Chapter 6: Linear Model Selection & Regularization

In the regression setting, the standard linear model is commonly used to describe the relationship between a response Y and a set of variables $X_1, \ldots, X_p$.

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$$

typically fit w/ least squares

Upcoming: more general models (non-linear)

The linear model has distinct advantages in terms of inference and is often surprisingly competitive for prediction. How can it be improved?

replace least squares w/ alternative fitting procedures.

We can yield both better *prediction accuracy* and *model interpretability*:

prediction accuracy: if true relationship is ≈ linear => least squares will have low bias.

If $n >> p$ => also have low variance => perform well on test data!

If n not much larger than p => high variability => poor performance.

If $p > n$ => no longer have a unique solution => variance = ∞ => cannot be used at all!

goal: reduce variance without adding too much bias.

model interpretability: often many variables used in a regression are not associated w/ response.

By removing (setting $\hat{\beta}_i = 0$), we can obtain a more easily interpretable model.

Note: least squares will hardly ever result in $\hat{\beta}_i = 0$.

=> need variable selection.

Same ideas apply to logistic regression.

1 Subset Selection

We consider methods for selecting subsets of predictors.

1.1 Best Subset Selection.

To perform *best subset selection*, we fit a separate least squares regression for each possible combination of the p predictors. $\binom{p}{2} = \frac{p(p-1)}{2}$ models with exactly 2 predictors, etc.

Algorithm:

1. Let $\mathcal{M}_0$ denote the model with no predictors.
2. For $k = 1, \ldots, p$
   (a) Fit all $\binom{p}{k}$ models that contain k predictors.
   (b) Pick the best of those (call it $\mathcal{M}_k$). "Best" is defined by $\downarrow$RSS ($\uparrow R^2$).
3. Select a single best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using CV error, C_p, AIC/BIC, or adjusted R^2 (traditional metrics, more later.)

Why can't we use R^2 for step 3? as $p\uparrow$, $R^2\uparrow$ always. Why might we not want to do this at all? computation. Fitting 2^p models! $p = 10 \Rightarrow$ 1000 models.

We can perform something similar with logistic regression.

1.2 Stepwise Selection

For computational reasons, best subset selection cannot be performed for very large p. → "impossible" with $p \geq 40$.

Best subset may also subset when p large because w/ a large search space can find good models in training that perform poorly on test data
$\Rightarrow$ high variability & overfitting can occur.

Stepwise selection is a computationally efficient procedure that considers a much smaller subset of models.

Forward Stepwise Selection: start with no predictors and add one predictor at a time until all predictors are in the model, choose the "best" from these.

1. Let $\mathcal{M}_0$ denote the null model - no predictors
2. For $k = 0, \ldots, p-1$
   (a) Consider $p-k$ models that augment the predictors in $\mathcal{M}_k$ w/ 1 additional predictor.
   (b) Choose the best among these $p-k$ and call it $\mathcal{M}_{k+1}$ ($\uparrow R^2$).
3. Select a single best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using CV error, C_p, AIC/BIC, or adjusted R^2.

Now we fit $1 + \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$ models!

Backward Stepwise Selection: Begin w/ full model and take predictors away one at a time until you get to the null model.

1. Let $\mathcal{M}_p$ denote the full model, contains all predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) consider all models (k) that contain all but one of the predictors in $\mathcal{M}_k$ ($k-1$ predictors).
   (b) Choose the best among them, call it $\mathcal{M}_{k-1}$ ($\uparrow R^2$).
3. Select the single best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using CV, C_p, AIC/BIC, or adjusted R^2.

* Neither forward nor backwards stepwise selection are guaranteed to find the best model containing a subset of the p predictors.

Forward Selection can be used when $p > n$ (but only up to $n-1$ predictors (not up to p)).

1.3 Choosing the Optimal Model

Best subset, forward selection, backward selection all need a way to pick the "best" model – according to test error.

- RSS & R^2 are proxy for training error $\Rightarrow$ not estimates of test error
  - ↳ (1) estimate this directly (CV) or
  - (2) adjust training errors for model size.

(2) $C_p = \frac{1}{n}\left(\text{RSS} + 2d\hat{\sigma}^2\right)$ — $\hat{\sigma}^2$: estimate of variance of ε (full model).

RSS: subset model. d: # predictors in subset model.

adds a penalty to training error (RSS) to adjust for underestimation of test error.

(choose model w/ lowest value).

(2) AIC & BIC can get for models fit w/ MLE

$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\text{RSS} + 2d\hat{\sigma}^2\right).$$

$$\text{BIC} = \frac{1}{n\hat{\sigma}^2}\left(\text{RSS} + \log(n)\, d\hat{\sigma}^2\right).$$

choose model w/ lowest AIC or BIC.

↑ $\log(n) > 2$ for $n > 7$ $\Rightarrow$ heavier penalty on models w/ many variables $\Rightarrow$ results in smaller models.

(2) Adjusted R^2 (only for least squares).

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$ Always ↑ as d ↑

$$\text{Adj } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$$ choose model w/ highest Adj R^2.

(1) Validation and Cross-Validation

Directly estimate test error w/ Validation or CV and choose model w/ lowest estimated error.

Very general (can be used for any model) even when it's not clear how many "predictors" we have.

Now have fast computers $\Rightarrow$ these are preferred.

2 Shrinkage Methods

The subset selection methods involve using least squares to fit a linear model that contains a subset of the predictors. As an alternative, we can fit a model with all p predictors using a technique that constrains (*regularizes*) the estimates.

↳ shrinks estimates towards zero.

Shrinking the coefficient estimates can significantly reduce their variance!

Help us to avoid overfitting!

2.1 Ridge Regression

Recall that the least squares fitting procedure estimates $\beta_1, \ldots, \beta_p$ using values that minimize

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2$$

residual sum of squares.

$\hat{\beta}^R$

Ridge Regression is similar to least squares, except that the coefficients are estimated by minimizing

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

note we are not penalizing β_0
we want to penalize the relationships, not the intercept
(mean value of response when $x_{i1} = \cdots = x_{ip} = 0$).

$\lambda \geq 0$ tuning parameter (determine separately from fitting).

trades off 2 criteria: minimize RSS to fit data well

minimize $\lambda \sum_{j=1}^{p} \beta_j^2$ "shrinkage penalty" will be small when β_j close to zero $\Rightarrow$ shrink estimates towards zero.

The tuning parameter λ serves to control the impact on the regression parameters.

When $\lambda = 0$, penalty has no effect $\Rightarrow$ ridge regression = least squares.

As $\lambda \to \infty$, impact of penalty grows $\Rightarrow$ $\hat{\beta}^R \to 0$.

Ridge regression will produce a different set of coefficients for each penalty ($\hat{\beta}^R_\lambda$).

Selecting a good λ is critical! How to choose? Cross validation!

The standard least squares coefficient estimates are scale invariant.

Multiply X_j by a constant c leads to a scaling of OLS coef estimate by a factor of $\frac{1}{c}$.

$\Rightarrow$ regardless of how the j^{th} predictor is scaled $X_j \hat{\beta}_j$ will remain the same.

In contrast, the ridge regression coefficients $\hat{\beta}_\lambda^R$ can change substantially when multiplying a given predictor by a constant.

e.g. say we have an income variable in (1) dollars vs. (2) thousands of dollars

due to the sum of squared coef. term, this change will not simply cause the coefficient to change by a factor of 1000.

$\Rightarrow$ $X_j \hat{\beta}_{j,\lambda}^R$ depends not only on λ, but also on the scaling of X_j.

(may even depend on scaling of other predictors)

Therefore, it is best to apply ridge regression *after standardizing the predictors* so that they are on the same scale:

i.e. have standard deviation of one.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

st. dev. of j^{th} predictor.

(1) standardize data

(2) tune model to choose λ (via cross validation).

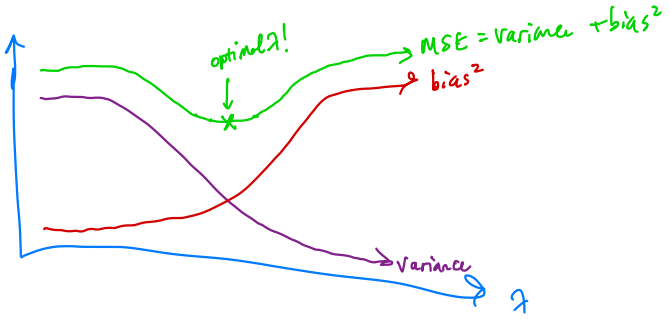
(3) fit ridge regression on standardized data w/ chosen λ.

Why does ridge regression work?

Because of the bias-variance trade-off!

As $\lambda \uparrow$, the flexibility of the ridge regression fit $\downarrow$

$\Rightarrow$ variance $\downarrow$ and bias $\uparrow$



In situations where relationship between response & predictors is $\approx$ linear

OLS will have low bias.

when p almost as large as $n \Rightarrow$ OLS will have high variability!

if $p > n$ least squares doesn't even have a unique solution.

ridge regression can still perform well in these scenarios by trading off a small amount of bias for a decrease in variance.

↳ Ridge regression works best in high variance scenarios.

Also:

Cost advantage over subset selection

because for fixed λ, only fitting one model! (very fast model to fit).

Ridge regression improves predictive performance.

Does it also help us with interpretation?

2.2 The Lasso

Ridge regression does have one obvious disadvantage.

Unlike best subset, forward or backward selection, ridge regression will include all p variables in the final model.

penalty $\lambda \sum \beta_j^2$ will shrink all $\beta_j \to 0$ but $\beta_j \neq 0$ (unless $\lambda = \infty$)!

This may not be a problem for prediction accuracy, but it could be a challenge for model interpretation when p is very large.

We will always have all variables in the model, whether there is a relationship w/ Y or not.

"Least absolute shrinkage and selection operator"

The *lasso* is an alternative that overcomes this disadvantage. The lasso coefficients $\hat{\beta}_\lambda^L$ minimize

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

ℓ_1 penalty

($\sum \beta_j^2$ = "ℓ_2 penalty").

As with ridge regression, the lasso shrinks the coefficient estimates towards zero.

ℓ_1 penalty also has the effect of forcing some coefficients to be exactly zero when λ is sufficiently large.

$\Rightarrow$ much like our selection methods, lasso performs variable selection!

As a result, lasso models are generally easier to interpret.

The lasso yields sparse models – models w/ only a subset of the variables.

Again selecting a good λ is critical. Use cross validation!

Why does the lasso result in estimates that are exactly equal to zero but ridge regression does not? One can show that the lasso and ridge regression coefficient estimates solve the following problems

$\Leftrightarrow$ equivalent to our formulation w/ λ

lasso: minimize $\left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \right\}$ subject to $\sum_{j=1}^{p} |\beta_j| \le s$

ridge: minimize $\left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 \right\}$ subject to $\sum_{j=1}^{p} \beta_j^2 \le s$

} constrained optimization problems.

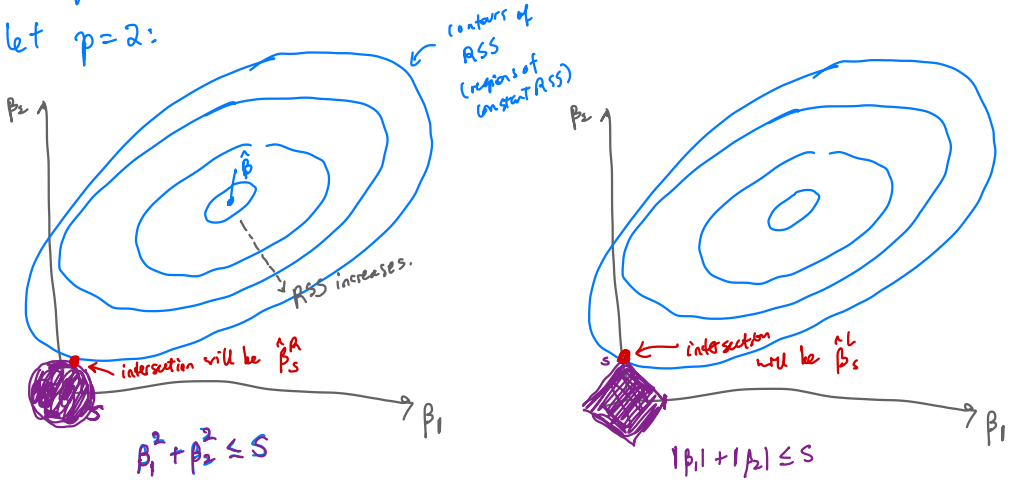
constraints

In other words, when we perform the lasso we are trying to find the set of coefficient estimates that lead to the smallest RSS, subject to the contraint that there is a budget s for how large $\sum_{j=1}^{p} |\beta_j|$ can be.

When s is very large, this is not much of a constraint $\Rightarrow$ coef estimates can be large (same for ridge).

But why does the lasso result in coefficient estimates exactly equal to 0?

let $p = 2$:



Solution to lasso or ridge is the first point in the ellipses (RSS) contact the constraint region.

Ridge has a circular region $\Rightarrow$ no sharp points, intersection won't generally occur on the axis.

Lasso corners on each axis $\Rightarrow$ ellipse often intersects at the axis $\Rightarrow$ one of the coefficients to equal zero.

If we believe there are predictors that do not have a relationship w/ Y (we just don't know which ones) lasso will perform better.

If not (every thing is important), ridge will perform better.

Use CV to pick!

2.3 Tuning

We still need a mechanism by which we can determine which of the models under consideration is "best".

For subset we have C_p, AIC/BIC, adjusted R^2, CV error. ↙ equivalent

For both the lasso and ridge regression, we need to select λ (or the budget s).

How? penalization parameter

0. scale data.
1. choose a grid of λ values
2. Compute CV error (k-fold) for each λ.
3. Select λ for which CV error is smallest
4. Refit chosen model using all available observations and selected λ.

* Note * still important to scale variables $x_1, \ldots, x_p$ for lasso to have st. dev. $= 1$.

3 Dimension Reduction Methods

So far we have controlled variance in two ways:

(1) Using 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 $X_1, \dots, X_p$.

We now explore a class of approaches that

(1) transform the predictors

(2) fit least squares using the 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 $\phi_{1m}, \dots, \phi_{pm}$ $m = 1, \dots, M$.

(2) fit the linear regression model using least squares

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

(θ_0, θ_m: regression coefficients)

If choose ϕ_{jm} well, this can outperform least squares (w/ original data).

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 $M < p$.

$\beta_0, \beta_1, \ldots, \beta_p$ $\qquad$ $\theta_0, \theta_1, \ldots, \theta_M$

Note:

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

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

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 model (with β_j constrained).

↳ can introduce bias to coefficient estimates

↳ 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 (ϕ_{jm} are obtained).

② fit model using M transformed predictors from ①.

The selection of ϕ_{jm}'s can be done in multiple ways. We will talk about 2.

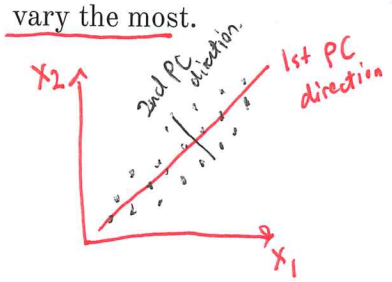
3.1 Principle Component Regression

how to choose $z_1, \dots, z_M$ (one way).

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 data matrix X $(n \times p)$.

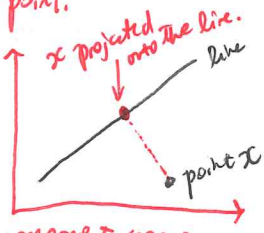
The *first principal component* directions of the data is that along which the observations vary the most.



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

↓

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



out of every possible linear combination of X_1 and X_2 such that ϕ_{11}^2 and $\phi_{12}^2 = 1$, choose such that

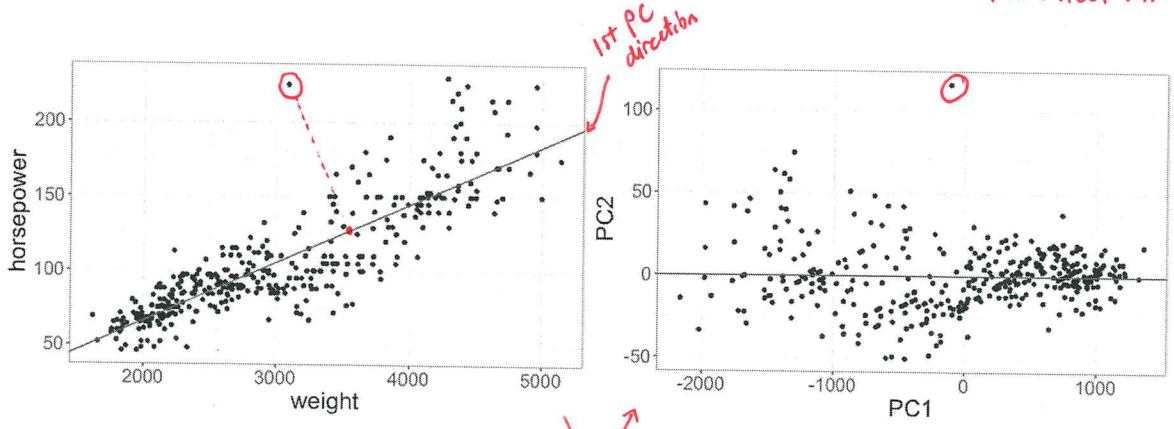
$\text{Var}\left(\phi_{11}(x_1 - \bar{x}_1) + \phi_{12}(x_2 - \bar{x})\right)$ is maximized.

$\Rightarrow z_{1i} = \phi_{11}(x_{1i} - \bar{x}_1) + \phi_{21}(x_{2i} - \bar{x}_2)$ $i = 1, \dots, n$ are 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.

$\Rightarrow$ perpendicular to 1st PC direction.

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



projected onto Principal component directions

The 1st PC contains the most information → pth PC contains the least.

The Principal Components Regression approach (PCR) involves

1. Construct first M principal components $Z_1, \ldots, Z_M$
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/ 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 to $X_1, \ldots, X_p$ 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 $\uparrow$, we will see U-shape in test MSE as a function of M.

Note: PCR is not feature selection!

each of the M PCs used in the linear regression is a linear combination of all p of the original features!

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

more like ridge than the lasso.

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

3.2 Partial Least Squares

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

We identified these directions in an unsupervised way (response Y not used to determine directions)

Consequently, PCR suffers from a drawback

There is no guarantee the directions best explains the predictors will also be the best directions to explain relationship w/ response.

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

① Identify new features $Z_1, \ldots, Z_M$ linear combinations of $X_1, \ldots, X_p$

② Fit OLS using transformed features $Z_1, \ldots, Z_M$.

PLS also uses Y (not just X) to find linear combinations of $X_1, \ldots, X_p$ (i.e. use Y & X to find $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$, $m = 1, \ldots, M$)

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 coefficient (slope) from simple linear regression $Y \sim X_j$

↳ proportional to

Since the coefficient from SLR of $Y \sim X_j$ $\propto$ $\text{Cor}(Y, X_j)$

$\Rightarrow$ PLS places highest weight on variables most strongly related to response.

To identify the second PLS direction,

① regress each predictor $X_1, \ldots, X_p$ on Z_1 and take 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 the slope coefficient from SLR $Y \sim r_j$ ← residuals from step 1.

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

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

$\Rightarrow$ CV!

Generally standardize predictors & response before performing PLS!

In practice PLS usually performs no better than ridge or PCR

4 Considerations in High Dimensions

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

this is because historically bulk of scientific problems have been low dimensional.

e.g. Think about predicting BP based on age, gender, BMI
$p = 3$, you could have thousands of patients.

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, sample availability.

e.g. rather than predicting BP on age, gender, and BMI, might also collect half a million SNPs → individual DNA mutations common in population.

Now $p \approx 500{,}000$ but expensive to collect might only get 200 of them.

$p > n$

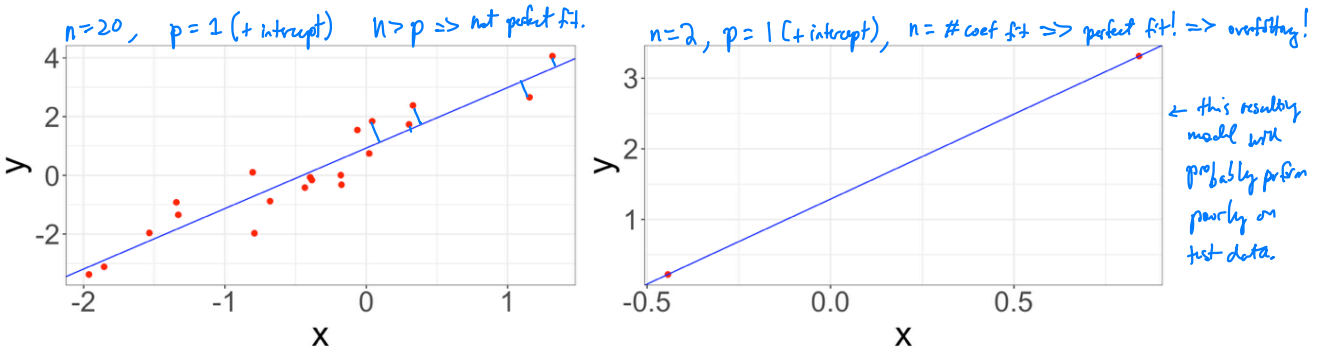
Data sets containing more features than observations are often referred to as *high-dimensional*.

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

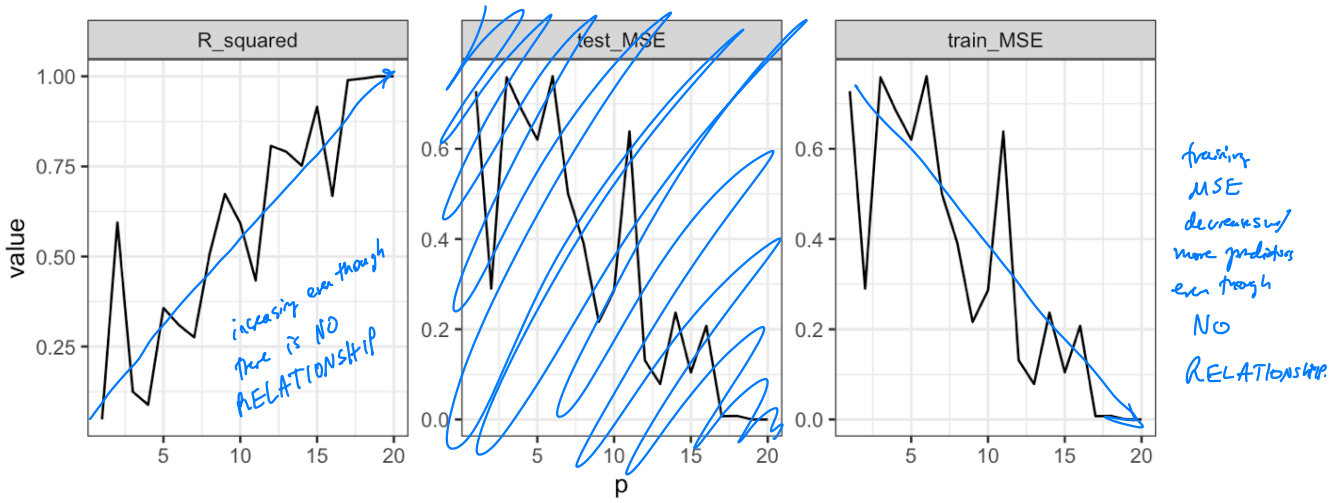
⇒ we need to be careful when $n \approx p$ or $n > p$.

What can go wrong in high dimensions? going to talk about least squares, but some issues for logistic regression or LDA.

If p is as large as or larger than n, regardless of if there is a relationship btw/ y and $\underline{X}$, we can find a perfect fit to the data $\Rightarrow$ residuals $= 0$.



Simulated data w/ $n=20$ and regression w/ between 1 and 20 features.
Features were generated w/ NO relationship to response.



We must be careful when analyzing data in high dimensions.

$\Rightarrow$ Always evaluate model performance on test set (held out data set not seen by the fitted model).

never very good b/c not a good predictive fit NO RELATIONSHIP!

Note: we saw methods to adjust training MSE to better reflect test MSE (C_p, BIC, AIC, adj R^2)
in high dim setting, we cannot compute these.

Many of the methods 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. test error tends to increase as $p \uparrow$ unless additional features are truly associated w/ response.
↑ this is due to curse of dimensionality.

adding additional signal feature will improve a fitted model but adding noise will deteriorate your fitted model $\Rightarrow \uparrow$ test error.

($\uparrow$ dimension $\Rightarrow \uparrow$ risk of overfitting due to noise looking important by chance).

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 the high dimensional setting, it's more likely that variables will be highly correlated.
$\Rightarrow$ 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 truly predictive of the response.
$\Rightarrow$ we can never identify which are the best variables to include

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

$\Rightarrow$ when we use lasso/feature selection, etc. we should be clear we have identified one of many possible models for predicting the response and should be validated on many independent test sets.

Also important to report test errors (not R^2, training errors, etc.) because we know $R^2 \uparrow$ as $p \uparrow$, but doesn't mean it's a good model.