Representative approach for big data dimension reduction with binary responses

Xuelong Wang, Jie Yang

Department of Mathematics, Computer Science and Statistics
University of Illinois at Chicago

July 23, 2019

Fundamental assumption

Let random variable $X \in \mathbb{R}^p$, $Y \in \mathbb{R}$ and $\eta \in \mathbb{R}^{p \times d}$, where $d << p$

$$Y|X \sim Y|\eta^T X$$

Example

1. Linear regression: $Y = a + \beta_1^T X + \beta_2^T X + \epsilon$
2. NonLinear regression: $Y = a + \exp(\beta^T X) + \sin(\beta_2^T X)\epsilon$
3. Generalized linear regression: $probit(p) = a + \beta_1^T X + \beta_2^T X$

Where η is a set of basis of $span(\beta_1, \beta_2)$

Sufficient dimension reduction

Dimension-reduction subspace

$$Y \perp\!\!\!\perp X | \eta^T X \rightarrow Y \perp\!\!\!\perp X | (\eta A)^T X \rightarrow Y \perp\!\!\!\perp X | P_S X,$$

Where $P_{\mathcal{S}}$ is the projection matrix of subspace $\mathcal{S}$
$\mathcal{S}$ is called the dimension-reduction subspace
However, the $\mathcal{S}$ is not unique, i.e. if $\mathcal{S} \subset \mathcal{S}_1$, then $\mathcal{S}_1$ is also a dimension-reduction space.

Central Subspace

$$S_{Y|X} = \cap S_{SDR}$$

The target of sufficient dimension reduction is to estimate the structure of $S_{Y|X}$

Estimating the central subspace

Y_1	$\mathbf{x}_1(=(x_{11}, x_{12}, \cdots, x_{1p})')$
Y_2	$\mathbf{x}_2(=(x_{21}, x_{22}, \cdots, x_{2p})')$
Y_3	$\mathbf{x}_3(=(x_{31}, x_{32}, \cdots, x_{3p})')$
Y_4	$\mathbf{x}_4(=(x_{41}, x_{42}, \cdots, x_{4p})')$
Y_5	....
.	....
.	....
.	....
Y_n	$\mathbf{x}_n(=(x_{n1}, x_{n2}, \cdots, x_{np})')$

Original data

$Y_{(1)}$	$\mathbf{x}_{(1)}(=(x_{(1)1}, x_{(1)2}, \cdots, x_{(1)p})')$
$Y_{(2)}$	$\mathbf{x}_{(2)}(=(x_{(2)1}, x_{(2)2}, \cdots, x_{(2)p})')$
$Y_{(3)}$	$\mathbf{x}_{(3)}(=(x_{(3)1}, x_{(3)2}, \cdots, x_{(3)p})')$
$Y_{(4)}$	$\mathbf{x}_{(4)}(=(x_{(4)1}, x_{(4)2}, \cdots, x_{(4)p})')$
$Y_{(5)}$	....
.	....
.	....
.	....
$Y_{(n)}$	$\mathbf{x}_{(n)}(=(x_{(n)1}, x_{(n)2}, \cdots, x_{(n)p})')$

Sorted and sliced by y

$\tilde{Y}_1$	$\bar{\mathbf{x}}_1(=(\bar{x}_{11}, \bar{x}_{12}, \cdots, \bar{x}_{1p})')$
$\tilde{Y}_2$	$\bar{\mathbf{x}}_2(=(\bar{x}_{21}, \bar{x}_{22}, \cdots, \bar{x}_{2p})')$
.	....
.	....
.	....
$\tilde{Y}_H$	$\bar{\mathbf{x}}_H(=(\bar{x}_{H1}, x_{H2}, \cdots, x_{Hp})')$

Slice means of standardized data

$$\hat{V} = n^{-1} \sum_{h=1}^{H} n_h \bar{x}_h \bar{x}_h^T$$

Estimated Covariance matrix

Conduct PCA on $\hat{V}$
Find the first Kth eigenvectors $\hat{\eta}$

$$\hat{\beta}_k = \hat{\eta}_k \Sigma_{xx}^{-1/2}$$

Problem with Binary response

- Limited the number of sliced
- For SIR, it can only find one basis at most
- For SAVE, it also suffers from the limit number of slices

Probability Enhanced method for binary response

Main idea

- $S_{Y|X} = S_{P(Y|X)|X}$
- Estimated the Probability related rank by weighted support vector machine(WSVM)
- It enriches the information of response

Scalability of large data

- Kernel matrix
- tunning parameter

Representative approach

Representative

A Representative is a summary statistic of data points within a cluster: For $(X_i, Y_i), i \in I_k$

$$X_k^* = R(X_1, \ldots, X_{nk}), \quad Y_k^* = R(Y_1, \ldots, Y_{nk}),$$

where $R : \mathbb{R}^p \to \mathbb{R}^p$ is the summarizing function.

Main idea

After transformation Y^* will become continuous, but the relation (the $\beta's$) of Y^* and X^* will almost keep the same:

$$Y = f(X^T\beta_1, \ldots, X^T\beta_k) \to Y^* \approx G({X^*}^T\beta_1, \ldots, {X^*}^T\beta_k)$$

Method

Steps

- Split the (X, Y) into K clusters $I_1, \ldots, I_K$
- Summary the representative for each cluster k

$$Y_k^* = \bar{Y}_k = \frac{\sum_i Y_i}{nk}, \quad X_k^* = \bar{X}_k = \frac{\sum_i X_i}{nk}, \quad i \in I_k$$

  Note that we choose the cluster average as the summary statistics R
- Apply SDR methods on the representatives

Method

The representatives keeps the relations $\beta's$

The representatives of Y actually is actually the conditional probability of $P(Y|X)$,

$$\bar{Y}_k \to P(Y=1|X=X_k) \text{ as } N, K, N/K \to \infty$$

It's can be shown that

$$S_{Y|X} = S_{P(Y|X)|X},$$

Additional value: Big data solution (n is large)

Clustering step

Clustering step reduced the sample size from N to K

- $(Y_1, X_1) \dots (Y_N, X_N) \rightarrow (Y_1^*, X_1^*) \dots (Y_K^*, X_K^*)$
- Note if the data set is too large, we could also use the online clustering method

Additional value: Big data solution (n is large)

Parallel Algorithm for SIR and SAVE

1. Split the sliced data into b blocks, $X_1, \ldots X_b$
2. Load each block X_B and Calculate the statistics for each block such as $\bar{X}_b, \bar{X}_{hb}, n_{hb}, X_{hb}^T X_{hb}$
3. Summary the statsitics across the blocks and slices to get the candidate matrix M_{SIR}, M_{SAVE}

   # Simulation result

Simulation setup

Data generation Model:laten model

$$Y = \begin{cases} 0 & (X\beta_1)^2 * e^{(X\beta_2)} * \sin(X\beta_3) + \epsilon < 0 \\ 1 & \text{Otherwise} \end{cases}$$

where

- $X \in \mathbb{R}^6 \sim N(0_6, I_6)$
- $\beta_i = e_i = (0, \ldots, 1, 0, \ldots, 0)^T$, so in our case the linear combination is X_1, X_2, X_3
- $\epsilon \sim N(0, 1)$

Simulation result

Performance Evaluation

- Hypothesis Test: Test how many bases of the Central space
- Distance: Measure the distance between the estimated $\hat{\beta}'s$ and true $\beta's$

Result summary

- The true basis is (e_1^T, e_2^T, e_3^T)
- For SAVE, it can only find 2 of the 3 basis
- For the representative SAVE, it can find all of them

Simulation result

	sir_original				sir_rep				sir_p			
	Log_n											
Direction/Distance	3	4	5	6	**3**	**4**	**5**	**6**	3	4	5	6
0D vs >= 1D	1.0000000	1.0000000	1.0000000	1.0000000	**0.7500000**	**1.0000000**	**1.0000000**	**1.0000000**	1.0000000	1.0000000	1.0000000	1.0000000
1D vs >= 2D	0.7500000	0.7300000	0.6600000	0.6850000	**0.1650000**	**1.0000000**	**1.0000000**	**1.0000000**	1.0000000	1.0000000	1.0000000	1.0000000
2D vs >= 3D	NaN	NaN	NaN	NaN	**0.0100000**	**0.0100000**	**0.0000000**	**0.0100000**	0.0450000	0.0400000	0.0350000	0.0650000
3D vs >= 4D	NaN	NaN	NaN	NaN	**0.0000000**	**0.0000000**	**0.0000000**	**0.0000000**	0.0000000	0.0050000	0.0050000	0.0050000
4D vs >= 5D	NaN	NaN	NaN	NaN	**0.0000000**	**0.0000000**	**0.0000000**	**0.0000000**	0.0000000	0.0000000	0.0000000	0.0000000
5D vs >= 6D	NaN	NaN	NaN	NaN	**0.0000000**	**0.0000000**	**0.0000000**	**0.0000000**	0.0000000	0.0000000	0.0000000	0.0000000
ave_frob	1.1256210	1.0494941	1.0566714	1.0944659	**1.2669218**	**1.2743468**	**1.2507151**	**1.2509262**	1.3655995	1.2860677	1.2446272	1.2866278
ave_Q	0.6846554	0.6934016	0.7257302	0.7498777	**0.6523792**	**0.6635268**	**0.6479987**	**0.6336554**	0.6547422	0.6545569	0.6319888	0.6936892
ave_R	0.1419350	0.1370806	0.1388120	0.1486747	**0.1490037**	**0.1504459**	**0.1465382**	**0.1456828**	0.1773883	0.1530598	0.1451360	0.1539329

Simulation result

	save_original	save_rep			
		Log_n			
Direction/Distance	3 4 5 6	**3**	**4**	**5**	**6**
0D vs >= 1D	[illegible]	**0.0000000**	**0.0500000**	**1.0000000**	**1.0000000**
1D vs >= 2D	[illegible]	**0.0000000**	**0.0000000**	**1.0000000**	**1.0000000**
2D vs >= 3D	[illegible]	**0.0000000**	**0.0000000**	**0.0500000**	**1.0000000**

Future work

- A different choice of K will affect the performance of SDR methods