Problem 3

Consider the truncated power series representation for cubic splines with K interior knots

$$f(x) = \sum_{j=0}^{3} \beta_j X^j + \sum_{k=1}^{K} \theta_k (X - \varepsilon_k)_+^3$$

Prove that the natural boundary conditions for natural cubic spline imply the following linear constraints on the coefficients

$$\beta_2 = 0, \quad \sum_{k=1}^{K} \theta_k = 0$$

$$\beta_3 = 0, \quad \sum_{k=1}^{K} \varepsilon_k \theta_k = 0$$

As per the above given constraints β_2 & β_3, we get the following eqn.

$$f(x) = \beta_0 X^0 + \beta_1 x^1 + \cancel{\beta_2} x^2 + \cancel{\beta_3} x^3 + \sum_{k=1}^{K} \theta_k (x - \varepsilon_k)_+^3$$

$$= \beta_0 + \beta_1 x^1 + \sum_{k=1}^{K} \theta_k (x - \varepsilon_k)_+^3$$

Here $\beta_0 \cdot \underbrace{1}_{N_1(X) = 1}$ and $\beta_1 \underbrace{x^1}_{N_2(x) = X}$ [Given in the question]

we here constructed a new basis with the first two basis function.

Let us now consider the θ constraints, let the write down that

$$\sum_{k=1}^{K-2} \theta_k = -\theta_{K-1} - \theta_K \quad \text{and} \quad \sum_{k=1}^{K-2} \varepsilon_k \theta_k = -\varepsilon_{K-1}\theta_{K-1} - \varepsilon_K \theta_K$$

Now consider the truncated function and use the last two terms like done above

i.e consider $\sum_{k=1}^{K} \theta_k (x - \varepsilon_k)_+^3 = \sum_{k=1}^{K} \theta_k (x - \varepsilon_k)_+^3 +$

$$\underbrace{\theta_{K-1} (x - \varepsilon_{K-1})_+^3}_{(i)} + \underbrace{\theta_K (x - \varepsilon_K)_+^3}_{(ii)} \rightarrow (1)$$

To the above equation, apply the θ constraints to show last two terms of the above equation can be written as the sum of the N-2 first terms

We should now consider the last two terms for the computation from the equation 1

Let us now consider the term (i) from equation 1

$$\theta_{k-1}(x-\varepsilon_{k-1})^3_+ = \frac{(x-\varepsilon_{k-1})^3_+}{(\varepsilon_k-\varepsilon_{k-1})}(\theta_{k-1}\varepsilon_k - \theta_{k-1}\varepsilon_{k-1})$$

We got the above equation by multiplying and dividing $(\varepsilon_k - \varepsilon_{k-1})$ on the LHS..

$$= \frac{(x-\varepsilon_{k-1})^3_+}{(\varepsilon_k-\varepsilon_{k-1})}(\theta_{k-1}\varepsilon_k - \theta_{k-1}\varepsilon_{k-1})$$

$$= \frac{(x-\varepsilon_{k-1})^3_+}{(\varepsilon_k-\varepsilon_{k-1})}(\theta_{k-1}\varepsilon_k - \theta_{k-1}\varepsilon_{k-1} + \theta_k\varepsilon_k - \theta_k\varepsilon_k)$$

(Add and subract $\theta_k \varepsilon_k$)

$$= \frac{(x-\varepsilon_{k-1})^3_+}{(\varepsilon_k-\varepsilon_{k-1})}(\varepsilon_k(\theta_{k-1}+\theta_k) - \varepsilon_{k-1}\theta_{k-1} - \theta_k\varepsilon_k)$$

$$= \frac{(x-\varepsilon_{k-1})^3_+}{(\varepsilon_k-\varepsilon_{k-1})}\left(-\varepsilon_k\sum_{k=1}^{k-2}\theta_k + \sum_{k=1}^{k-2}\theta_k\varepsilon_k\right)$$

$$= -\frac{(x-\varepsilon_{k-1})^3_+}{(\varepsilon_k-\varepsilon_{k-1})}\sum_{k=1}^{k-2}\theta_k(\underset{\text{big } k}{\varepsilon_k} - \underset{\text{small } k}{\varepsilon_k})$$

$$= -\sum_{k=1}^{k-2}\theta_k(\varepsilon_k-\varepsilon_k)\left(\frac{x-\varepsilon_{k-1}}{\varepsilon_k-\varepsilon_{k-1}}\right)^3_+ \rightarrow \text{A}$$

Let us now consider the term (ii) from equation 1

$$\theta_k(x-\varepsilon_k)^3_+ = \frac{(x-\varepsilon_k)_+}{(\varepsilon_k-\varepsilon_{k-1})}(\theta_k\varepsilon_k - \theta_k\varepsilon_{k-1})$$

$\because$ Multiply and didide by $(\varepsilon_k - \varepsilon_{k-1})$

$$= \frac{(x-\varepsilon_k)_+^3}{(\varepsilon_k - \varepsilon_{k-1})} \left(\theta_k \varepsilon_k - \theta_k \varepsilon_{k-1} + \theta_{k-1}\varepsilon_{k-1} - \theta_{k-1}\varepsilon_{k-1}\right)$$

Add and subtract $(\theta_{k-1}\varepsilon_{k-1})$

$$= \frac{(x-\varepsilon_k)_+^3}{(\varepsilon_k - \varepsilon_{k-1})} \left(-\varepsilon_{k-1}(\theta_{k-1}+\theta_k) + \varepsilon_{k-1}\theta_{k-1} + \varepsilon_k\theta_k\right)$$

$$= \frac{(x-\varepsilon_k)_+^3}{(\varepsilon_k - \varepsilon_{k-1})} \left(\varepsilon_{k-1}\sum_{k=1}^{k-2}\theta_k - \sum_{k=1}^{k-2}\theta_k\varepsilon_k\right)$$

$$= (x-\varepsilon_k)_+^3 \sum_{k=1}^{k-2}\theta_k \frac{(\varepsilon_{k-1}-\varepsilon_k)}{(\varepsilon_k - \varepsilon_{k-1})}$$

Multiple and divide by $(\varepsilon_k - \varepsilon_k)$

$$= (x-\varepsilon_k)_+^3 \sum_{k=1}^{k-2}\theta_k \frac{(\varepsilon_{k-1}-\varepsilon_k)}{(\varepsilon_k - \varepsilon_{k-1})} \times \frac{(\varepsilon_k - \varepsilon_k)}{(\varepsilon_k - \varepsilon_k)}$$

Add and Subtract ε_k from the above equation and rearrange terms

$$= (x-\varepsilon_k)_+^3 \sum_{k=1}^{k-2}\theta_k \underbrace{\frac{(\varepsilon_k-\varepsilon_k)(\varepsilon_{k-1}-\varepsilon_k+\varepsilon_k-\varepsilon_k)}{(\varepsilon_k-\varepsilon_k)(\varepsilon_k-\varepsilon_{k-1})}}_{\text{Representing the above equation in a different form}}$$

$$= (x-\varepsilon_k)_+^3 \sum_{k=1}^{k-2}\theta_k (\varepsilon_k - \varepsilon_k) \underbrace{\left[\frac{1}{(\varepsilon_k-\varepsilon_{k-1})} - \frac{1}{(\varepsilon_k-\varepsilon_k)}\right]}_{\text{Taking LCM and solving will give the above equation representation}}$$

$\hookrightarrow$ B

We have equation 1

$$\sum_{k=1}^{k}\theta_k (x-\varepsilon_k)_+^3 = \sum_{k=1}^{k-2}\theta_k (x-\varepsilon_k)_+^3 + \theta_{k-1}(x-\varepsilon_{k-1})_+^3 + \theta_k (x-\varepsilon_k)_+^3$$

Substitute the values (A) and (B) in (1)

$$= \sum_{k=1}^{K-2} \theta_k (x-\xi_k)_+^3 - \sum_{k=1}^{K-2} \theta_k (\theta_k - \xi_k) \frac{(x-\xi_{K-1})_+^3}{(\xi_K - \xi_{K-1})}$$

$$+ (x-\xi_K)_+^3 \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) \left(\frac{1}{\xi_K - \xi_{K-1}} - \frac{1}{\xi_K - \xi_k} \right)$$

multiply and divide by $(\xi_K - \xi_k)$ from the first term too.

$$= \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) \frac{(x-\xi_k)_+^3}{(\xi_K - \xi_k)} - \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) \frac{(x-\xi_{K-1})_+^3}{(\xi_K - \xi_{K-1})}$$

$$+ \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) \left(\frac{(x-\xi_K)_+^3}{\xi_K - \xi_{K-1}} - \frac{(x-\xi_K)_+^3}{\xi_K - \xi_k} \right)$$

$$= \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) \left(\frac{(x-\xi_k)_+^3}{\xi_K - \xi_k} - \frac{(x-\xi_{K-1})_+^3}{\xi_K - \xi_{K-1}} + \frac{(x-\xi_K)_+^3}{\xi_K - \xi_{K-1}} - \frac{(x-\xi_K)_+^3}{\xi_K - \xi_k} \right)$$

$$= \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) \left(\frac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k} - \frac{(x-\xi_{K-1})_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_{K-1}} \right)$$

$$\Rightarrow \sum_{k=1}^{K} \theta_k (x-\xi_k)_+^3 = \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) (d_k(x) - d_{K-1}(x))$$

where, $d_k(x) = \dfrac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k}$

$d_{K-1}(x) = \dfrac{(x-\xi_{K-1})_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_{K-1}}$

Hence, the value $d_k(x)$ is as below i.e

$$\boxed{d_k(x) = \frac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k}}$$

Problem 1

Principle Component Analysis

Show first principle component minimizes the residual sum of squares

We consider the below case of a 1D projection of the datapoints. The p dimensional vectors are projected on a line through the origin.

→ Let the line be represented by vector $\vec{w}$ = unit vector

→ Let the project of data points be represented by vector $\vec{x}_i$

→ This projection on the line will give $\vec{w}\vec{x}_i$ and this is the distance of that data point from the origin.

→ The coordinate in p dimensional space is $(\vec{x}_i \cdot \vec{w})\vec{w}$

We know that the mean of projected datatypes is 0 because mean of x_i is 0

ie $\sum_{i=1}^{n} \vec{x}_i = 0$

when number of observations are n, we have below equation

$$\Rightarrow \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i \cdot \vec{w})\vec{w}$$

The residual of the projection is given by $\|\vec{x}_i - (\vec{w} \cdot \vec{x}_i)\vec{w}\|^2$

$$\|\vec{x}_i - (\vec{w} \cdot \vec{x}_i)\vec{w}\|^2 = \|\vec{x}_i\|^2 - 2(\vec{w} \cdot \vec{x}_i)(\vec{w} \cdot \vec{x}_i) + \|\vec{w}\|^2$$

$$= \|\vec{x}_i\|^2 - 2(\vec{w} \cdot \vec{x}_i)^2 + 1$$ (because w is a unit vector)

We will add the residuals of all the vectors

$$\therefore RSS(\vec{w}) = \sum_{i=1}^{n} \|\vec{x}_i\|^2 - \sum_{i=1}^{n} 2(\vec{w}\vec{x}_i)^2$$

In the above equation the first term does not depend on $\vec{w}$ and hence does not contribute to minimize the RSS.

To make RSS small we should maximize the term obtained

$$\sum_{i=1}^{n} (\vec{w} \cdot \vec{x_i})^2$$

In this case n is not depending on $\vec{w}$, so we should maximize

$$\frac{1}{n} \sum_{i=1}^{n} (\vec{w} \vec{x_i})^2$$

i.e mean of $(\vec{w} \vec{x_i})^2$

We know that the mean of a square is always equal to the square of the mean added to some variance

i.e $$\frac{1}{n} \sum_{i=1}^{n} (\vec{w} \cdot \vec{x_i})^2 = \left(\frac{1}{n} \vec{x_i} \vec{w}\right)^2 + \text{Var}[\vec{w} \cdot \vec{x_i}]$$

we initially saw that the mean of the projections is always zero. So minizing RSS will be equal to maximizing the variance of the projections

$$\frac{1}{n} \sum_{i=1}^{n} (\vec{w} \cdot \vec{x_i})^2 = \text{Var}[\vec{w} \cdot \vec{x_i}]$$

$$\therefore \boxed{\sigma_{\vec{w}}^2 = \frac{1}{n} \sum_{i=1}^{n} (\vec{w_i} \cdot \vec{x_i})^2}$$

2) a) MARS model is given by

$$f(x) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(x) \quad -① \qquad h_m(X) = \begin{cases} (x-t) & \text{if } x > t \\ 0 & \text{otherwise} \end{cases}$$

Where $h_m(x)$ is a piece wise linear basis functions.

Now consider the classifications and regressiont trees algorithm (that is CART)

the formula is :

$$f(x) = \sum_{m=1}^{M} C_m I\{x \in R_m\} \quad \text{where} \quad -②$$

Wher I is a identity function that returns 1 if x is in subset R_m

Thus we can see that MARS to be a modification of CART algorithm with a better regression setting.

Hence by replacing the peicewise linear basis functions by step identity functions $I(x-t>0)$ & $I(x-t\leq 0)$ i.e.,

$$f(x) = \sum_{m=0}^{M} \beta_m I_m(x-t) \qquad \text{where } I_m(x-t) \begin{cases} R_1 & x>t \\ R_2 & x<t \end{cases}$$

In otherwords & MARS multiplicative model h_m is replaced with interaction method I_m, a reflected step function pair.

Finally, To get a Binary tree representation of, the step function should be restricted not to split more than more than once.

By following the two steps, we can modify MARS method to behave like a decision tree.

Scanned by CamScanner

(b) Since MARS use piece-wise ~~lee~~ linear basis functions, they are more powerful regression compared to identity step functions. Hence, MARS can ~~of~~ express ~~b~~/ represent better the underlying data distribution. better in comparision to binary ~~de~~ trees. Hence for very high dimensional inputs which al common ~~are~~. in real world applications, MARS method is a better regressor model.

At the same time, the process of adding ~~the~~ a basis function to regressor model in MARS is a very computationally expensive procedure. It can be shown that for 'N' data points & 'p' predictors and m backfitting algorithm cycles,

(i) trees take $2(pN\log N)$ operations. (worst case $pN\log N + N^2p$)

(ii) For a M-term ~~needs~~ MARS model require $NM^3 + pM^2N$ computations, if M is reasonable fraction of N then it is very expensive.

Clearly MARS methods due to its complexity can ~~can~~ get prohibitively expensive compared to decison trees.

Problem 4 (P, 20 Points)

1. (4P) Apply best subset selection to the training set.Generate plots for R2, adjusted R2, Cp, and BIC in dependence of the number of features. What can you observe? Which model would you chose and why? Which features are used in this model? Calculate training and test error measured in MSE for this model.

   Solution. Refer to section 4.1 in the code.

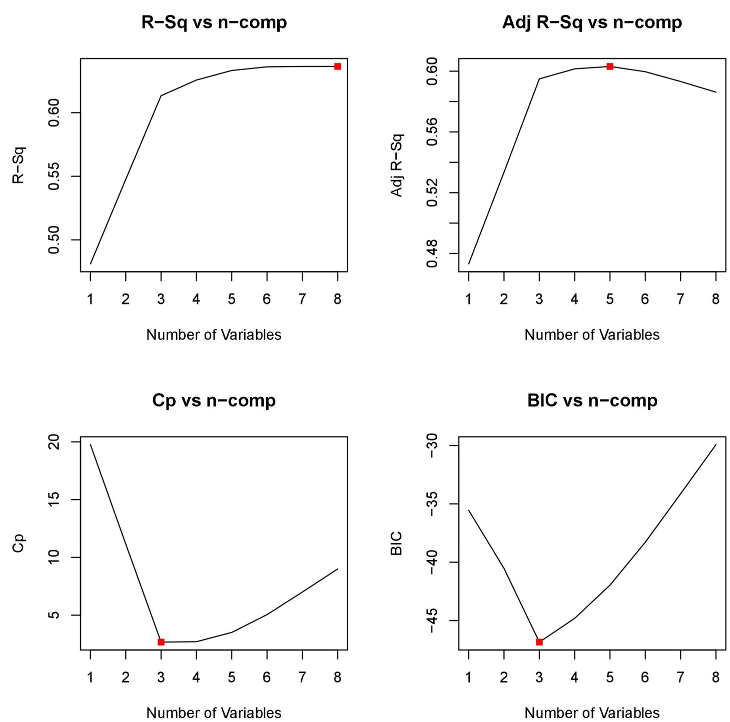


Figure 1: Subset selection plots

   In **Figure1** the best model (number of components) for each metric is indicated by red point.

   We can see that the R^2 gives the best model with all 8 components while adjusted R^2 suggests model with five components. The problem is that these evaluation metrics are not unbiased and do consider the factor of overfitting. Hence, the other two evaluation metrics, Cp and BIC are unbiased evaluators and they also penalize the model based on number of predictors leading to a simpler model. This is evident since the model suggested by both Cp and BIC are with three components. In conclusion, the model chosen is the one with three features.

   The features selected are: **"lcavol", "lweight" and "svi"**.

   For the chosen model, the train error is **0.5040965** and the test error is **0.4497825** ■

2. (4P) Fit principal components regression models for M = 1 ,...,8. Plot the train and test error against the number of principal components M. What can you observe?

Solution. Refer to section 4.2 in the code.

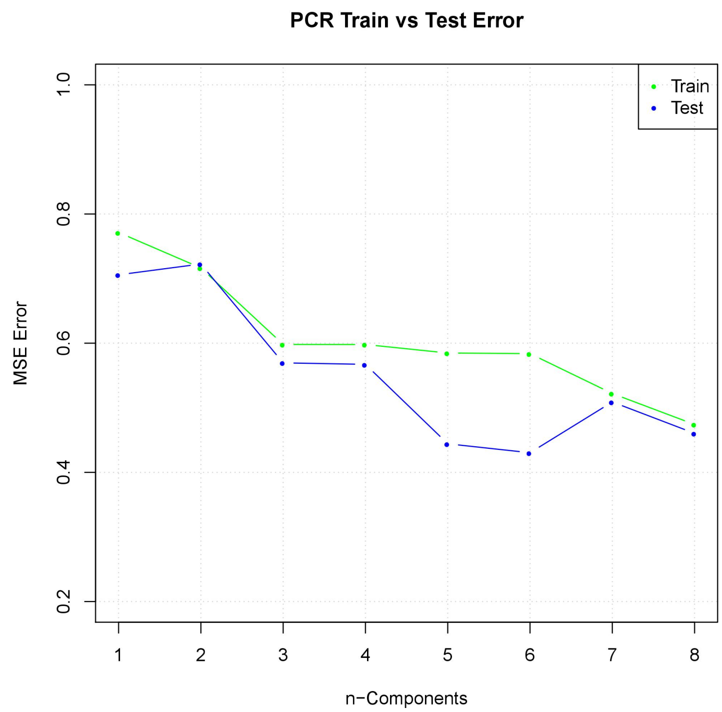


Figure 2: PCR error plots

■

We can observe from the **Figure2** that as the number of components increase the corresponding train and test error decreases. This is because the model as the number of components increases the amount of variation captured from original data also increases.

3. (4P) Fit partial least squares models for M = 1,...,8. Plot the train and test error against the number of directions M. What can you observe? Compare to the results you obtained when using PCA.

   Solution. Refer to section 4.3 in the code.

   We can observe from the **Figure3** that as the number of components increases the error decreases in this case as well similar to PCR. The main difference between the two methods is the loss at initial components. Clearly, the the first three components of PLS capture much more variation in the data than PCR method as the components increase both the methods converge to the same value. Hence, PLS is a better fit than PCR method.

■

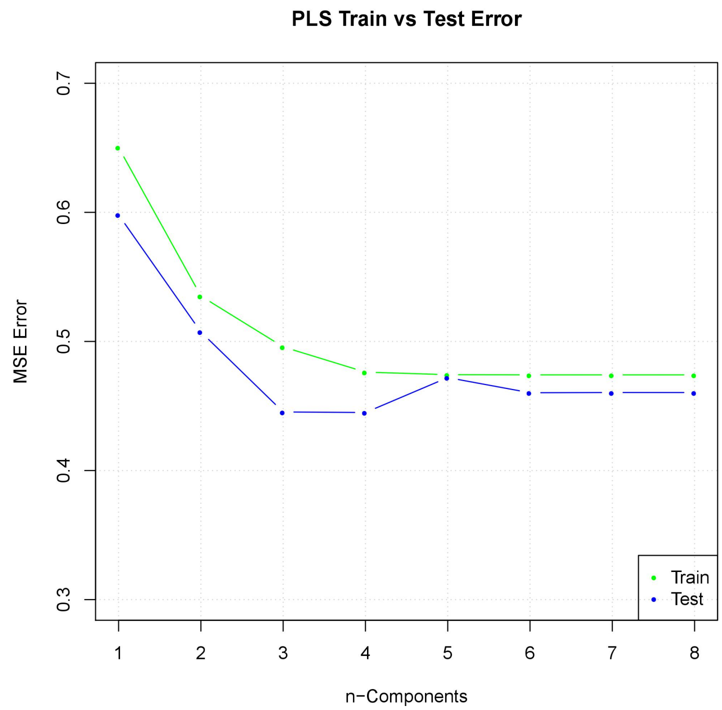
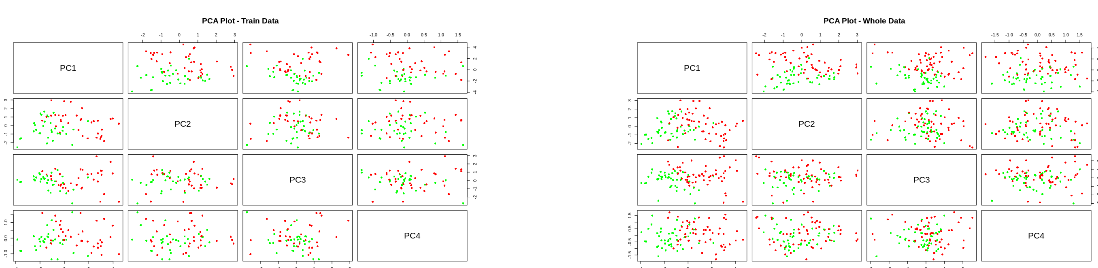


Figure 3: PLS error plots

4. (3P) Visualize the whole data set (combining training and test data) and the training data only projected on the first four principal components (using the scores obtained by PCA). Color the data points according to their lpsa value: Set a threshold at 2.5, all samples with an lpsa below should be colored in one color, all other samples in a different color. What can you observe?

Solution. Refer to section 4.4 in the code.



(a) Train data

(b) Test + Train data

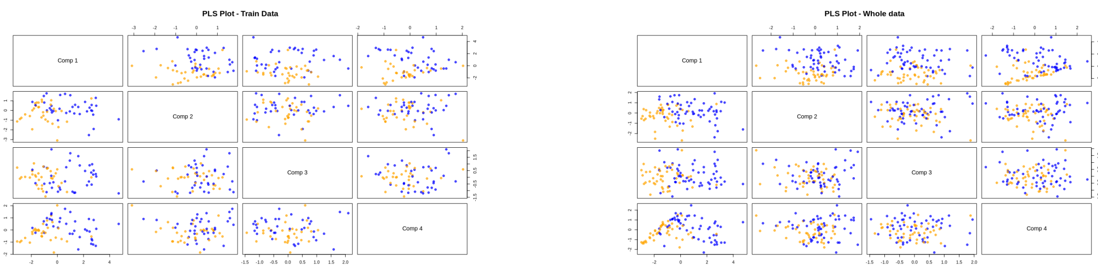
Figure 4: PCA plots for first four components

From the **Figure4** it is evident in both train data and the complete data that the PCA components clearly separate the data with the threshold on the response variable *lpsa* >= 2.5. In addition, we can also observe that the first component is very crucial in drawing the decision boundary.

■

5. 3P) Perform the same visualization task using the first four PLS directions. Compare the resulting plots to the PCA plots.

Solution. Refer to section 4.5 in the code.



(a) Train data

(b) Test + Train data

Figure 5: PLS plots for first four components

From the **Figure5** we can conclude that separation of data based PLS compoenets with *lpsa* thresholding is relatively difficult compared to PCA.

■

6. (2P) Explain the role of M in the bias-variance trade-off. Which model would you choose for PCR and PLS, respectively?

Solution. The number of components in PCA and PLS plays a very important role in data bias-variance trade-off. As the number of components increase, the PCA/ PLS model begins to represent the original least square model and thus becomes more biased. In order to get a unbiased model, we need to use fewer components that capture the data variance sufficiently. By reducing the number of principal components, variance increases.

For the PCR model from the **Figure2** we can see that the test error is least for n-components equal to six. Unfortunately this model is almost same as using all the components (increasing the chances of a biased model). As an good engineering practice, we can pick the model to the left of the least value and is within the 1-standard error, from this the best model would be with three components.

For the PLS model, the least error is when the n-components is three. Since the error the left this point is quite high and the model with three components is less complex. This can be chosen as the best model. ■