3 Dimension Reduction Methods

So far we have controlled variance of estimates in two ways:

① Use a subset of original variable
- best subset, forward/backward selection, lasso

② shrinking coefficients towards zero
- ridge regression, lasso

These methods all defined using original predictor variables $X_1, \dots, X_p$.

We now explore a class of approaches that

① transform predictors

② then fit least squares regression model using transformed variables

We refer to these techniques as *dimension reduction* methods.

① Let $Z_1, \dots, Z_M$ represent $M<p$ linear combinations of our original predictors.

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

for constants ϕ_{jm}, $m=1, \dots, M$.

② Fit linear regression model using least squares

$$y_i = \theta_0 + \boxed{\sum_{m=1}^{M} \theta_m z_{im}} + \varepsilon_i, \quad i=1, \dots, n$$

regression coefficients.

If ϕ_{jm} chosen well, this can outperform least squares.

The term *dimension reduction* comes from the fact that this approach reduces the problem of estimating $\underbrace{p+1}_{\beta_0, \beta_1, \ldots, \beta_p}$ coefficients to the problem of estimating $\underbrace{M+1 \text{ coefficients}}_{\theta_0, \theta_1, \ldots, \theta_M}$ where $M < p$.

z = linear combination of x's

Note: $\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \boxed{\sum_{m=1}^{M} \theta_m \phi_{jm}} x_{ij}$

$= \sum_{j=1}^{p} \boxed{\beta_j} x_{ij}$

Dimension reduction serves to constrain β_j, since now they must take a particular form.

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

$\Rightarrow$ special case of original linear regression problem

with β_j constrained $\to$ can bias our coefficient estimates

$\to$ if $p > n$ (or $p \approx n$), selecting $M \ll p$ can reduce variance.

All dimension reduction methods work in two steps.

1. transformed predictors are obtained.
2. model is fit using M transformed predictors.

The selection of ϕ_{jm}'s can be done in multiple ways.

↳ We will talk about 2 ways.

First way to choose ϕ_{jm}'s $\Rightarrow Z_1,\dots,Z_M$.

3.1 Principle Component Regression

Principal Components Analysis (PCA) is a popular approach for deriving a low-dimensional set of features from a large set of variables.

PCA is an unsupervised approach for reducing the dimension of a $n \times p$ data matrix X.

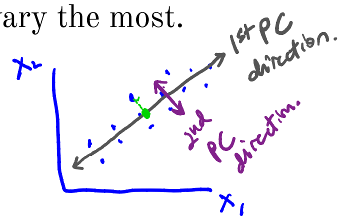
① Finding PC directions
② projecting data onto those directions

The *first principal component* directions of the data is that along which the obervations vary the most.

The 1st principal components are obtained by projecting the data onto the 1st PC direction.

↳ A point is projected onto a line by finding the point on the line closest to the original point.

$p=2$



1st PC direction = direction along which data varies most = line closest to all observations! (least squares line).

out of every possible linear combination of X_1 and X_2 such that $\phi_{11}^2 + \phi_{21}^2 = 1$, choose linear combination such that

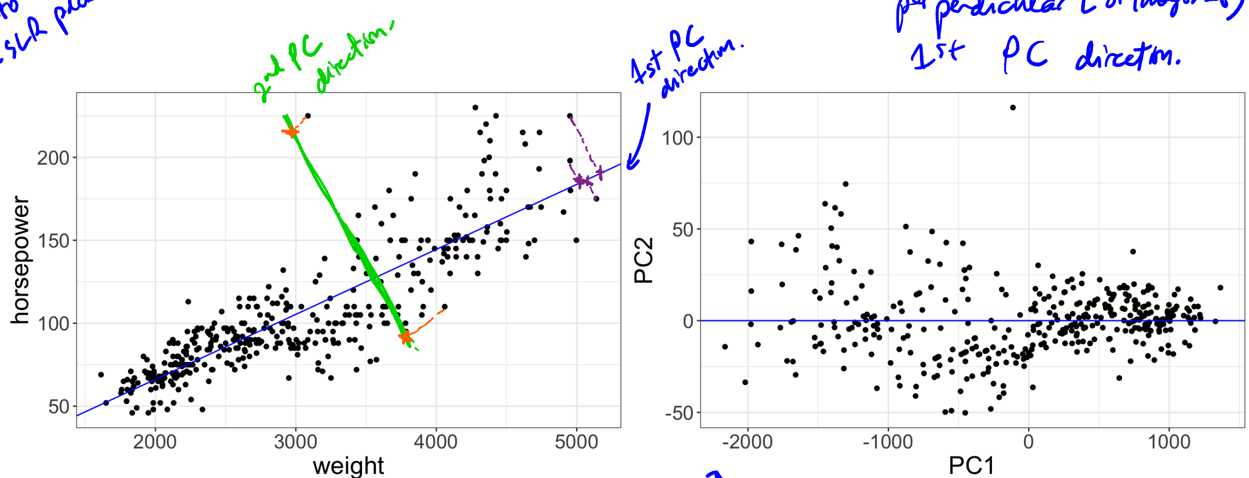
$$\text{Var}\left[\boxed{\phi_{11}}(X_1-\bar{X}_1) + \boxed{\phi_{21}}(X_2-\bar{X}_2)\right] \text{ is maximized.}$$

$\Rightarrow z_{i1} = \phi_{11}(x_{1i}-\bar{x}_1) + \phi_{21}(x_{2i}-\bar{x}_2)$ for $i=1,\dots,n$ are "1^{st} principal component scores"

We can construct up to p principal components, where the 2nd principal component is a linear combination of the variables that are uncorrelated to the first principal component and has the largest variance subject to this constraint.

$\Rightarrow$ 2^{nd} PC direction is perpendicular (orthogonal) to 1^{st} PC direction.

From Auto data in ISLR package.



projected onto 1^{st} + 2^{nd} PC directions

The 1^{st} PC contains the most information $\longrightarrow$ p^{th} PC contains the least.

The Principal Components Regression approach (PCR) involves

1. Construct first M principal components $Z_1, \ldots, Z_M$

2. fit linear regression model predicting Y using $Z_1, \ldots, Z_M$ by least squares.

Key idea: often a small # of principal components will suffice to explain most of variability in the data X, as well as the relationship with the response.

In other words, we assume that the directions in which $X_1, \ldots, X_p$ show the most variation are the directions that are associated with Y.

This is not guaranteed to be true, but often works well in practice.

If this assumption holds, fitting PCR will lead to better results than fitting least squares on $X_1, \ldots, X_p$.

(We can mitigate over fitting).

How to choose M, the number of components?

M is tuning parameter,
use C.V.

Note: PCR is not feature selection!

Z_i depend on all X's.

PCR is not sparse model

PCR more like ridge than lasso.

3.2 Partial Least Squares

The PCR approach involved identifying linear combinations that best represent the predictors $X_1, \dots, X_p$.

$Z_1, \dots, Z_M$ $\quad Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$

Consequently, PCR suffers from a drawback

Explains var in X, not necessarily Y.

Alternatively, *partial least squares (PLS)* is a supervised version.

$Z_m = \sum \phi_{jm} X_j$

Determine ϕ with both X and Y

Roughly speaking, the PLS approach attempts to find directions that help explain both the reponse and the predictors.

ϕ, linear combinations

The first PLS direction is computed,

1. Std'ize X $\quad var(X_j) = 1$
2. Set ϕ_{j1} according to simple regression $Y \sim X_j$
   $\phi_{j1} = \beta$
   Places more weight on predictive X's

To identify the second PLS direction,

1. Regress each $X_j \sim Z_1$, take residuals
2. Repeat above process, but with residuals instead of X's.

As with PCR, the number of partial least squares directions is chosen as a tuning parameter. M is a tuning parameter

use C.V. to find M.

Neither PCR nor PLS is "better"

4 Considerations in High Dimensions

Most traditional statistical techniques for regression and classification are intendend for the low-dimensional setting.

$n >> p$. Throughout history, low dimension was most prevalent.

Ex: y = blood pressure, X = age, gen, bmi $p = 3$
$n = 100$

In the past 25 years, new technologies have changed the way that data are collected in many fields. It is not commonplace to collect an almost unlimited number of feature measurements.

Ex: Blood pressure. Now, we may have X's for every SNP (gene sequencing)
$p >> n$

Ex: shopping patterns
X for every other purchase
X for shopping cards, wish list
$p >> n$

Data sets containing more features than observations are often referred to as *high-dimensional*.

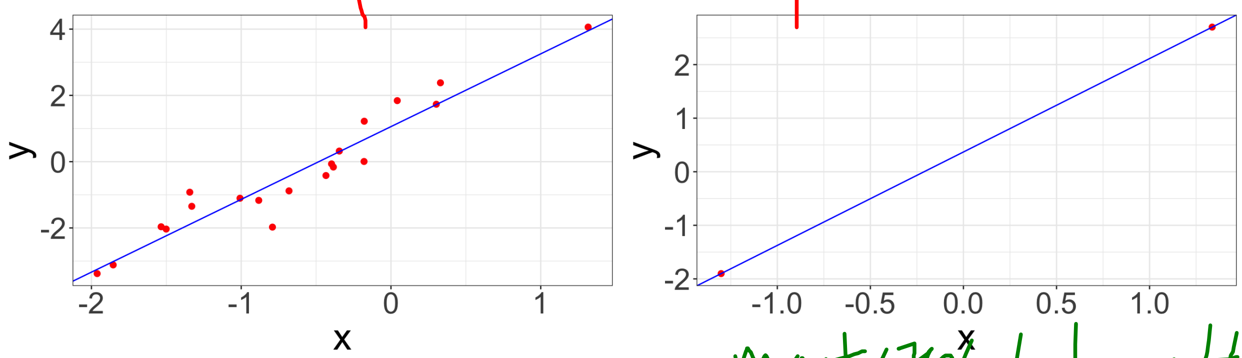
Least squares + other standard methods may not work here.
Need to be careful when $n \approx p$ or $n << p$

What can go wrong in high dimensions?

$n > p$

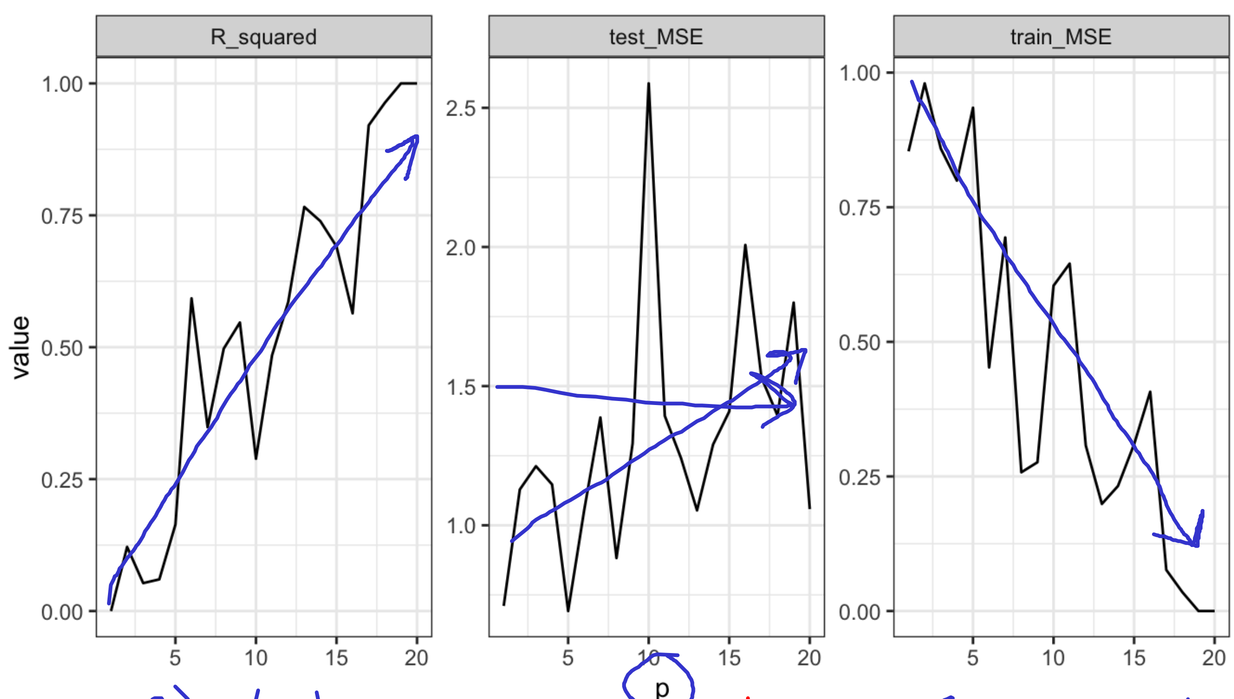
residuals $= 0$!

$n = p = 2$ (intercept)



most likely bad results on test data

1. Be careful! 2. Always check test set performance



R^2 looks good!

Test set looks meh

Training set looks good!

Many of the methds that we've seen for fitting *less flexible* models work well in the high-dimension setting.

1. Regularization or shrinkage (lasso, ridge)
2. Appropriate tuning parameter (C.V.)
3. Test error ↑ p ↑ unless new predictors are totally associated w/ y. ← curse of dimensionality

Adding predictors will improve training perf. at the cost of more variance. ⇒ test error ↑
Risk of overfitting

When we perform the lasso, ridge regression, or other regression procedures in the high-dimensional setting, we must be careful how we report our results.

In high dimensional setting, it is more likely that variables will be highly correlated.
⇒ any variable in the model could be written as a linear combination of other variables in the model.

This means we can never really know if any variables are <u>truly</u> predictive of the response.
⇒ we can never identify which are <u>best</u> variables to include.

at best, we can only hope to assign large regression coefficients to variables that are highly correlated to variables that are truly predictive of the response.

⇒ When we use lasso/feature selection, etc. we should be clear that we have identified one of <u>many possible</u> models for predicting the response and should be <u>validated on many independent data sets</u>. (reproducibility)

also important to report <u>test</u> errors (not R^2, training errors) because we knows R^2 ↑ as p ↑ but this doesn't mean we have a good model.