Penalized neural network sufficient dimension reduction

-In preparation-

Ko Young Hun

Department of Statistic
Korea University

Reasearch Questions

Main Questions

- Is it possible to estimate the Central Mean Subspace of SDR through MLP?
- Is it possible to estimate the SDR subspace and the number of dimensions at the same time?

Introduction-PCA

PCA algorithm: To find out top k eigenvalues/eigenvectors

$$X = \mathrm{N} \times \mathrm{m} \text{ data matrix}$$

each data point x_i = column vector, $i = 1, 2, \ldots, m$

- $\bar{\mathrm{X}} = \frac{1}{m} \sum_{i=1}^{m} X_i$
- $X \leftarrow$ subtract mean $\bar{\mathrm{X}}$ from each column vector X_i in $\bar{X}$
- $\Sigma \leftarrow XX^T$ ... covariance matrix of X
- $\{\lambda_i, u_i\}_{i=1,\ldots,N}$ = eigenvectors/eigenvalues of Σ

  $\ldots \lambda_1 \geq \lambda_2, \cdots \geq \lambda_N$
- Return $\{\lambda_i, u_i\}_{i=1,\ldots,k}$ top k-PCA components

Introduction-PLS

PLS algorithm: To find out the optimal low-dimensional linear combination of independent variables for explaining the relationship with the dependent variable.

Let $X \in \mathbb{R}^{\mathbb{P}},\ Y \in \mathbb{R},\ t = Xw$ then

$$
\begin{aligned}
\text{Max}\{Cov(t, Y)\} &= Cov(Xw, Y) \\
&= E[(Xw - E[Xw])(Y - E(Y))] \\
&= E(XwY) \quad \text{(Without Loss of generality)} \\
&= \frac{1}{n}\sum_{i=1}^{n}(Xw)_i Y_i \quad \text{(In sample level)} \\
&= \frac{1}{n} w^T (X^T Y)
\end{aligned}
$$

PLS algorithm

- Set $X_1 = X,\ Y_1 = Y$ and $w_1 = \frac{X_1^T Y_1}{||X_1^T Y_1||}$
- Using w_1, calculate $t_1 = X_1 w_1$
- calculate $b_1 = (t_1^T t_1)^{-1} t_1 Y_1$ (by least squares)
- $Y_2 = Y_1 - t_1 b_1,\ X_2 = X_1 - t_1 p_1^T$ where $p_1^T = (t_1^T t_1)^{-1} t_1 X_1$
- Repeat the above process to calculate

Drawbacks of PCA and PLS

There are several drawbacks of PCA and PLS

- **Loss of information**:Since PCA selects the Principal component with the largest variance, information loss occurs.

  $\Rightarrow$ Do not caputure regression information

- **Do not capture nonlinearity**: PLS generates the new variables through linear combinations of variables which implies It can not express nonlinearity of each variables

Why we need SDR?

Sufficient dimension reduction can address these two drawbacks!

Sufficient Dimension Reduction

Supervised dimension reduction

- **Problem**: regression problem $\Rightarrow$ $X \in \mathbb{R}^{\mathbb{P}}, Y \in \mathbb{R}$
- model: $Y = f(X_1, X_2, \ldots, X_p) + \epsilon$ where $X \perp\!\!\!\perp \epsilon$
- we want to the dimenson $\beta_1, \beta_2, \ldots, \beta_d$ such that

$$Y = g(\beta_1^T X, \ldots, \beta_d^T X) + \epsilon$$

- if $d << p$ then fitting g is much easier than f

Sufficient Dimension Reduction

Supervised dimension reduction

Basic Setting: $X \in \mathbb{R}^{\mathbb{P}}, Y \in \mathbb{R}$

Goal: Find $\beta \in \mathbb{R}^{\mathsf{p} \times \mathsf{d}}$ such that

$$Y \perp\!\!\!\perp X | \beta^T X \Leftrightarrow Y | \beta^T X \stackrel{\mathsf{d}}{=} Y | X$$

Advantages:

- We can replace X with $\beta^T X$ which have smaller predictor without loss of information
- It can be used for futher analysis such as regression or classification

Central Subspace

Basic Setting: Let $\delta(W)$ be class of subspace on X such that

$$Y \perp\!\!\!\perp X | W^T X$$

then central subspace is defined as follows

$$\cap\{\varphi \in \delta(W)\}$$

which implies all intersaction of $\delta(W)$

Now central subspace is denoted by $S_{Y|X}$

Central Mean Subspace

Basic Setting: $\delta(\beta)$ denote a general subspace of $\mathbb{R}^p$ which implies dimension reduction subspace then we want to find the β such that

$$Y \perp\!\!\!\perp E(Y|X)|\beta^T X$$

then following statements are equivalent

- $COV(Y, E(Y|X)|\beta^T X)$
- $E(Y|X)$ is function of $\beta^T X$

Now central mean subspace is denoted by $S_{E(Y|X)}$

$\Rightarrow$ **The main goal of this study is to estimate β that satisfies the above conditions through the FCN model!**

Group Lasso

Group lasso can be expressed as follows

$$\underset{\beta}{\text{argmin}} \frac{1}{2}\|Y - \sum_{l=1}^{m} X^{(l)}\beta^{(l)}\|_2^2 + \lambda \sum_{l=1}^{m} \sqrt{p_l}\|\beta^{(l)}\|_2$$

- $X^{(l)}$ is submatrix of X
- $\beta^{(l)}$ is coefficient of correspoding $X^{(l)}$
- p_l denote the length of $\beta^{(l)}$

Group Lasso

The typical reasons for using Group Lasso are as follows

- if the variables are highly correlated, Lasso delete the correlated variables **Randomly**
- if the number of group variables equals to number of variables then It is same as Lasso
- Group Lasso can be solved same as Lasso

The parameters of MLP is given in matrix form, so in this study, we use Group Lasso to force specific rows or columns to converge to zero

Multi Layer Perceptron

Suppose we have $\eta = \{\theta_a, b_a\}_{a=0}^{N+1}$, then final output value is as follows

$$f_\eta(x) = \theta_{N+1}^T z_N(x) + b_{N+1}$$

- where $z_a(x) = \psi_a(\theta_a^T z_{a-1}(x) + b_a)$
- $\psi_a(.)$ is a-th activation function
- we use 3 hidden layer
- we use $\psi_a(x) = (e^x - e^{-x})/(e^x + e^{-x})$ for $a = 1, 3$ and $\psi_2(x) = \max(0, x)$
- To minimize confusion, the notation θ_0 is now replaced with β

Directional Regression

Directional regression is inverse regression method to estimate central subspace

Assumption

Inverse regression method typically requires two statistical assumptions

- **Linearity:** $E(X|\beta^T X) = \Sigma\beta(\beta^T\Sigma\beta)^{-1}\beta^T X$
- **Constant:** $\text{cov}(X|\beta^T X) = \Sigma - \Sigma\beta(\beta^T\Sigma\beta)^{-1}\beta^T\Sigma$

where $\Sigma = \text{cov}(X)$

Directional Regression

Assuming $d = S_{Y|X}$ is known then

$$\begin{aligned} M_{\mathsf{DR}} &= 2\mathsf{E}^2\{\mathsf{E}(X|Y)\mathsf{E}(X^T|Y)\} \\ &\quad + 2\mathsf{E}\{\mathsf{E}(X^T|Y)\mathsf{E}(X|Y)\}\mathsf{E}\{\mathsf{E}(X|Y)\mathsf{E}(X^T|Y)\} \\ &\quad + 2\mathsf{E}\{\mathsf{E}^2(XX^T)\} - 2\Sigma \end{aligned}$$

then using M_{DR} solve GEV

$$M_{\mathsf{DR}}\beta_i = \lambda_i \Sigma \beta_i$$

- if $i = j, \quad \beta_i^T \Sigma \beta_j = 1$
- if $i \neq j, \quad \beta_i^T \Sigma \beta_j = 0$
- then $\beta = (\beta_1, \ldots, \beta_d)$ is basis of $S_{Y|X}$
- we use Directional regression to initialize β

Ladle Estimator

The Ladle estimator Li-[1] is an order determination method that utilizes both eigenvalues and eigenvectors.
Let's define f_n and ϕ_n as follows:

$$f_n^0(k) = \begin{cases} 0, & k = 0 \\ n^{-1} \sum_{i=1}^{n} \{1 - |\det(\hat{B}_k{}^T B_{k,i}^*)|\}, & k = 1, 2, \ldots, p-1 \end{cases}$$

where $B_{k,i}^*$ is calculated by bootstrap samples. Using this,

$$f_n : \ \{0, 1, \ldots, p-1\} \to \mathbb{R}, \quad f_n(k) = f_n^0(k) / \{1 + \sum_{i}^{p-1} f_n^0(i)\}$$

Ladle Estimator

Let $\hat{\lambda}_1, \ldots, \hat{\lambda}_p$ is eigenvalue of $\hat{M}$ define the function

$$\phi_n : \{0, 1, \ldots, p-1\} \to \mathbb{R} \quad \phi_n(k) = \hat{\lambda}_{k+1} / (1 + \sum_{i=0}^{p-1} \hat{\lambda}_{i+1})$$

Then we define g_n as follows

$$g_n(k) = f_n(k) + \phi_n(k) \equiv f_n^0(k) / \{1 + \sum_{i}^{p-1} f_n^0(i)\} + \hat{\lambda}_{k+1} / (1 + \sum_{i=0}^{p-1} \hat{\lambda}_{i+1})$$

Theorem

$\hat{d} = \text{argmin}\{g_n(k) : k \in D(g_n)\}$ where $D()$ is domain of some function

- we use ladle estimator to give some information about dimension of $S_{E(Y|X)}$

Objective function

In this study, Two scenarios will be considered.

The rank of $S_{E(Y|X)}$ is known and unknown

Objective function(d is known):

$$L_{x,y}(\eta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, f_\eta(x_i)) + \lambda_0 ||\beta^T \Sigma_X \beta - I_d||_F^2$$

Objective function(d is unknown):

$$L_{x,y}(\eta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, f_\eta(x_i)) + \lambda_0 ||\beta^T \Sigma_X \beta - I_d||_F^2 + \lambda_1 \sum_{j=1}^{d} ||\theta_{1,j}||_2$$

Description

Basic Idea: It combines SDR and MLP.

Computation: In Computation, It combines Gist algorithm Gong-[2] and Blockwise Descent algorithm Simon-[3] The brief description is as follows.

- **First**, Update using Smooth term (Specifically, it includes loss function and Frobenius norm)
- **Second**, Using Soft Theresholding Operator, Update first layer parameter
- **Finally**, if Some criterion is satisfied(Line Search) then update. if Not, then Do not update and learning rate scheduling

Algorithm

Since objective function when d is known is Not differentiable, we adopted Gong-[2]'s idea

Algorithm 1 Training Penalized neural network with unknown d

Initialization: $\beta = \hat{\beta}_{DR}$ and get initial value of $\{\theta_1, b_1, \ldots, \theta_4\}$ using **Xaiver**

Set the nodes of first layer as $m_1 = \begin{cases} p-1 & \text{for } p \leq 10 \\ \lfloor p/\log p \rfloor & \text{for } p > 10 \end{cases}$

for $t = 1, 2, 3, \ldots$ **do**

$\eta^{(\iota,1)} = \eta^{(\iota-1)} - lr\nabla\{\frac{1}{n}\sum_{i=1}^{n}(y_i - f_\eta(x_i))^2 + \lambda_0||\beta^T\Sigma_X\beta - I_d||_F^2\}$

for $j = 1, 2, \ldots, m_1$ **do**

$$\theta_{(1,j)}^{(\iota,2)} = \left(1 - \frac{\text{lr}\lambda_1}{||\theta_{(1,j)}^{(\iota,1)}||_2}\right)_+ \theta_{(1,j)}^{(\iota,1)}$$

end for

if line search criterion is satisfied **then then**

$\eta^{(\iota)} = \eta^{(\iota,2)}$

else

$\eta^{(\iota)} = \eta^{(\iota-1)}$

$lr = \frac{9}{10}lr$

end if

end for

Distance measure of two linear subspace

General Loss

$$L_G(\hat{\beta}, \beta) = \|\hat{\beta}\hat{\beta}' - \beta\beta'\|_F^2$$

Projection Loss

$$L_P(\hat{\beta}, \beta) = \|P_{\hat{\beta}} - P_\beta\|_F$$

- where $P_A = A(A^T A)^{-1} A^T$
- The above losses were proposed by Gao-[4] and Li-[1], respectively, and are indicators that measure the distance between two linear spaces.
- Smaller values mean better performance
- we use **projection loss**

Simulation Study

Descriptions of experiment:

- ϵ_i are i.i.d from $N(0, 1)$
- $X \sim \text{MVN}(0, \Sigma)$ where $\Sigma_{(i,j)} = 0.5^{|i-j|}$
- Performing 5-fold cross validation, to find out optimal hyperparameter (ex: lr,λ_0, λ_1 ...)
- Using the best hyperparameter, 100 re-training processes are conducted while changing the seed value.
- We variate the sample sizes $n = 200, 500, 1000$
- Mean model indicates that $\hat{Y} = \bar{Y}$

Simulation Study

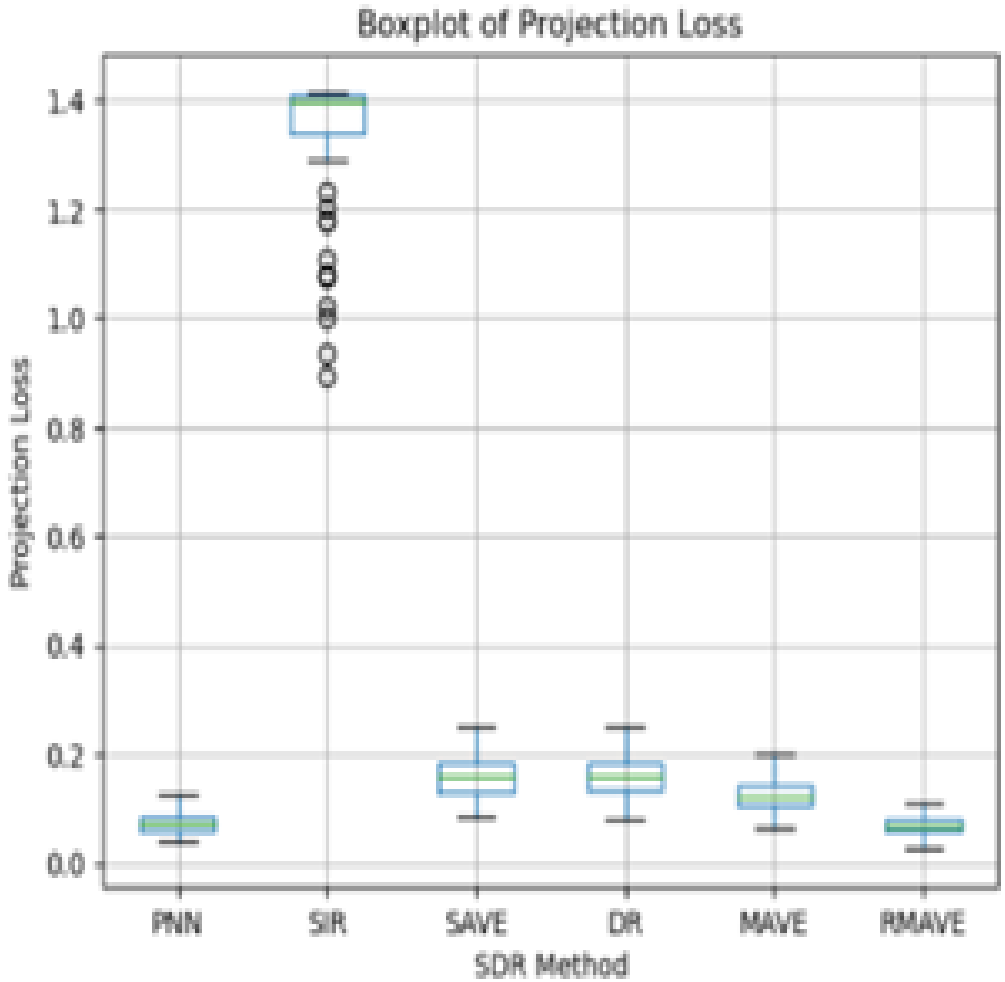
We generated the response variables using the equations below.

$$\textbf{Model I: } Y = 0.5(\beta_1^T X)^2 + 0.5\epsilon$$
$$\textbf{Model II: } Y = \arcsin\left(1/1 + |0.5 + \beta_1^T X|\right) + 0.2\epsilon$$
$$\textbf{Model III: } Y = \sin\left(\beta_1^T X\right) + 0.5\epsilon$$

- This model are used in Zhu-[5]
- The Number of true dimension is 1 and we variate $p = 10, 20$
- And we simulated another type of models when $d = 2$

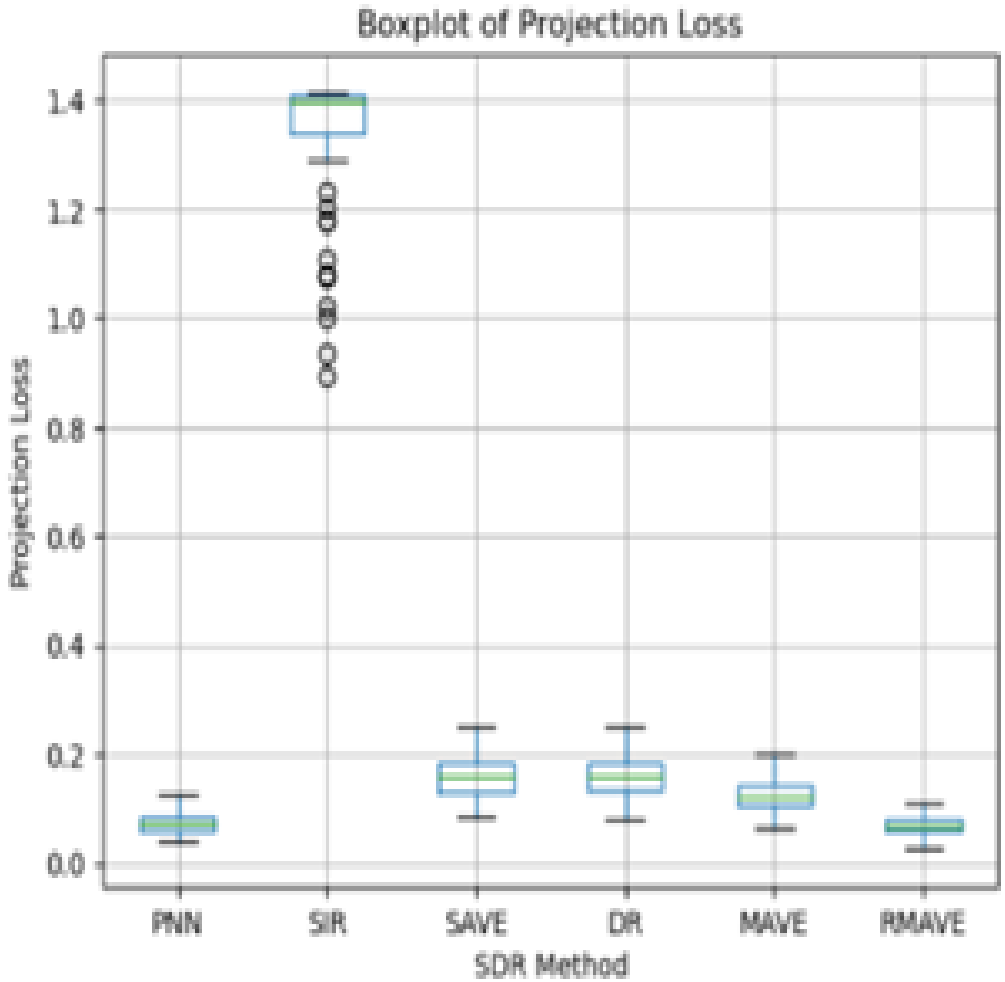
Simulation Study(d is known)

Model $Y = 0.5(\beta_1^T X)^2 + 0.5\epsilon \quad (n = 500, p = 20)$



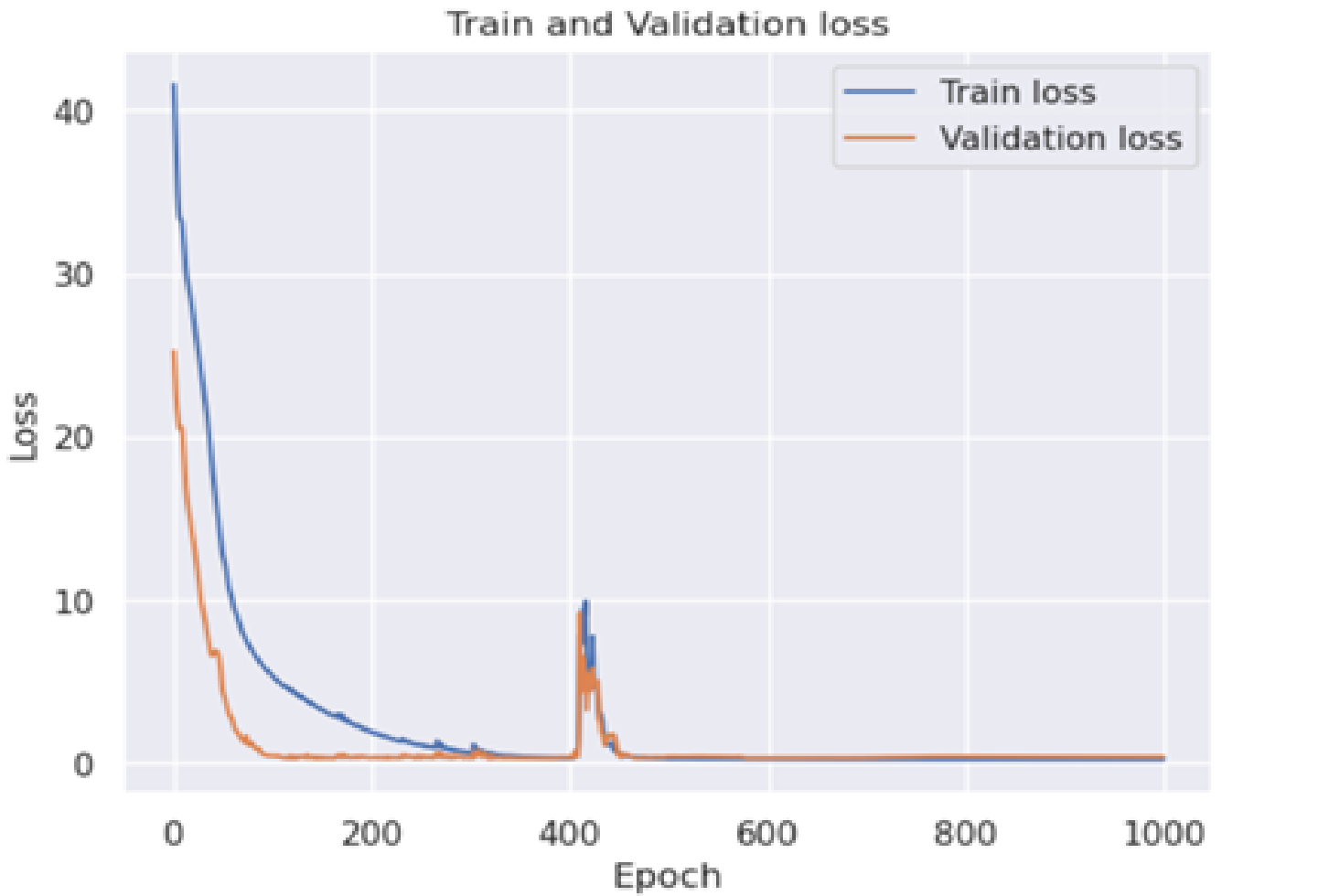
Simulation Study(d is known)

Model $Y = \arcsin\left(1/1 + |0.5 + \beta_1^T X|\right) + 0.2\epsilon \ (n = 500, p = 20)$



Simulation Study(d is known)

Model $Y = \arcsin\left(1/1 + |0.5 + \beta_1^T X|\right) + 0.2\epsilon \ (n = 500, p = 20)$



Simulation Study(d is known)

- PNN indicate Our model and SIR,SAVE, ... is other SDR model
- As seen on the previous page, it can be observed that PNN demonstrates superior model performance.
- Consistent with theory, it has also been observed that SIR experiences significantly reduced performance when dealing with symmetric distribution
- It has been confirmed that the loss function consistently exhibits a decreasing trend overall.

Additional work(To do)

- Hyperparameter tunning when d is unknown
- Comparison with other SDR models when d is unknown
- Real data analysis

Expected contribution

- Even if you don't know the true number of dimensions, You can estimate automatically
- Compared to other sufficient dimensional reduction methods, the calculation process is easier to understand.
- Applying regularization within group, also facilitates interpretation.
- This model is based on a neural network model, so it is also suitable for large datasets.

Reference

[1] Bing Li and Shaoli Wang. On directional regression for dimension reduction. *Journal of the American Statistical Association*, 102(479):997–1008, 2007.

[2] Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Huang, and Jieping Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems, 2013.

[3] Noah Simon, Jerome Friedman, and Trevor Hastie. A blockwise descent algorithm for group-penalized multiresponse and multinomial regression, 2013.

[4] Chao Gao, Zongming Ma, Zhao Ren, and Harrison H. Zhou. Minimax estimation in sparse canonical correlation analysis. *The Annals of Statistics*, 43(5):2168–2197, 2015.

[5] Yanyuan Ma and Liping Zhu. A semiparametric approach to dimension reduction. *Journal of the American Statistical Association*, 107(497):168–179, 2012. PMID: 23828688.