3 Dimension Reduction Methods

So far we have controlled variance in two ways:

(1) Used a subset of original variables
- best subset, forward/backward selection, lasso

(2) shrinking coefficients towards zero
- ridge regression, lasso.

These methods all defined using original predictor variables $x_1, \ldots, x_p$.

We now explore a class of approaches that

(1) transform the predictors

(2) then perform least squares using transformed variables.

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

(1) Let $z_1, \ldots, z_M$ represent $M < p$ linear combinations of our original predictors.

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

for constants $\phi_{1m}, \ldots, \phi_{pm}$ $m = 1, \ldots, M$

(2) Fit the linear regression model using least squares

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

(θ_0, θ_m: regression coefficients.)

If $\{\phi_{jm}\}_{\substack{j=1,\ldots,p \\ m=1,\ldots,M}}$ chosen well, this can outperform least squares.

The term *dimension reduction* comes from the fact that this approach reduces the problem of estimating $p+1$ coefficients to the problem of estimating $M+1$ coefficients where $M < p$.

$\beta_0, \beta_1, \ldots, \beta_p$

$\theta_0, \theta_1, \ldots, \theta_M$

NOTE:

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \left[\sum_{j=1}^{p} \phi_{jm} x_{ij} \right] = \sum_{j=1}^{p} \left[\sum_{m=1}^{M} \theta_m \phi_{jm} \right] x_{ij}$$

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

from least squares

have to come from somewhere else (hopefully we've picked well).

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)

↳ it can bias coefficient estimate

↳ 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 (get $\{\phi_{jm}\}_{\substack{j=1,\ldots,p \\ m=1,\ldots,M}}$)

(2) Model is fit using M transformed predictors from (1).

(linear)

selection of $\{\phi_{jm}\}_{\substack{j=1,\ldots,p \\ m=1,\ldots,M}}$ can be done in multiple ways.

We will talk about 2.

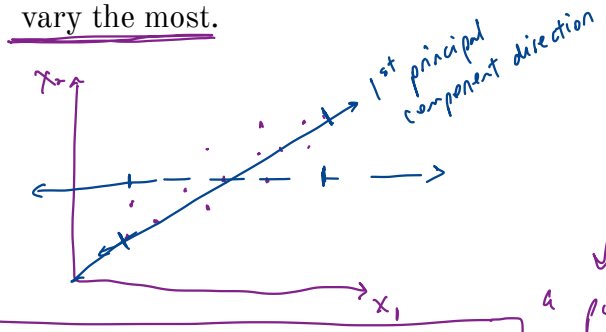
One way to choose $Z_1, \ldots, Z_M$

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.

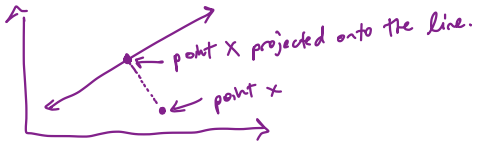
PCA is an unsupervised approach for reducing the dimension of an $n \times p$ data matrix X.

The *first principal component* direction 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 principal component direction.

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

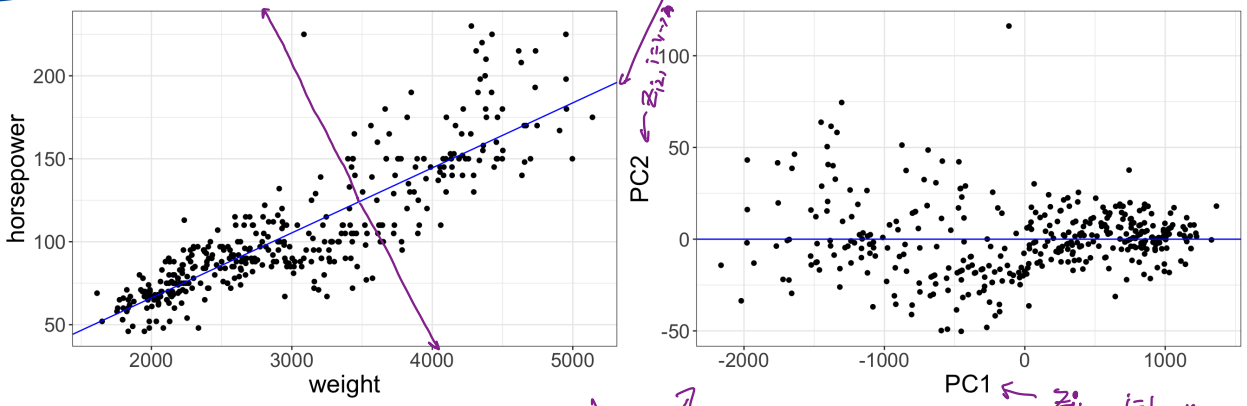


out of every possible linear combination of X_1 and X_2 such that $\phi_{11}^2 + \phi_{21}^2 = 1$, choose ϕ_{i1} s.t. $\text{Var}\left(\phi_{11}(X_1 - \bar{X}_1) + \phi_{21}(X_2 - \bar{X}_2)\right)$ is maximized.

$Z_{11} = \phi_{11}(x_{1i} - \bar{x}_1) + \phi_{21}(x_{2i} - \bar{x}_2)$ for $i = 1, \ldots, n$ are called 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.

$\Rightarrow$ 2nd PC direction is perpendicular to 1st PC direction.



1st PC direction = dimension along which data vary the most.

projected onto principal component directions.

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

The Principal Components Regression approach (PCR) involves

1. Construct first M principal components $Z_1, \ldots, Z_M$ (score vectors). — a choice we are making.
2. Fit a linear regression model w/ $Z_1, \ldots, Z_M$ as predictors using least squares.

Key idea: Often a small # of PC suffice to explain most of the variability in the data, as well as the relationship w/ predictor.

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 model on $X_1, \ldots, X_p$ because we can mitigate overfitting.

How to choose M, the number of components?

M can be thought of as a tuning parameter
$\Rightarrow$ use CV method to choose!

as $M \uparrow p$, PCR $\rightarrow$ least squares. $\Rightarrow$ bias $\downarrow$ but variance $\uparrow$, will see bias-variance tradeoff in the form of a U-shape in the test MSE.

Note: PCR is not feature selection!

each of the M principal components used in the linear regression is a linear combination of all p of the original predictors!

$\Rightarrow$ while PCR works well to reduce variance, it doesn't give us a sparse model.

PCR more like ridge regression than the lasso. (not going to help w/ interpretation)

NOTE: recommended standardizing predictors $X_1, \ldots, X_p$ to each have st. dev. = 1 before getting the PCs.

can do this using pls function in pls package.

3.2 Partial Least Squares

directions

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

We identified these directions in an unsupervised way (response Y is not used to help us determine the directions).

Consequently, PCR suffers from a drawback

There is no guarantee that the direction that best explains the predictors will also be the best directions to explain the relationship btw predictors and response.

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

(1) identify new features $Z_1, \ldots, Z_m$ linear combination of original predictors

(2) fit linear model (least squares) using transformed predictors.

i.e. uses Y & X to find $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$ $m=1,\ldots,M$.

PLS also going to use Y (not just X) to find linear combinations of $X_1, \ldots, X_p$

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

linear combinations

The first PLS direction is computed,

(1) standardize the p predictors (all have st. dev. $=1$).

(2) set each ϕ_{j1} equal to the coefficient from a simple linear regression $Y \sim X_j$

Since the coefficient from SLR of $Y \sim X_j$ $\propto$ cor(Y, X_j) PLS places highest weight on variables that are strongly related (linearly) to the response.

To identify the second PLS direction, $X_j \sim Z_1$

(1) regress each variable $X_1, \ldots, X_p$ on Z_1 and get residuals ($r_{ji} = X_{ji} - \hat{X}_{ji}$ $i=1,\ldots,n$ $j=1,\ldots,p$)

(2) Compute Z_2 by setting each ϕ_{j2} equal to coefficient from SLR $Y \sim r_j$ ← residuals from step (1)

The residuals $r_1, \ldots, r_p$ ≈ remaining information not explained by 1st PLS direction.

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

Generally standardize the predictors and response before performing PLS.

In practice, PLS usually performs no better than ridge or PCR.

↳ supervised nature of problem does reduce, but often increases variance. ⇒ not always better.

4 Considerations in High Dimensions

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

This is because throughout the history of the field, the bulk of scientific problems requiring statistics have been low dimensional.

e.g. ag field trials.

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. (p very large).

But n can still be limited due to cost, sampling availability, etc.

e.g. could predict BP on age, gender, BMI, and also collect half million SNPs

now $p \approx 500,000$ but SNPs are expensive to collect, maybe only get ~ 200 of them.

e.g. Consider trying to model online shopping patterns. We could treat all search terms in a person's month-long browsing history as features in a "bag-of-words" model.

But we may only only have a few thousand people who have consented to use their history.

For a given use the features would be absence (0) or presence (1) of each potential search term. $\Rightarrow$ p large but $n \approx 2000$

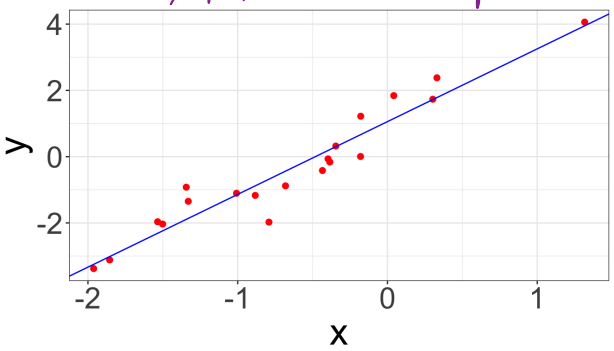
Data sets containing more features than observations are often referred to as *high-dimensional.* $p > n$

classical approaches (like least squares) are not appropriate in this setting. (why? think bias-variance trade-off and overfitting).

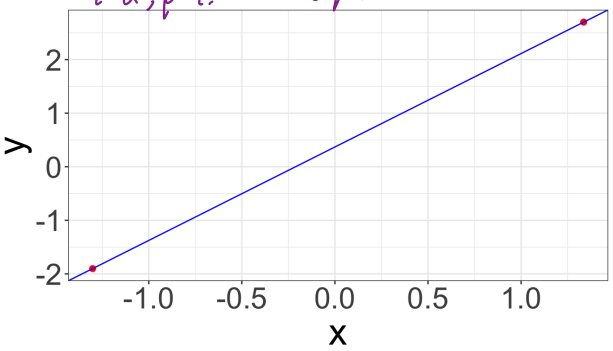
$\Rightarrow$ we need to be careful when $n \approx p$ or $n < p$.

What can go wrong in high dimensions? Going to talk about least squares but same issues arise w/ logistic regression or LDA

If $p \approx n$ or $p > n$ regardless of if there is a relationship w/ response least squares will yield a set of coefficients that is an (almost if $n < p$ but $n \approx p$) perfect fit. $p = n$. $\Rightarrow$ residuals $= 0$.



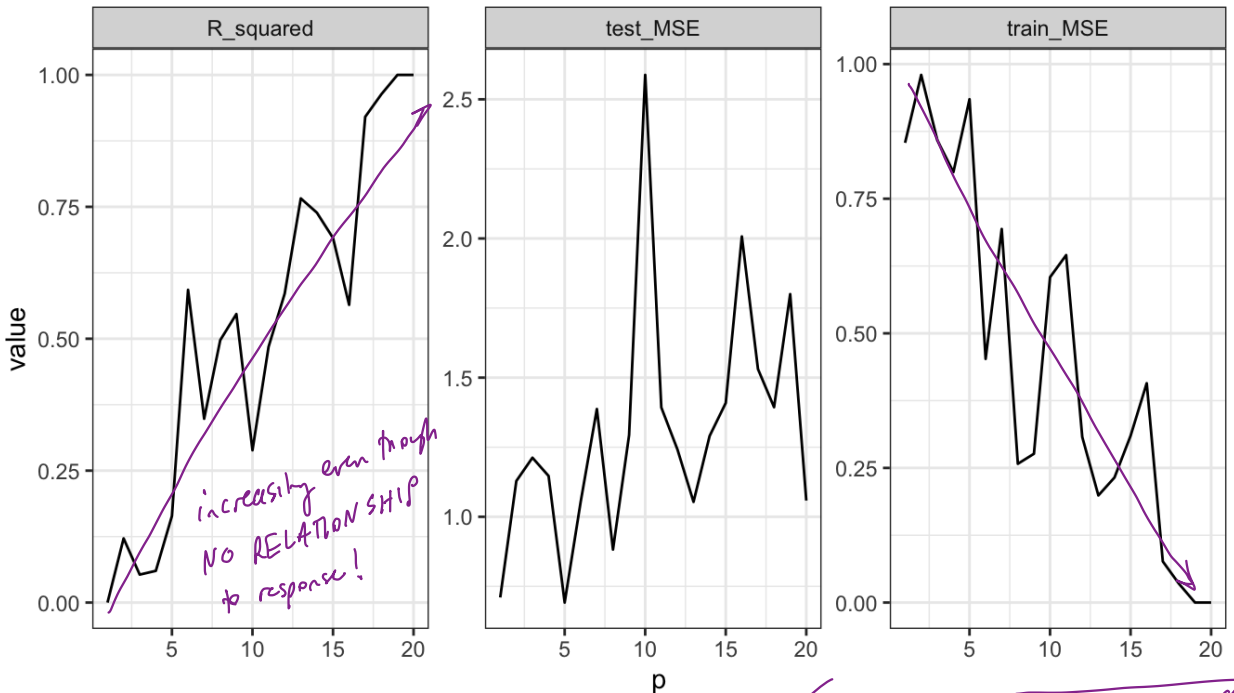
$n = 20$, $p = 1$ $n > p$ => not perfect fit.



$n = 2$, $p = 1$ (+ intercept). $n =$ # coefs => perfect fit => overfitting.

The resulting model will most likely have poor predictive performance on test data.

Simulated data w/ $n = 20$ and regression performed with between 1 and 20 features. features generated w/ no relationship to response



increasing even though NO RELATIONSHIP to response!

training MSE decreases w/ more predictors even though no relationship to response

test MSE never very good b/c not a good predictive fit (no relationship)

=> we need to be very careful when analysing data w/ many predictors. Always need to evaluate model performance on indep test set

Note: we did see methods for adjusting R^2 to better estimate test MSE: C_p, AIC, BIC, adj R^2

But: in high dim settings, we can't compute. (Linear models).

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

Key points

1. regularization or shrinkage plays a key role in high dimensional problems.

2. appropriate tuning parameter selection is critical for good predictive performance.

3. The test error tends to ↑ as p↑ UNLESS the additional features are truly associated w/ response.

This is related to the curse of dimensionality

adding noise will deteriorate our fitted model => ↑ test error
vs. adding signal features will improve our model (fitted).

(↑ dimension => ↑ risk of overfitting due to noise.)

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 predictors will be highly correlated
=> any variable in the model could be written (almost) as a linear combination of other variables in the model.

This means we can never really know if any are truly predictive of the response.
=> we can never identify which are the best 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 many possible models for predicting the response and should be validated on many independent test data sets.

✱ Also important to report test errors (not R^2, training errors, etc), because R^2↑ as p↑ but doesn't mean we have a good model.