Report About Improved Screening in High Dimensional Regression

ABSTRACT: Inspired by Xiong [2] and [1], we arrange a algorithm to find M variables with small RSS(residual sum of square) in regression model, in which the M is fixed. According to the asymptotic criterion that better fitting means better submodel supposed by Xiong [2], our improved method is useful in some cases. Due to the high dimensionality, our algorithm can not find the submodel with the least RSS. But we can reduce the RSS based on the other methods such as LAR, FS or SIS.

Keywords: Screening, Branch and Bound Algorithm, EM Algorithm, Orthogonalizing Subset Screening

1 Introduction

1.1 Branch and Bound Algorithm

See [1] for detail...

BBA actually gives a method to search all the combination of any M-variables. In some degree, it construct a tree of models and search it using Breadth-First-Search. The standard BBA is obviously computational infeasible in relatively high dimensions. A pre-ordering BBA is considered to find the subset with smaller RSS efficiently, which is adopted in our algorithm.

1.2 Orthogonalizing Subset Screening

See [2] for detail...

It must be noted that there is an $\beta^{(ini)}$ and a $\beta^{(result)}$ in each iterative in this algorithm. According to xiong, $\beta^{(result)}$ behaves better than the $\beta^{(ini)}$.

On the other hand, if X-matrix is standardized, we can regard the each absolute value of each β in the model as a order to measure the importance of each variable in the model.

Since $\beta_0 = \beta^{(ini)}$, according to [2],

$$\beta_{k+1} = S_M\{a^{-1}X^T y + (I - a^{-1}X^T X)\beta^{(k)}\} \tag{1}$$

in our article, to give every variable a unique order, for the k_{max}(largest k) in the iterative:

$$\beta^{(result)} = a^{-1}X^T y + (I - a^{-1}X^T X)\beta^{(k_{max}-1)} \tag{2}$$

Hence

$$\beta^{(k_{max})} = S_M\{\beta^{(result)}\}$$

.

2 Our method

We prompt a combination of the two methods above.

When we have a $\beta^{(ini)}$, similar to the setting of node in [1], each node consist (S, k, β), where $|\beta| = |S|$. β is the estimates of coefficients in S computed by OSS(Orthogonalizing Subset Screening), which is considered to be the base of sorting.

In which the maximum and the N is pre-given.

flag = 0;
n = $|V|$;
Insert (V, 0, $\beta^{(ini)}$) into node list;
while $(n_s > M)\&(flag < maximum)$ **do**
  flag = flag + 1;
  Extract (S, k, β) from node list;
  Update β by OSS under variable set S with N iterations;
  $n_s = |S|$;
  Resort S by abs ofβ;
  compute residuals of $[s_1, ..., s_M]$ and update the min-residuals;
  for $i \leftarrow (k+1)$ **to** $\min(M, n_s - 1)$ **do**
    $S' \leftarrow drop(S, i)$;
    $\beta' \leftarrow$ drop(β, i);
    Insert (S', i-1, β') into node list;
  end
end

Algorithm 1: Method

3 Some strategies

Due to the searching-process in our method, our method is relatively computational intensive compared to the others(such as the iterative OSS). To mitigate this problem in some degree, we adopt some strategies as following:

3.1 Radius pre-ordering

Similar to the pre-ordering BBA, we could try to define radius for each node. The root nodes of a larger tree owns the smaller radius. The update of β using OSS algorithm is the corresponding procedure of pre-ordering in our case.

In which, P is the threshold radius fixed in advance.

flag = 0;
n = $|V|$;
Insert (V, 0, $\beta^{(ini)}$) into node list;
while $(n_s > M)\&(flag < maximum)$ **do**
  flag = flag + 1;
  Extract (S, k, β) from node list;
  $n_s = |S|$;
  ρ = k+n-n_s;
  if $\rho < P$ **then**
    Update β by OSS under variable set S with N iterations;
    Resort S by abs of β;
  end
  compute residuals of $[s_1, ..., s_M]$ and update the min-residuals;
  for $i \leftarrow (k+1)$ **to** $\min(M, n_s - 1)$ **do**
    $S' \leftarrow drop(S, i)$;
    $\beta' \leftarrow$ drop(β, i);
    Insert (S', i-1, β') into node list;
  end
end

Algorithm 2: Radius-Method

3.2 Heuristic strategies

To be more efficient, we attempt to use some heuristic strategies. In this section, simulated annealing are used to drop some modes with much higher RSS compared to their father nodes. Since each node is generated by dropping a variable from his father node, if a node with higher RSS than his father node, it is much probably that the dropped variable is important. So we will cast this node at the probability of $P(RSS_{current}, RSS_{father}, T)$ if $RSS_{current} > RSS_{father}$. where

$$P(A, B, T) = 1 - \exp\{-(A - B)/T\}$$

.

flag = 0;
n = $|V|$;
$RSS_{father} = \infty$; Insert (V, 0, $\beta^{(ini)}$, RSS_{father}) into node list;
while $(n_s > M)\&(flag < maximum)$ **do**
    flag = flag + 1;
    Extract (S, k, β, RSS_{father}) from node list;
    $n_s = |S|$;
    ρ = k+n-n_s;
    if $\rho < P$ **then**
        Update β by OSS under variable set S with N iterations;
        Resort S by abs of β;
    end
    $RSS_{current}$ = residuals of $[s_1, ..., s_M]$;
    update the min-residuals;
    if $P(RSS_{current}, RSS_{father}, T) > random()$ **then**
        continue;
    end
    $RSS_{father} = RSS_{current}$;
    for $i \leftarrow (k+1)$ **to** $\min(M, n_s - 1)$ **do**
        $S' \leftarrow drop(S, i)$;
        $\beta' \leftarrow$ drop(β, i);
        Insert (S', i-1, β',RSS_{father}) into node list;
    end
end

Algorithm 3: Heuristic Radius-Method

3.3 Cutoff some variables

In some high dimensional situations, searching the better subset using our method are remains slowly. Intuitively, true variables are tent to rank relatively in front even though some of them are not in top-M (I am not sure about whether there is such theory or not...). So we try to cast Q least-important variables in each node. We set Q equals to $(n_s - M - 1)/L$, in which L is a tuning parameter used for control.

4 Simulations

Some of the simulation result:

Table 1: n=50, p=100, sd=1

(a) Coverage Rate(%)

	ini	EM-Im	HRBBA
SIS	71.20	89.90	93.60
SF	83.70	83.70	83.70
Lar	96.00	96.60	96.50
LASSO	95.00	96.80	97.00
Elastic	95.80	96.70	96.80
GLM	88.20	90.80	93.00
SCAD	90.10	91.50	92.70

(b) Residual Sum of Square

	ini	EM-Im	HRBBA
SIS	296.83	23.95	11.70
SF	14.85	14.52	10.93
Lar	48.45	19.57	10.73
LASSO	51.66	18.56	10.24
Elastic	44.84	18.57	10.16
GLM	93.49	22.82	11.57
SCAD	43.32	17.69	10.57

Table 2: n=100, p=300, sd=1.5

(a) Coverage Rate(%)

	ini	EM-Impr	H-Radius
SIS	77.20	91.90	95.70
SF	97.50	97.50	97.40
Lar	97.80	98.10	99.30
LASSO	98.20	98.70	99.10
Elastic	98.70	98.40	99.30
GLM	87.20	92.50	96.90
SCAD	99.10	99.10	99.10

(b) Residual Sum of Square

	ini	EM-Impr	H-Radius
SIS	851.78	144.92	103.46
SF	92.57	92.47	87.72
Lar	316.41	133.64	98.98
LASSO	227.49	127.14	98.22
Elastic	223.97	128.50	99.09
GLM	450.35	152.48	104.07
SCAD	136.26	103.73	90.65

Acknowledgements

References

[1] Hofmann, Marc, Cristian Gatu, and Erricos John Kontoghiorghes. "Efficient algorithms for computing the best subset regression models for large-scale problems." *Computational Statistics & Data Analysis* 52.1 (2007): 16-29.

[2] Xiong, Shifeng. "Better subset regression." Biometrika (2013): ast041.