlmann, 2008), Least Angle Regression (LARS) (Efron, Hastie, Johnstone and Tibshirani, 2004) which use a stepwise algorithm to select a parsimonious subset for prediction. Guo and Li (2002) illustrate that the Akaike information criterion (AIC) and Bayesian information criterion (BIC) work well for selecting the correct model in the error-in-variables Poisson regression.

In practice, number of regressors may be large; it is impractical to implement an exhaustive search method. It is reasonable to think of a stepwise procedure which is less time consuming and easy to implement.

Many researchers have proposed such stepwise procedures to select the best subset in linear regression set up. Some of which are due to, Thompson (1978a, b), Efroymson (1960), Yamashita, Yamashita and Kamimura, (2007), etc. In the present chapter, an effort is made to devise a criterion for subset selection based on the deviance. A stepwise procedure for subset selection has been provided.

This chapter is organized as follows. In Section 5.2, we take a brief review of Poisson regression model. In Section 5.3, a new criterion for subset selection in Poisson regression is proposed and some results on the same are obtained. In Section 5.4, the proposed method is illustrated with numerical examples. A stepwise procedure with illustration is presented in Section 5.5. Performance of the proposed method is also evaluated through simulation study in this section. Concluding remarks are given at the end of the chapter.

5.2. Poisson Regression Model and Some Basic Terms

The assumptions and model that encompass Poisson regression model are as follows. Let $Y_1, Y_2,\ldots,Y_n$ be necessarily independent Poisson variates with respective mean μ_i (i=1,2,…,n). Let $X'_i= [X_{i0}\ X_{i1}\ X_{i2}\ \ldots\ X_{ik}]$, $X_{i0} = 1$, i=1,2,…,n be the set of regressors. A set of regressors and μ_i are related via the model

$$\mu_i = \exp(X_i'\beta)\ , \qquad i = 1,2,\ldots,n \tag{5.2.1}$$

where $\beta = (\beta_0,\beta_1,\ldots.\beta_k)'$.

Parameter Estimation Method:

The log likelihood function for observed values $y_1, y_2,\ldots,y_n$ is given by (Myers, Montgomery and Vining, 2002)

$$\ln L(\beta, y) = \sum_{i=1}^{n} \left(-\exp(X_i'\beta) + y_i X_i'\beta - \ln(y_i!)\right)$$

Differentiating the above likelihood function with respect to β and equating it to zero, score equation can be obtained. The score function for the estimation of β is given by

$$\sum_{i=1}^{n}\left(y_i - \exp(X_i'\beta)\right)X_i = 0$$

$$\sum_{i=1}^{n}(y_i - \mu_i)X_i = 0$$

In matrix notation, we can write the above score function as,

$$X'(Y-\mu) = 0.$$

This is nonlinear in β. Thus, iterative reweighted least squares (IRLS) method can be used to obtain MLE of β (see Myers, Montgomery and Vining, 2002). The stepwise procedure for the IRLS on the Newton-Raphson method using log link is as follows:

Step I : Obtain an initial estimate of β, say $\hat{\beta}_0$ using ordinary least squares.

Step II : Use $\hat{\beta}_0$ to estimate $V = diag(1/\mu_i)$, $i = 1, \ldots, n$ and μ.

Step III: Let $\eta_0 = X\hat{\beta}_0$;

Step IV: Compute $z = \eta_0 + (Y - \hat{\mu})\frac{d\eta}{d\mu}$;

Step V: Obtain new estimate $\hat{\beta}_1 = (X'V^{-1}X)^{-1}X'V^{-1}z$ and iterate until some suitable convergence criterion is satisfied.

Deviance and Goodness of Fit:

First we define the deviance of the model.

Definition: Deviance: The deviance is defined as

$$D(\hat{\beta}) = -2\ln\left(\frac{L(\hat{\beta})}{L(\hat{\mu})}\right),$$

where, $L(\hat{\beta})$ is the likelihood of fitted model with β replaced by its MLE and $L(\hat{\mu})$ is the likelihood for saturated model in which n parameters $\mu_1, \mu_2, \ldots, \mu_n$ are estimated without the help of regressors.

In Poisson regression, deviance is given by

$$D(\hat{\beta}) = 2\sum y_i \ln\left(\frac{y_i}{\hat{\mu}_i}\right).$$

Many statistical tests are available in literature to test the goodness of fit of the model. One of these tests uses likelihood ratio criterion. Deviance

theoretically allows one to determine if the fitted regression model is significantly worse than the saturated model (Myers, Montgomery and Vining, 2002).

If the model is correct, model deviance $D(\hat{\beta})$ is asymptotically distributed as a chi-square variate with n-p degrees of freedom (Myers, Montgomery and Vining, 2002), where p indicates the number of parameters in the model. The smaller values of $D(\hat{\beta})$ will indicate that model fits well to the data while large values of $D(\hat{\beta})$ will be an indication of inadequacy of the model. Thus, a model is adequate or not can be determined based on the comparison of $D(\hat{\beta})$ (full set of regressors) with $D(\hat{\beta}_0)$ (without any regressors). Extending this idea further, we give a criterion for subset selection which is based on deviance $D(\hat{\beta})$ in Poisson regression.

5.3. Deviance Based Subset Selection Method

The full model is given by

$$E(Y)=\exp\{X\beta\} \tag{5.3.1}$$

We partition the $X_{(n \times k)}$ matrix as $\{X_{1(n \times p)} : X_{2(n \times k-p)}\}$ and $\beta_{(k \times 1)}$ as $\{\beta_{1(p \times 1)} : \beta_{2(k-p \times 1)}\}$.

Using the above partition, we rewrite the model in (5.3.1) as

$$E(Y)=\exp\{X_1\beta_1 + X_2\beta_2\}$$

Define the subset model 'A' as

$$E(Y)=\exp\{X_1\beta_1\}$$

The deviance for the model 'A' is given by

$$D(\hat{\beta}_A)=2\sum y_i \ln\left(\frac{y_i}{\hat{\mu}_{iA}}\right)$$

where, $\hat{\mu}_{iA}$ is MLE of μ_i based on model 'A'.

We compare the deviance $D(\hat{\beta}_A)$ with the deviance $D(\hat{\beta})$ corresponding to the full model by taking the difference

$$D(\hat{\beta}_A) - D(\hat{\beta}) \qquad (5.3.2)$$

Obviously, if the deviance of the subset model 'A' is close to the deviance of full model, then a subset model 'A' will be enough to describe the data. On the other hand, large value of (5.3.2) means that $D(\hat{\beta}_A)$ is far away from $D(\hat{\beta})$ and a subset 'A' is not as good as full model. If $D(\hat{\beta})$ is 'almost close' to the value of $D(\hat{\beta}_A)$, we can safely deploy the subset model for prediction purpose.

In the light of above discussion, we now define a criterion for subset selection based on deviance function.

Definition: D_p Statistic:

The statistic D_p is defined as

$$D_p = D(\hat{\beta}_A) - D(\hat{\beta}) - (k-2p), \qquad (5.3.3)$$

where k and p are the number of parameters in full model and subset model respectively, $D(\hat{\beta}_A)$ and $D(\hat{\beta}_A)$ are deviance of the full model and Subset model A respectively.

5.3.1. Distribution of Statistic D_p:

Consider the following testing of hypothesis problem:

$$H_0: \beta_2 = 0 \text{ Vs } H_1: \beta_2 \neq 0$$

Here, the null hypothesis is that the subset model is adequate for the prediction. The quantity $D(\hat{\beta}_A) - D(\hat{\beta})$ is always positive because deviance of subset model is always larger than that of full model. Moreover, $D(\hat{\beta}_A)$ has n-p degrees of freedom and $D(\hat{\beta})$ has n-k degrees of freedom. Hence, $D(\hat{\beta}_A) - D(\hat{\beta})$ has k-p degrees of freedom. If the null hypothesis is true and if n is large, the quantity $D(\hat{\beta}_A) - D(\hat{\beta})$ has a chi-square distribution with k-p degrees of freedom (Montgomery, Peck and Vining, 2006). To support this statement following simulation study is carried out.

Simulation Design:

A Poisson model comprising of four regressors X_1, X_2, X_3 and X_4 was considered. X_1 was generated from B(1, 0.4), and X_2, X_3 and X_4 from U(0,1). We define two different regression models. The first model is based on X_1 and X_2 while the second model is based on X_1, X_2 and X_3. Y_i's were generated from Poisson distribution with mean $\mu_i = e^{X_i'\beta}$, $i = 1, 2, ..., n$. . The specifications of the two regression models considered are as follows.

Model	Model Specification
I	μ =exp{5+X_1 + 2X_2}
II	μ = exp{5+X_1 + 2X_2 + 3X_3}

Three sets each consisting 100 samples corresponding to Model I and II were generated. In the first set, samples of size 20 were generated. In the second set, sample size was 50 and in the third set it was 100. We compute $D(\hat{\beta}_A) - D(\hat{\beta})$ for each model defined above, where $D(\hat{\beta})$ is the deviance for the model which consist of four regressors X_1, X_2, X_3 and X_4 and $D(\hat{\beta}_A)$ is the deviance for the subset model. The mean and variance of $D(\hat{\beta}_A) - D(\hat{\beta})$ are computed for each model and reported in the Table 5.3.1.

Table 5.3.1
Mean and variance of D_p statistic excluding constant

Model	Model I				Model II			
Sample size n	Sample Mean	Theoretical Mean	Sample Variance	Theoretical Variance	Sample Mean	Theoretical Mean	Sample Variance	Theoretical Variance
20	1.7576	2	2.2787	4	0.8894	1	1.1903	2
50	1.8659	2	3.2850	4	0.8963	1	1.8109	2
100	2.129	2	3.870	4	0.9591	1	2.1173	2

The chi-square test for goodness of fit was carried out for fitting of chi-square distribution with k-p d.f to $D(\hat{\beta}_A) - D(\hat{\beta})$. The results are as shown in the Table 5.3.2

Table 5.3.2
Chi-square test for goodness of fit

Model	Model I		Model II	
Sample size n	χ^2_{cal}	p-value	χ^2_{cal}	p-value
20	8.674597	0.122769	6.536736	0.257435
50	1.606772	0.952089	2.276621	0.517015
100	1.865771	0.867389	0.991026	0.911154

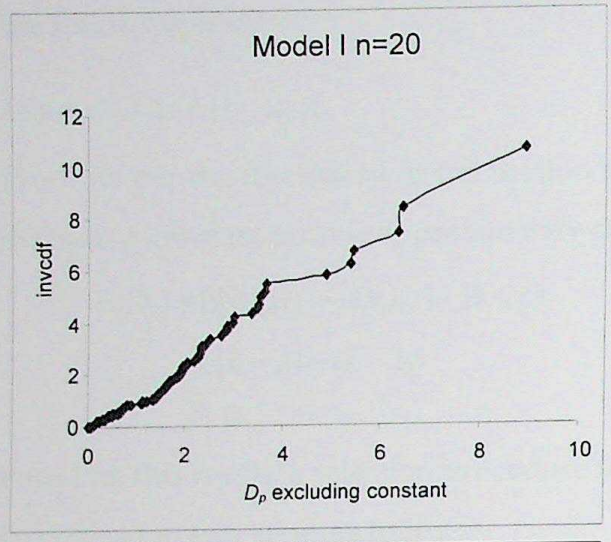
Table 5.3.2 reveals that the quantity $D(\hat{\beta}_A) - D(\hat{\beta})$ follows approximately chi-square distribution with k-p d.f.

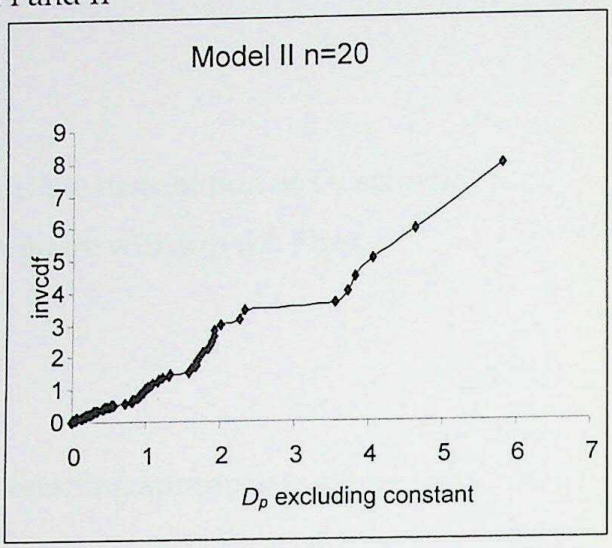
A simple graphical tool Q-Q plot (D'Agostino and Stephens, 1986) was used to assess distribution of $D(\hat{\beta}_A) - D(\hat{\beta})$. Q-Q plots were constructed for each sample and these are presented in Fig. 5.3.1.

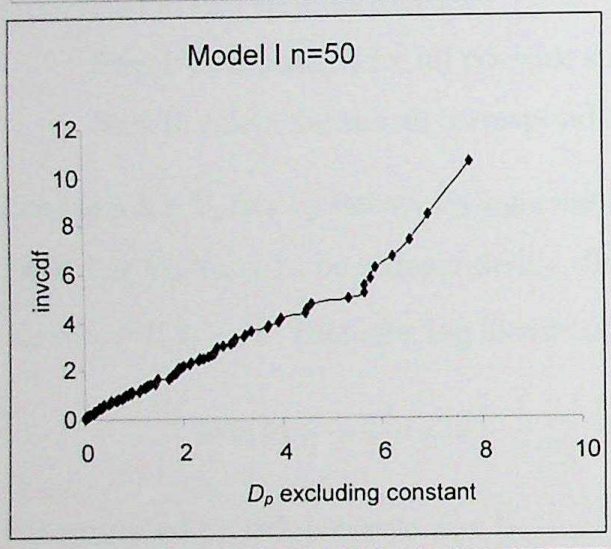
From Table 5.3.1, 5.3.2 and Q-Q plots, we see that the quantity $D(\hat{\beta}_A) - D(\hat{\beta})$ follows approximately chi-square distribution with k-p d.f. when sample size is large.

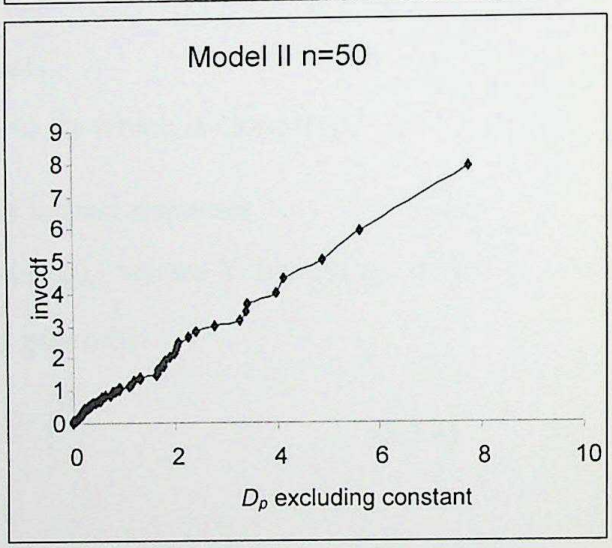
As a final comment, we observe from Table 5.3.1 and 5.3.2 as well as the Q-Q plots that for a sample size of 20 or less, the chi-square approximation is not as good as that for a larger sample size. Nevertheless, this is not a serious limitation of the proposed method as in real-life except possibly in situations like the medical data on rare diseases, sample size is invariably large (at least greater than 30 or so) and so the distribution of the Deviance can be closely approximated by chi-square distribution.

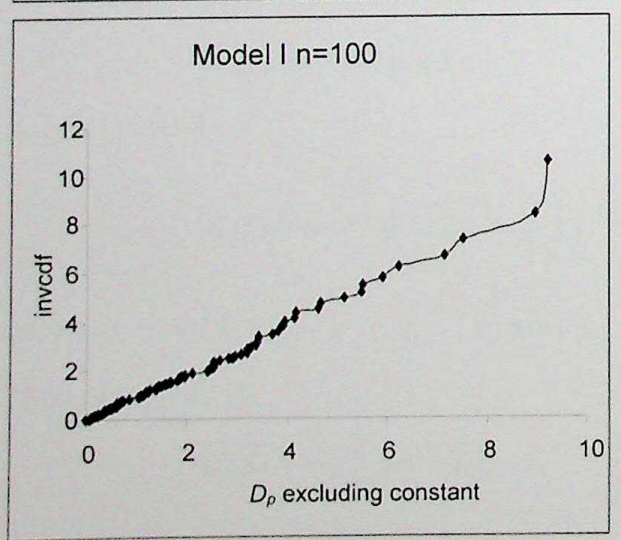
Fig. 5.3.1
Q-Q Plots for Model I and II

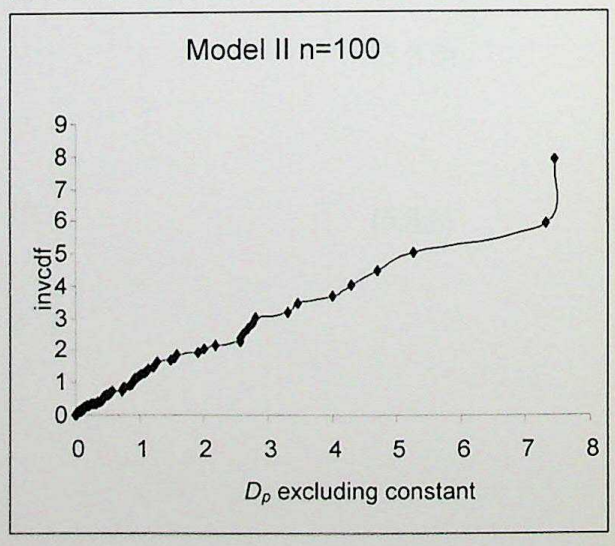












5.3.2. Some Properties of Statistic D_p:

We discuss below some of the relevant properties of D_p. To this end, we prove the following results:

Lemma 5.3.1: $E(D_p) \cong p$

Proof: As per the discussion in the Section 5.3.1, the distribution of D_p statistic excluding constant term is approximately chi-square with k-p d.f. Thus,

$$\begin{aligned} E(D_p) &= E[\, D(\hat{\beta}_A) - D(\hat{\beta}) \,] - (k-2p) \\ &\cong (k-p) - (k-2p) \\ &= p \end{aligned}$$

Based on this result, a selection procedure for selecting appropriate subset can be devised in two steps as follows:

Step I: Compute D_p for all possible subsets.

Step II: Select the subset corresponding to D_p which is close to p.

Lemma 5.3.2: *D_p and C_p statistic are equivalent for normal responses.*

Proof: Let $Y_1, Y_2, \ldots, Y_n$ be independently distributed, where Y_i has $N(\mu_i, \sigma^2)$ variate, i=1, 2, …, n. Then, the log likelihood is given by

$$\ln L(\mu) = -n \ln\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2}\sum_{i=1}^{n}\left(\frac{Y_i - \mu_i}{\sigma}\right)^2 \tag{5.3.4}$$

For saturated model, we have $\mu_i = Y_i$

$$\ln L(\mu) = -n \ln\left(\sigma\sqrt{2\pi}\right) \tag{5.3.5}$$

For full model,

$$\ln L(\hat{\beta}) = -n \ln\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{\mu}_i}{\sigma}\right)^2 \tag{5.3.6}$$

where $\hat{\mu}_i$ is MLE of μ_i, i= 1, 2, …, n, and $\hat{\mu}_i = X_i'\hat{\beta}$

Thus, deviance is

$$\begin{aligned} D(\hat{\beta}) &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \\ &= \frac{RSS_k}{\sigma^2} \end{aligned} \tag{5.3.7}$$

On the similar lines for subset model A, deviance under H_0 (subset model is an adequate) is given by

$$D(\hat{\beta}_A) = \frac{RSS_p}{\sigma^2} \tag{5.3.8}$$

Thus, D_p statistic is given by

$$D_p = \frac{RSS_p}{\sigma^2} - \frac{RSS_k}{\sigma^2} - (k - 2p) \tag{5.3.9}$$

As σ^2 is unknown, replace it by

$$\hat{\sigma}^2 = \frac{RSS_k}{n-k}$$

Thus, (5.3.9) becomes

$$D_p = \frac{RSS_p}{\hat{\sigma}^2} - (n-k) - (k-2p)$$

$$= \frac{RSS_p}{\hat{\sigma}^2} - (n-2p)$$

$$= C_p$$

5.4. Numerical Examples

In this Section, we demonstrate the performance of proposed procedure through numerical examples. In the Example 5.4.1, we have used real data set and in the Example 5.4.2, simulated data is used.

Example 5.4.1: In this example, the mining data (Myers, Montgomery and Vining, 2002, pp. 142) is used. In the dataset, counts were observed on the number of injuries or fractures that occur in upper seam of mines in the coal fields of the Appalachian region in the Western Virginia. The variables include; X_1: Inner burden thickness in feet, X_2: Percent extraction of the lower previously mined seam, X_3: Lower seam height and X_4: Time that the mine has been opened.

We define for sake of completeness the AIC criterion (Akaike, 1974). It is given by

$$AIC = -2L + 2p,$$

where L is the maximum log-likelihood and p is the number of parameters in the fitted model. The D_p statistic and AIC criterion are applied to the data. The results are presented in the following table.

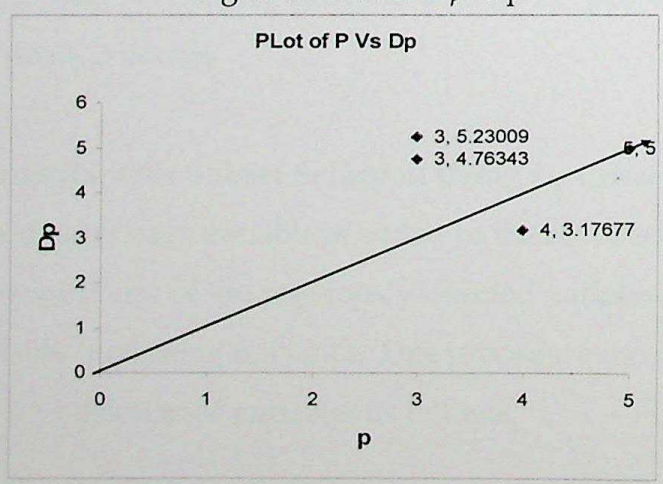
Table 5.4.1.
D_p and AIC for Mining Data

Model	AIC	D_p
X_1	172.1114	32.97601
X_2	148.8919	9.75651
X_3	174.1132	34.9778
X_4	172.3352	33.19984
X_1X_2	144.3654	5.23009
X_1X_3	173.3420	34.20666
X_1X_4	172.6622	33.52680
X_2X_3	149.8588	10.72348
X_2X_4	143.8988	4.76343
X_3X_4	173.4963	34.36098
$X_1X_2X_3$	146.0221	6.88675
$X_1X_2X_4$	**142.3121**	**3.17677**
$X_1X_3X_4$	174.0410	34.90568
$X_2X_3X_4$	145.2918	6.156402
$X_1X_2X_3X_4$	144.1354	5

It is clear from Table 5.4.1, these two criteria pick up the same subset namely $\{X_1, X_2, X_4\}$.

Alternatively, one can plot the values of D_p against p and select the subset corresponding to D_p which is close to the line $D_p = p$.

Fig. 5.4.1 Plot of D_pVs p



In the Fig. 5.4.1, the point (4, 3.17677) is close to the line D_p = p. Thus, the appropriate subset is the subset corresponding to D_p = 3.17677. Hence, the best model is based on X_1, X_2 and X_4.

Example 5.4.2: In this example, simulated data was used to study the performance of the proposed criterion. 30 observations on X_1 were generated from B(1, 0.4) and those on X_2 and X_3 from U(0,1). Y_i's were generated from Poisson distribution with mean $\mu_i = \exp\{2+2X_{1i} + X_{2i}\}$, i=1,2,…,n. D_p and AIC were computed and are reported in the following table.

Table 5.4.2.
D_p and AIC for simulated data.

Model	X_1	X_2	X_3	X_1X_2	X_1X_3	X_2X_3	$X_1X_2X_3$
AIC	278.5612	1127.0480	1092.1680	**182.6249**	279.5561	1078.4830	183.5106
D_p	99.0506	947.5328	912.6572	**3.1143**	100.0455	898.9727	4

The results of Table 5.4.2 reveal that both the methods pick up the correct subset namely, $\{X_1, X_2\}$. From the above examples, we conclude that the performance of proposed method and AIC is same.

As discussed earlier, in applications, we may have a large number of regressors and it is tedious to compute D_p for all possible $2^k - 1$ subsets of regressors. In the next section we propose a stepwise procedure for subset selection based on D_p criterion.

5.5. Stepwise Procedure for Subset Selection Using D_p Criterion

In this method, after each variable is added to the set of selected variables, a test is made to see if any of the previously selected variables can be deleted without appreciably increasing deviance. This procedure incorporates criteria for the addition and deletion of variables as follows.

a. Addition Criterion

Let $D(\hat{\beta})_p$ be the deviance with p-1 variables. Suppose the smallest deviance which can be obtained by adding another variable to the present set is $D(\hat{\beta})_{p+1}$. The ratio

$$R_1 = \frac{D(\hat{\beta})_p - D(\hat{\beta})_{p+1}}{D(\hat{\beta})_{p+1}/(n-p-1)} \quad . \tag{5.5.1}$$

is calculated and compared with $F_{1, n-p-1, \alpha}$; the upper α^{th} quantile of F-variate with 1 and n-p-1 degrees of freedom. If R_1 is greater than $F_{1, n-p-1, \alpha}$, then the corresponding variable is added to the selected subset of variables.

b. Deletion Criterion

With p - 1 variable in the selected subset, let $D(\hat{\beta})_{p-1}$ be the smallest deviance which can be obtained after deleting any variable from the previously selected variables. The ratio

$$R_2 = \frac{D(\hat{\beta})_{p-1} - D(\hat{\beta})_p}{D(\hat{\beta})_p/(n-p)} \tag{5.5.2}$$

is calculated and compared with an $F_{1, n-p, \alpha}$. If R_2 is less than $F_{1, n-p, \alpha}$, then the corresponding variable is deleted from the selected subset model.

Based on above addition and deletion criteria, we give an algorithm for selecting the appropriate subset.

c. Algorithm

Let S denotes the set of regressors included in the subset model. Initially, let $S = \{X_0\}$. The stepwise procedure is as follows.

Step I.

Regress Y on all the variables and compute deviance of the full model.

Step II.

Regress Y on each variable and compute D_p statistic. Add the variable to S that corresponds to minimum D_p. Suppose i^{th} variable is selected at this step. Then $S=\{X_0, X_i\}$.

Step III.

Regress response variable Y on previously selected variables and single variable from set of unselected variables. Obtain D_p statistic. Repeat the same for all unselected variables. Select that subset which corresponds to minimum D_p. For this subset, compute R_1. If $R_1 > F_{1, n-p-1, \alpha}$, include this variable in S and go to next step otherwise terminate.

Step IV.

If number of variables in selected model in step III is greater than two then go to the next step otherwise repeat step III.

Step V.

Regress response variable Y on currently selected variable and eliminating single variable from set of selected variables and obtain D_p statistic. Repeat this for all previously selected variables. Compute R_2 for the subset corresponding to minimum D_p. If $R_2 < F_{1, n-p, \alpha}$ then remove that variable from S otherwise go to next step.

Step VI.

Repeat Steps III to V until no addition is possible or all the variables are included in S.

5.5.1. Illustration

Let us illustrate the above stepwise procedure by applying it to the simulated data comprising of eight regressors generated from U(0, 1). Sample size was 50. The response variable Y was generated from Poisson distribution with mean $\mu = \exp\{2+X_1+X_2+X_3+2X_4+2X_5\}$. The results are summarized in Table 5.5.1.The total number of all possible subsets in this case is 255. The number of subsets analyzed to arrive at the decision is 42. We see that this method reduces computations to a greater extent (approximately 84%).

Table 5.5.1
Stepwise procedure for subset selection

Step	Minimum D_p	R_1	R_2	F_e	F_d	S	Decision
I	{4}* 12780.53	-	-	-	-	{4}	Continue
II	{4, 5} 4866.836	11.461	-	4.0471	-	{4, 5}	Continue
III	{3, 4, 5} 3140.985	24.9473	{3, 4} 102.8366	4.05179	4.05179	{3, 4, 5}	Continue
IV	{2, 3, 4, 5} 1785.858	33.3923	{2, 4, 5} 3211.631	4.05661	4.05661	{2, 3, 4, 5}	Continue
V	{1, 2, 3, 4, 5} **6.062996**	1666.76	{1, 2, 3,5} 1490.833	4.06170	4.06170	{1, 2, 3, 4, 5}	Continue
VI	{1, 2, 3, 4, 5, 6} 6.684952	1.07248	-	4.06704	-	{1, 2, 3, 4, 5}	No addition. Terminate

* { i, j, k, … } are the subscripts of X_i, X_j, X_k, ….

From Table 5.5.1, we see that stepwise procedure and D_p statistic select a proper subset namelyX_1, X_2, X_3, X_4, and X_5.

Model selection ability of the criterion D_p is demonstrated by using simulation. In this simulation study we consider two simulation designs as follows:

A: All the Regressors Generated from Uniform distribution

In this study, 500 random samples of size n on regressors X_1, X_2, X_3, X_4 and X_5 were generated from U(0, 1). Y_i's were generated from Poisson distribution with mean μ_i as specified below. The D_p criterion was applied to each sample to select the appropriate subset. Thereafter, number of times the criterion selects the correct model was counted. The same procedure was repeated for five different models specified in Table 5.5.2 and for sizes 10, 20 and 50. The model selection ability is presented in the Table 5.5.3.

Table 5.5.2
Submodel Specification

Model	Submodel Specification
I	$\mu = \exp\{2+2X_1 + X_2\}$
II	$\mu = \exp\{0.5+X_1 + 2X_2\}$
III	$\mu = \exp\{1+0.5X_1 + X_3 + X_4\}$
IV	$\mu = \exp\{1+X_1 + 1.5X_4\}$
V	$\mu = \exp\{X_1 + X_3\}$

Table 5.5.3
Model selection ability (%) of D_p

Sample size n	10		20		50	
Model	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
I	77	23	76	24	68	32
II	73	27	68	32	71	29
III	74	26	81	19	80	20
IV	78	22	79	21	69	31
V	69	31	83	17	70	30

B: All the Regressors Generated from Different Distributions

The same design in case (A) is followed except only four regressors are considered. X_1 was generated from B(1, 0.4), X_2 was generated from U(0, 1), X_3 was generated from N(0, 1) and X_4 was generated from Poisson(1). Subset model specifications used were same as that given in Table 5.5.2. Table 5.5.4 demonstrates the model selection ability of D_p.

Table 5.5.4
Model selection ability (%) of D_p

Sample size n	10		20		50	
Model	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
I	80	20	78	22	74	26
II	77	23	82	18	76	24
III	85	15	84	16	89	11
IV	82	18	82	18	81	19
V	80	20	83	17	82	18

It is clear that the performance of the proposed method is good irrespective of sub model specification and sample size (n). Further, we have

also verified that the performance of stepwise procedure is same as that of D_p statistic.

5.6. Concluding Remarks

The subset selection method introduced here is based on deviance and it is simple to implement. The performance of the proposed method to select appropriate subset model is more than 70%. The prominent feature of the proposed criterion D_p is to devise a stepwise procedure for subset selection. This procedure helps in reducing the time and cost for computational purpose when large number of variables are in the model. We have implemented the stepwise procedure using R software. The code is given in the appendix. A part of this Chapter is published as an article entitled "Subset Selection in Poisson Regression" in Journal of Statistical Theory and Practice (2011), Volume 5, Number 2, Page No. 207- 219.

Matlab Code for JRM Estimator:

Matlab codes for computation of the smaller MSE frequency and Average MSE of OLS, ORR, OJR, M- and JRM estimator.

```
clc
clear all
fid=fopen('OUTPUT1.xls', 'w');
for l=1:4
    p=4;
    samplesiz=[20 30 50 100];
    rowvector=[0.9 0.99 0.999];
    errsd=[1 2 3 10 20 30];
    for j=1:length(rowvector)
        rowvect=rowvector(j);
        for k=1:length(errsd)
            ERRSD=errsd(k);
            sumb=0; sumb1=0; sumb2=0; sumb3=0; sumb4=0; sumb5=0;
            MSEOLS=0; MSEORR=0; MSEOJR=0; MSEMES=0; MSEJRM=0;
            n=samplesiz(l)
            ro=rowvect
            EV=ERRSD          % error standard deviation
            itr=10000;
            clear x
            clear y
            clear s
            for i=1:itr
                i
                xnorm=normrnd(0,1,n,p+1);
                for i=1:p
                x(:,i)=sqrt((1-ro*ro))*xnorm(:,i)+ro*xnorm(:,p+1);
                end
                x;
                X=[ones(n,1) x];
                error=normrnd(0,EV,n,1);
                B=[15 4 8 2 5]';
                y=X*B+error;
                y111=y;
                b1=inv(X'*X)*X'*y;
                yhat1=X*b1;
                err=(y-yhat1);
                nn=abs(err);
                T= sort(nn);
                for i= 1: n
                if abs(y(i)-yhat1(i))==T(n)
                    y1(i)=20*y(i);
                else
                    y1(i)=1*y(i);
                end
                if abs(y(i)-yhat1(i))==T(n-1)
                    y11(i)=20*y1(i);
                else
                    y11(i)=1*y1(i);
                end
                if abs(y(i)-yhat1(i))==T(n-2)
```

```
    y2(i)=20*y11(i);
else
    y2(i)=1*y11(i);
end
end
y=y1';
%y=y11';
%y=y2';
siz=size(x);

% Conversion to Standardized and canonical form
mx=mean(x);
sx=sqrt(n-1)*std(x);
for i=1:p
    s(:,i)=(x(:,i)-mx(i)*ones(n,1))*inv(sx(i));
end
s;
S=[ones(n,1), s];    % standardized design matrix
bs=inv(S'*S)*S'*y;
[Q,u]=eig(s'*s);
z=s*Q;

% Computation of OLS estimator
Z=[ones(n,1), z];    % Canonical form of design matrix
a1=inv(Z'*Z)*Z'*y;   % OLS in canonical form
a=inv(z'*z)*z'*y;
yhat=Z*a1;
OLS1=Q*a;
OLS2=OLS1./sx';
OLS3=a1(1)-mx*OLS2;
OLS=[OLS3 OLS2']';   % OLS estimator

% Computation of M-estimator
WW=eye(n,n);
delta=0.000001;
max1=1.00;
while max1 > delta
    bz=inv(Z'*WW*Z)*Z'*WW*y;
    yhatz=Z*bz;
    r=y-yhatz;
    k1=1.4826*(median(abs(r-median(r))));
    F2=r/k1;
    c=1.345*ones(n,1);
    F1=min(F2,c);
    S1=-min(c,-F1);
    Wj=S1./F2;
    WW=diag(Wj);
    bm=inv(Z'*WW*Z)*Z'*WW*y;
    max1=max(abs(bm-bz));
end
bm;
am=bm(2:p+1);
MES1=Q*am;
MES2=MES1./sx';
MES3=bm(1)-mx*MES2;
MES=[MES3 MES2']';

% Computation of ridge parameter
yhatbm=Z*bm;
e=y-yhatbm;
```

```
M=1.4826*median(abs(e-median(e)));
k=p*(M*M)/(am'*am);                    % Ridge parameter

% Computation of ORR estimator
I=eye(p,p);
R=k*I;
A=u+R;
azORR=(I-R*inv(A))*a;
ORR1=Q*azORR;
ORR2=ORR1./sx';
ORR3=a1(1)-mx*ORR2;
ORR=[ORR3 ORR2']';                     % ORR estimator

% Computation of OJR estimator
azOJR=(I-R*R*inv(A*A))*a;
OJR1=Q*azOJR;
OJR2=OJR1./sx';
OJR3=a1(1)-mx*OJR2;
OJR=[OJR3 OJR2']';                     % OJR estimator

% Computation of Jackknifed Ridge M-estimator (JRM)
azJRM=(I-R*R*inv(A*A))*am;
JRM1=Q*azJRM;
JRM2=JRM1./sx';
JRM3=bm(1)-mx*JRM2;
JRM=[JRM3 JRM2']';                     % JRM estimator

% Computation of MSE of various estimators
MseJRM=(JRM-B)'*(JRM-B);               % MSE of JRM estimator
MseMES=(MES-B)'*(MES-B);               % MSE of M - estimator
MseOJR=(OJR-B)'*(OJR-B);               % MSE of JRR estimator
MseORR=(ORR-B)'*(ORR-B);               % MSE of ORR estimator
MseOLS=(OLS-B)'*(OLS-B);               % MSE of OLS estimator

MSEJRM=MSEJRM+MseJRM;
MSEMES=MSEMES+MseMES;
MSEOJR=MSEOJR+MseOJR;
MSEORR=MSEORR+MseORR;
MSEOLS=MSEOLS+MseOLS;

% Comparison of MSE of various estimators
MSEVECT=[MseJRM MseMES MseOJR MseORR MseOLS];
MIN=min(MSEVECT);
if MIN==MseJRM
    Sb1=1;
else
    Sb1=0;
end
if MIN==MseMES
    Sb2=1;
else
    Sb2=0;
end
if MIN==MseOJR
    Sb3=1;
else
    Sb3=0;
end
if MIN==MseORR
    Sb4=1;
```

```
                    else
                        Sb4=0;
                    end
                    if MIN==MseOLS
                        Sb5=1;
                    else
                        Sb5=0;
                    end

                    sumb1=sumb1+Sb1;
                    sumb2=sumb2+Sb2;
                    sumb3=sumb3+Sb3;
                    sumb4=sumb4+Sb4;
                    sumb5=sumb5+Sb5;
                end
                MJRM=MSEJRM/itr;
                MMES=MSEMES/itr;
                MOJR=MSEOJR/itr;
                MORR=MSEORR/itr;
                MOLS=MSEOLS/itr;
                sumb1;
                sumb2;
                sumb3;
                sumb4;
                sumb5;
                fprintf(fid,'%1.4f\t %1.4f\t %1.4f\t %1.4f\t %1.4f\t
                %1.4f\t %1.4f\t %1.4f\t %1.4f\t %1.4f\t\n', MJRM, MMES,
                MOJR, MORR, MOLS, sumb1, sumb2, sumb3, sumb4, sumb5);
            end
        end
    end
fclose(fid);
```

Matlab Code for computation of C_p, S_p, R_p and GS_p Statistics:

Matlab Code for computation of C_p, S_p, R_p and GS_p for all possible subset models.

```
clc
clear all
x=[          ];
y=[          ];
siz=size(x);
n=siz(1);
m=siz(2);
X=[ones(n,1) x];
bl=inv(X'*X)*X'*y;
yhatolsout=X*bl;
errout=(y-yhatolsout);
nn=abs(errout);
T= max(nn);
for i= 1: n
    if abs(y(i)-yhatolsout(i))==T
       y1(i)=100;
    else
       y1(i)=1*y(i);
    end
end
%y=y1';

%%%%%%%%%%%%% for Cp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

b=inv(X'*X)*X'*y;         % OLS estimator
yhatols=X*b;
err=(y-yhatols);
mrss=err'*err/(n-m-1);

%%%%%%%%%%% subset Selection using Cp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u=0;
for i=1:m
    h=nchoosek(m, i);
    B=[1:m];
    k111=nchoosek(B, i);
    for j=1:h
        u=u+1;
        ss=k111(j,:);
        s1=x(:,ss);
        o=ones(n,1);
        S1=[o s1];
        D1=size(S1);
        p=D1(2);
        bs=inv(S1'*S1)*S1'*y;
        yhatolss=S1*bs;
        errs=(y-yhatolss);
        mrsss=errs'*errs/(n-p);
        Cp(u)=errs'*errs/mrss-(n-2*p);
```

```
    end
end

%%%%%%%%%%%%% for Sp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

WW=eye(n,n);
delta=0.000001;
max1=1.00;
while max1 > delta
      bm=inv(X'*WW*X)*X'*WW*y;
      yhatm=X*bm;
      r=y-yhatm;
      k1=1.4826*(median(abs(r-median(r))));
      F2=r/k1;
      c=1.345*ones(n,1);
      F1=min(F2,c);
      S1=-min(c,-F1);
      Wj=S1./F2;
      WW=diag(Wj);
      bmf=inv(X'*WW*X)*X'*WW*y;
      max1=max(abs(bmf-bm));
end
bmf;                        % M-estimator
am=bmf(2:m+1);
yhatm=X*bmf;
em=y-yhatm;
M=1.4826*median(abs(em-median(em)));

%%%%%%%%%%% subset Selection using Sp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u=0;
for i=1:m
    h=nchoosek(m, i);
    B=[1:m];
    k111=nchoosek(B, i);
    for j=1:h
        u=u+1;
        ss=k111(j,:);
        s1=x(:,ss);
        o=ones(n,1);
        S1=[o s1];
        D1=size(S1);
        p=D1(2);
        bs=inv(S1'*S1)*S1'*y;
        WW=eye(n,n);
        delta=0.000001;
        max1=1.00;
        while max1 > delta
            bms=inv(S1'*WW*S1)*S1'*WW*y;
            yhatms=S1*bms;
            rs=y-yhatms;
            ks=1.4826*(median(abs(rs-median(rs))));
            F2=r/ks;
            c=1.345*ones(n,1);
            F1=min(F2,c);
            SS1=-min(c,-F1);
            Wj=SS1./F2;
            WW=diag(Wj);
            bmfs=inv(S1'*WW*S1)*S1'*WW*y;
```

```
            max1=max(abs(bmfs-bms));
        end
        bmfs;
        yhatms=S1*bmfs;
        Sp(u)=(yhatm-yhatms)'*(yhatm-yhatms)/M^2-(m+1-2*p);
    end
end

%%%%%%%%%%%% for Rp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mx=mean(x);
sx=sqrt(n-1)*std(x);
for i=1:m
    s(:,i)=(x(:,i)-mx(i)*ones(n,1))*inv(sx(i));
end
s;
S=[ones(n,1), s];        % standardized design matrix
bst=inv(S'*S)*S'*y;
bst1=bst(2:m+1);

% OLS and ridge parameter estimation

yhatst=S*bst;
rssst=(y-yhatst)'*(y-yhatst);
mrssst=rssst/(n-m-1);
kr=m*(mrssst)/(bst1'*bst1);
Id=eye(m+1,m+1);
br=inv(S'*S+kr*Id)*S'*y;
bro1=inv(diag(sx))*br(2:m+1);
bro2=mx./sx;
bro3=bro2*br(2:m+1);
bro4=br(1)-bro3;
bro=[bro4 bro1']';
yhatstr=X*bro;
rssstr=(y-yhatstr)'*(y-yhatstr);
mrssstr=rssstr/(n-m-1);
Id1=eye(m,m);
Hr=s*inv(s'*s+kr*Id1)*s';
THr=trace(Hr'*Hr);

%%%%%%%%%%% subset Selection using Rp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u=0;
for i=1:m
    h=nchoosek(m, i);
    B=[1:m];
    k111=nchoosek(B, i);
    for j=1:h
        u=u+1;
        ss=k111(j,:);
        s1=s(:,ss);
        o=ones(n,1);
        S1=[o s1];
        D1=size(S1);
        p=D1(2);
        bsts=inv(S1'*S1)*S1'*y;
        bsts1=bsts(2:p);
        yhatsts=S1*bsts;
        rsssts=(y-yhatsts)'*(y-yhatsts);
```

```
            mrsssts=rsssts/(n-m-1);
            krs=m*(mrsssts)/(bsts1'*bsts1);
            Id=eye(p,p);
            brs=inv(S1'*S1+krs*Id)*S1'*y;
            mx1=mx(ss);
            sx1=sx(ss);
            bro1=inv(diag(sx1))*brs(2:p);
            bro2=mx1./sx1;
            bro3=bro2*brs(2:p);
            bro4=brs(1)-bro3;
            bros=[bro4 bro1']';
            Xs1=x(:,ss);
            Xs=[o Xs1];
            yhatstrs=Xs*bros;
            Id2=eye(p-1,p-1);
            Hrs=s1*inv(s1'*s1+krs*Id2)*s1';
            THrs=trace(Hrs'*Hrs);
            Rp(u)=(yhatstr-yhatstrs)'*(yhatstr-yhatstrs)/mrssstr-
            THr+THrs+p;
        end
    end

%%%%%%%%%%%%% for GSp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mx=mean(x);
sx=sqrt(n-1)*std(x);
for i=1:m
    s(:,i)=(x(:,i)-mx(i)*ones(n,1))*inv(sx(i));
end
s;
S=[ones(n,1), s];        % Standardized design matrix
bs=inv(S'*S)*S'*y;
[Q,u]=eig(s'*s);
z=s*Q;

% lse and ridge parameter estimation

Z=[ones(n,1), z];        % Canonical form of design matrix
a1=inv(Z'*Z)*Z'*y;       % OLS in canonical form
a=a1(2:m+1);
yhato=Z*a1;
rss=(y-yhato)'*(y-yhato);
mrss=rss/(n-m-1);

% Estimation of M-estimator

WW=eye(n,n);
delta=0.000001;
max1=1000000.00;
while max1 > delta
      bz=inv(Z'*WW*Z)*Z'*WW*y;
      yhatz=Z*bz;
      r=y-yhatz;
      k1=1.4826*(median(abs(r-median(r))));
      F2=r/k1;
      c=1.345*ones(n,1);
      F1=min(F2,c);
      S111=-min(c,-F1);
      Wj=S111./F2;
```

```
      WW=diag(Wj);    .
      bm=inv(Z'*WW*Z)*Z'*WW*y;
      max1=max(abs(bm-bz));
end
bm;                          % M-estimator
am=bm(2:m+1);
yhat=Z*bm;
e=y-yhat;
M=1.4826*median(abs(e-median(e)));

% ridge parameter by M-estimator

k=m*(M*M)/(am'*am);

% Computation of Jackknifed Ridgetype M-estimator (JRM)

I=eye(4,4);
R=k*I;
A=u+R;
delta=(I-R*R*inv(A*A));
azJRM=(I-R*R*inv(A*A))*am;
aJRM=[bm(1) azJRM']';
betaJRM=aJRM;                % JRM estimator
yhatJRM=Z*betaJRM;
eJRM=y-yhatJRM;
T=(WW*z*inv(z'*WW*z)*delta*z');
Ts=trace(T'*T);

% Estimators of scale parameter

sigma2=1.4826*median(abs(eJRM-median(eJRM)));
scaledeJRM=eJRM/sigma2;
for i= 1:n
    if abs(scaledeJRM(i))<1.345
        sigma71(i)=1;
    else
        sigma71(i)=0;
    end
end
sigma72=sum(sigma71);
sigma73=1+((m+1)*(1-sigma72))/(n*sigma72);

for i= 1:n
    if abs(scaledeJRM(i))<=1.345
        sigma741(i)=1;
    else
        sigma741(i)=0;
    end
end
sigma742=(sum(sigma741)/n)^2;
sigma74=(n-m-1)*sigma742;
for i= 1:n
    if abs(scaledeJRM(i))<1.345
        sigma751(i)=(scaledeJRM(i))^2;
    else
        sigma751(i)=0;
    end
end
sigma75=(sum(sigma751))^2*sigma2^2*sigma73^2;
```


BARR. BALASAHEB KHARDEKAR LIBRARY
SHIVAJI UNIVERSITY, KOLHAPUR.

```
sigma7=sigma75/sigma74;

%%%%%%%%%%% subset Selection using GSp %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u=0;
for i=1:m
    h=nchoosek(m, i);
    B=[1:m];
    k111=nchoosek(B, i);
    for j=1:h
        u=u+1;
        ss=k111(j,:);
        s1=s(:,ss);
        o=ones(n,1);
        S1=[o s1];
        bs=inv(S1'*S1)*S1'*y;
        [Q1,u1]=eig(s1'*s1);
        z1=s1*Q1;
        Z1=[o z1];
        D1=size(S1);
        p=D1(2);
        a21=inv(Z1'*Z1)*Z1'*y;
        a2=a21(2:p);
        yhato2=Z1*a21;
        rss2=(y-yhato2)'*(y-yhato2);
        mrss2=rss2/(n-p);
        WW1=eye(n,n);
        delta1=0.000001;
        max11=1000000.00;
        while max11 > delta1
            bz1=inv(Z1'*WW1*Z1)*Z1'*WW1*y;
            yhatz1=Z1*bz1;
            r1=y-yhatz1;
            k1=1.4826*(median(abs(r1-median(r1))));
            F2=r/k1;
            c=1.345*ones(n,1);
            F1=min(F2,c);
            S121=-min(c,-F1);
            Wj=S121./F2;
            WW1=diag(Wj);
            bm1=inv(Z1'*WW1*Z1)*Z1'*WW1*y;
            max11=max(abs(bm1-bz1));
        end
        bm1;
        am1=bm1(2:p);
        yhat12=Z1*bm1;
        e1=y-yhat12;
        M11=1.4826*median(abs(e1-median(e1)));
        mse11=(y-yhat12)'*(y-yhat12)/(n-m-1);
        MM1=1.4826*median(abs(e1));
        I=eye(p-1,p-1);
        ks=(p-1)*(M11*M11)/(am1'*am1);
        R1=ks*I;
        u12=u1*ones(p-1,1);
        u2=u12(1:p-1);
        u3=diag(u2);
        A1=u3+R1;
        azJRM1=(I-R1*R1*inv(A1*A1))*am1;
        aJRM1=[bm1(1) azJRM1']';
```

```
        delta1=(I-R1*R1*inv(A1*A1));
        T1=(WW1*z1*inv(z1'*WW1*z1)*delta1*z1');
        Ts1=trace(T1'*T1);
        beta1=aJRM1;
        yhatJRM1=Z1*beta1;
        eJRM1=y-yhatJRM1;
        GSp(u)=( yhatJRM1- yhatJRM)'*( yhatJRM1-
        yhatJRM)/(sigma2^2)-Ts+Ts1+p;
    end
end
[Cp' Sp' Rp' GSp']
```

R Code for Subset Selection when Error follows Non-normal Distribution:

(Standard Cauchy Distribution when n = 30)

```
Q<-500
for (i in 1:Q)
{
print(i)
N<-30
m<-4
errorN1<-rnorm(N, mean=0, sd= 1)
errorN2<-rnorm(N, mean=0, sd= 1)
errorC<-errorN1/errorN2
x1<-runif(N, min=0, max=1)
x2<-runif(N, min=0, max=1)
x3<-runif(N, min=0, max=1)
x4<-runif(N, min=0, max=1)
x0<-as.matrix(rep(1,N),nrow=N,ncol=1)
b11<-as.matrix(c(15, 5, 8, 0, 0))
x5<-as.matrix(cbind(x0,x1,x2,x3,x4))
x6<-x5%*%b11+errorC
x7<-as.matrix(x5[1:N,c(2:(m+1))])
x<-as.matrix(cbind(x6,x7))
N<-nrow(x)
C<-ncol(x)-1
z1<- as.matrix(x[1:N,c(2:(C+1))])
zt<-as.matrix(t(z1))

# Extracting rows and adding row #

y1<-as.matrix(x[1:N,1])
y<-y1-mean(y1)
meanz1<-1:C
for(i in 1:C)
{
meanz1[i]<-mean(z1[,i])
}
tmeanz1<-t(meanz1)
v1<-as.matrix(rep(1,N),nrow=N,ncol=1)
mean<-v1%*%tmeanz1
zc<-z1-mean
west<-(solve(t(zc)%*%zc))%*%(t(zc)%*%y)
yest<-z1%*%west
delta<-0.00005
max1<-1000000
while(max1>=delta)
{
res<-(y1-yest)
rank1<-as.matrix((rank(res)-((N+1)/2)))
z1<-as.matrix(z1[1:N,c(1:C)])
meanz1<-1:C
for(i in 1:C)
{
meanz1[i]<-mean(z1[,i])
}
tmeanz1<-t(meanz1)
```

```
v1<-as.matrix(rep(1,N),nrow=N,ncol=1)
mean<-v1%*%tmeanz1
zc<-z1-mean
d<-solve(t(zc)%*%zc)%*%t(zc)%*%rank1
w<-as.matrix(z1%*%d)
K<-(N*(N-1)/2)
th<-1:K
th2<-0
th4<-0
k<-0
i<-0
for(i in 1:(N-1))
{
for(j in (i+1):N)
{
k<-k+1
th[k]<-w[i]-w[j]
th1<-th[k]
ratio1<-res[i]-res[j]
th10<-ratio1/th1
th2[k]<-th10
th3<-abs(th1)+th4
th4<-th3
}
}
i<-1:K
wh<-0
for(i in 1:K)
{
th5<- th[i]
wh4<-abs(th5)/th4
wh[i]<-wh4
}
so<-data.matrix(cbind(th2,wh))
e<-so[order(th2),]
cusum<-cumsum(e[,2])
da<-cbind(e,cusum)
c<-da[length(which(cusum<=0.5)),1]
bestnew<-west+c*d
yestnew<-z1%*%bestnew
yest<-yestnew
diff<-as.matrix(abs(bestnew-west))
max1<-max(diff)
west<-bestnew
}
residual<-y1-yest
intercept<-median(residual)

beta<-as.matrix(cbind(intercept,t(bestnew)))

ones<-as.matrix(rep(1,N),ncol=1)
z2<-as.matrix(cbind(ones,z1),ncol=2)
pred<-z2%*%t(beta)
finalres<-(y1-pred)
rank2<-as.matrix((rank(finalres)-((N+1)/2)))

srwrF<-t(rank2)%*%finalres

nofire<-nrow(finalres)
nm<-nofire*(nofire+1)/2
```

```
k<-0
soaij<-0
aij<-1:nm
for(i in 1:N)
{
for(j in i:N)
{
k<-k+1
aij[k]<-(finalres[i]+finalres[j])/2
}
}
soaij<-sort(aij)
a<-N*(N+1)/4
b<-sqrt(N*(N+1)*(2*N+1)/24)
c1<-round((1/2)+a-1.645*b)
c2<-round((1/2)+a+1.645*b)
f<-sqrt(N/(N-(C+1)))
tou<-0
tou<-f*sqrt(N)*(soaij[c2]-soaij[c1])/(2*1.645)
tou
beta


##########################Subset Selection ########################

z5<-as.matrix(cbind(rep(1,N),z1))
I1<-ncol(z5)-1
for(I in 1:I1)
{
comb<-combn(1:I1,I)
choose<-choose(I1,I)
for(n in 1:choose)
{

# Extracting rows and adding row #

z11<-z5[1:N,c(1,comb[,n]+1)]
c11<-ncol(z11)
z6<-as.matrix(z11[1:N,c(2:c11)])
nc<-ncol(z6)
C<-ncol(z6)
zt<-as.matrix(t(z6))

# Extracting rows and adding row #

y1<-as.matrix(x[1:N,1])
y<-y1-mean(y1)
meanz6<-1:C
for(i in 1:C)
{
meanz6[i]<-mean(z6[,i])
}
tmeanz6<-t(meanz6)
v1<-as.matrix(rep(1,N),nrow=N,ncol=1)
mean<-v1%*%tmeanz6
zc<-z6-mean
west<-(solve(t(zc)%*%zc))%*%(t(zc)%*%y)
yest<-z6%*%west
delta<-0.00005
max1<-1000000
```

```
while(max1>=delta)
{
res<-(y1-yest)
rank1<-as.matrix((rank(res)-((N+1)/2)))
z6<-as.matrix(z6[1:N,c(1:C)])
meanz6<-1:C
for(i in 1:C)
{
meanz6[i]<-mean(z6[,i])
}
tmeanz6<-t(meanz6)
v1<-as.matrix(rep(1,N),nrow=N,ncol=1)
mean<-v1%*%tmeanz6
zc<-z6-mean
d<-solve(t(zc)%*%zc)%*%t(zc)%*%rank1
w<-as.matrix(z6%*%d)
K<-(N*(N-1)/2)
th<-1:K
th2<-0
th4<-0
k<-0
i<-0
for(i in 1:(N-1))
{
for(j in (i+1):N)
{
k<-k+1
th[k]<-w[i]-w[j]
th1<-th[k]
ratio1<-res[i]-res[j]
th10<-ratio1/th1
th2[k]<-th10
th3<-abs(th1)+th4
th4<-th3
}
}
i<-1:K
wh<-0
for(i in 1:K)
{
th5<- th[i]
wh4<-abs(th5)/th4
wh[i]<-wh4
}
so<-data.matrix(cbind(th2,wh))
e<-so[order(th2),]
cusum<-cumsum(e[,2])
da<-cbind(e,cusum)
c<-da[length(which(cusum<=0.5)),1]
bestnew<-west+c*d
yestnew<-z6%*%bestnew
yest<-yestnew
diff<-as.matrix(abs(bestnew-west))
max1<-max(diff)
west<-bestnew
}
residual<-y1-yest
intercept<-median(residual)
beta<-as.matrix(cbind(intercept,t(bestnew)))
ones<-as.matrix(rep(1,N),ncol=1)
```

```
z7<-as.matrix(cbind(ones,z6),ncol=2)
spred<-z7%*%t(beta)
finalres1<-(y1-spred)
rank3<-as.matrix((rank(finalres1)-((N+1)/2)))

srwrS<-t(rank3)%*%finalres1

np<-(srwrS-srwrF)/((N+1)/sqrt(48)*tou)-((I1+1)-2*c11)

g<-colnames(z6)
print(g)
cbin<-matrix(c(np),nrow=1,ncol=,byrow=TRUE,
       dimnames = list(c("np"),c("NP value ")))
print(t(cbin))
}
}
}
```

R-Code for Stepwise Procedure:

Following is the R function devised to implement the stepwise procedure developed in Section 5. The function is named as "subpoisreg". The usage of the function is as follows:

>subpoisreg(x)

where 'x', argument to the function, is a dataframe with first column as response, second as column of ones and rest of the columns as regressors.

```
subpoisreg<-function(x)
{
C<-ncol(x)
N<-nrow(x)
z<- as.matrix(x[1:N,c(2:ncol(x))])
Y<- as.matrix(x[1:N,c(1:1)])
m1<-glm(Y~z, family=poisson);
m2<-deviance(m1);
m2

########### one/two variable ########################################
############# Selection of -1 ########################################
no<-ncol(x)-2
if(no<3) {
if(no>0){
z<- x[,c(2:ncol(x))]
y<- matrix(x[,1])
y=x[,1]
z<-(x[,3:C])
z20<-as.matrix(z)
cz<-ncol(z20)
m2<-1:cz
m3<-1:cz
for(i in 1:cz)
{
z1<-z20[,i]
m1<-glm(y~z1, family=poisson)
m2[i]<-deviance(m1)
m3[i]<-i
}
m4<-cbind(m3,m2)
D1<-min(m2)
v<-which(m2==D1)
c1<-colnames(z[v])
X0<-rep(1,N)
z<-cbind(X0,z)
z2<-z[,c(1,v+1)]
if(no<2) c1<-c("X1") else c1<-
colnames(z2[2])
z2<-z2[,-1]
if(no>1) {z<-z[,c(-(v+1))]} else print("no
variable for selection")
z} else print("selected model is only intercept model")
```

```
############## Selection of - 2 ######################################
z20<-cbind(z2,z)
z1<-as.matrix(z20[,-(2)])
c2<-colnames(z20[3])
m1<-glm(y~z1, family=poisson)
D2<-deviance(m1)
p<-ncol(z1)
r1<-(D1-D2)/(D2/(n-p-1))
f1<-qf(0.95,1,(n-p-1))
c2<-cbind(c1,c2)
if(r1<f1) c3<-c1 else c3<-c2
print(t(c3))
print("terminate")
} else {
2

############## Selection of -1 ######################################
no<-ncol(x)-2
if(no>0){
z<- x[,c(2:ncol(x))]
y<- matrix(x[,1])
y=x[,1]
z<-(x[,3:C])
z20<-as.matrix(z)
cz<-ncol(z20)
m2<-1:cz
m3<-1:cz
for(i in 1:cz)
{
z1<-z20[,i]
m1<-glm(y~z1, family=poisson)
m2[i]<-deviance(m1)
m3[i]<-i
}
m4<-cbind(m3,m2)
D1<-min(m2)
v<-which(m2==D1)
c1<-colnames(z[v])
X0<-rep(1,N)
z<-cbind(X0,z)
z2<-z[,c(1,v+1)]
if(no<2) c1<-c("X1") else c1<-
colnames(z2[2])
print(t(c1))
z2<-z2[,-1]
if(no>1) {z<-z[,c(-(1),-(v+1))]} else print("no
variable for selection")
} else print("selected model is only intercept model")

############## Selection of - 2 ######################################
if(no>1) {
cz<-ncol(as.matrix(z))
m2<-1:cz
m3<-1:cz
z20<-as.matrix(z)
for(i in 1:cz)
{
z1<-as.matrix(cbind(z2,z20[,i]))
m1<-glm(y~z1, family=poisson)
```

```
m2[i]<-deviance(m1)
m3[i]<-i
}
m4<-cbind(m3,m2)
D2<-min(m2)
v<-which(m2==D2)
p<-ncol(z1)
if(no<3) {z21<-cbind(X0,z20)
c2<-colnames(z21[2])} else {c2<-
colnames(z[v])}
c11<-cbind(c1,c2)
print(t(c2))
z2<-cbind(z2,z[,v])
z<-z[,-(v)]
r<-rank(z)
r1<-(D1-D2)/(D2/(n-p-1))
f1<-qf(0.95,1,(n-p-1))
D1<-D2
z} else print(t(c1))

############## Iteration 3 ##############################################
while(r1>f1)
{
if(no>2) {
cz<-ncol(as.matrix(z))
cz
if(cz<1) cz=1
m2<-1:cz
m3<-1:cz
for(i in 1:cz)
{
z1<-cbind(z2,z[,i])
z23<-as.matrix(z1)
m1<-glm(y~z23, family=poisson)
m2[i]<-deviance(m1)
m3[i]<-i
}
m4<-cbind(m3,m2)
D2<-min(m2)
r1<-(D1-D2)/(D2/(n-p-1))
f1<-qf(0.95,1,(n-p-1))
if(r1>f1)
{
v<-which(m2==D2)
3
c3<-colnames(z[v])
p<-ncol(z1)
z2<-cbind(z2,z[,v])
z<-z[,-(v)]
D1<-D2
print(c3)
c4<-cbind(c(c11,c3))
print(t(c4))
colnames(z2)<-c4
} else
{ z2=z2
colnames(z2)<-c11
}
```

```
############# f delete ##############################################
if(r1>f1)
{
w<-ncol(z2)
z3<-z2[,-w]
o<-rep(1,N)
z3<-cbind(o,z3)
z7<-cbind(o,z2)
z4<-z7[,c(1,w+1)]
u<-colnames(z4)
I1<-ncol(z3)-1
I1
a<-1000000000000000000
comb<-combn(1:I1,(I1-1))
choose<-choose(I1,I1-1)
m5<-1:(I1-1)
m7<-1:(I1-1)
for(l in 1:choose)
{
# Extracting rows and adding row #
z5<-z3[1:N,c(1,comb[,l]+1)]
z6<- as.data.frame(cbind(z4,z5))
z6<-as.matrix(z6[c(-1,-3)])
z8<-z6
m1<-glm(Y~z8, family=poisson);
m5<-deviance(m1)
m7[l]<-m5
if(m5<a) m6=m5
if(m5>a) m6=a
a<-m6
if(l<2) z9=z6
if(m5<=a) z9=z6 else z9=z9
}
v<-which(m7==a)
z9<-as.data.frame(z9)
c5<-colnames(z9[v])
p<-ncol(z9)
n<-nrow(z9)
dr1<-(a-D1)/(D1/(n-p-1))
df1<-qf(0.95,1,(n-p-1))
if(dr1<df1) colnames(z9)
if(dr1>df1) colnames(z2)
if(dr1<df1) z10=z9
if(dr1>df1) z10=z2
z2<-z10
z22<-as.matrix(z2)
m11<-glm(Y~z22, family=poisson);
D1<-deviance(m11)
c2<-colnames(z)
c11<-colnames(z10)
} else {
print("terminate the procedure")
print("proper subset is")
print(t(c11))}
}
}
}
}
```

List of Paper Published and Presented:

I. Published

1. Sakate, D., M., Kashid, D., N. (2011). Subset Selection in Poisson Regression. Journal of Statistical Theory and Practice, 5 (2): 207-219.
2. Jadhav, N.,H., Kashid, D.,N. (2011). A Jackknifed Ridge M-estimator for Regression Model with Multicollinearity and Outliers. Journal of Statistical Theory and Practice, 5 (4): 659-673.

II. Presented

A. International Conferences / Seminars

1. **Submodel Selection in Linear Regression,** International Conference on 'Probability and Data Mining'. *Raleigh, USA, April 21-24, 2011.*
2. **Model Selection in Regression with Non-normal Error Terms,** International Conference on 'Statistics Probability and Related Areas'. *Cochin, December 19-22, 2011.*

B. National Conferences / Seminars

1. **Subset Selection in Poisson Regression,** National Conference on 'Industrial Statistics and Computational Mathematics'. *Jalgaon, January 21-23, 2010.*
2. **A Jackknifed Ridge M-estimator for Regression Model with Multicollinearity and Outliers,** National Conference on 'Advances in Statistical Theory and Applications'. *Bangalore, November 16-18, 2011.*
3. **Variable Selection Problem in Regression,** National Conference on 'Interface between Statistics and Other Sciences'. *Aurangabad, February 24-25, 2012.*