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 $X_1, \ldots, X_p$.

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon.$$

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 with alternative fitting procedures.

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

prediction accuracy: If the true relationship is ≈ linear, least squares will have low bias

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

But if n is not much larger than p => high variability => poor performance.

If $p > n$: no longer 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 in fact associated with the response.

By removing them (setting $\hat{\beta}_i = 0$) we could 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 w/ exactly 2 predictors, etc.

Algorithm:

1. Let $\mathcal{M}_0$ denote null model - 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 ($\mathcal{M}_k$). Best is defined by ↓RSS (↑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 more later.

Why can't we use R^2 for step 3? as p↑, R^2↑ always. Why might we not want to do this? 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 suffer when p large because w/ a large search space we can find spurious good models that work on training data but perform poorly w/ test data.

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

Forward Stepwise Selection:

Start w/ no predictors and add predictors one 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 all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with 1 additional predictor.
   (b) Choose the best among those $p-k$ and call it $\mathcal{M}_{k+1}$ (↑R^2, ↓RSS).
3. Select a single best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using CV error, C_p, AIC/BIC, adj 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 we get to null model.

1. Let M_p denote the full model — contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) consider all k models that contain all predictors except one of predictors in M_k ($k-1$ predictors).
   (b) choose best among these and call it M_{k-1} (↑ R^2, ↓ RSS).
3. Select the single best model from $M_0, \ldots, M_p$ using CV error, C_p, AIC/BIC, adj 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 p!).

1.3 Choosing the Optimal Model

Need a way to pick "best" model that depends on test error (training error not a good estimate of this)
↳ either estimate this directly or adjust training errors for model size.

$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$ ← estimate of variance for full model.
↑ # of predictors in subset model

add penalty to training error (RSS) to adjust for underestimate of test error.

as d↑, C_p↑ (choose model w/ lowest value).

(gives us proportional some answer)

AIC & BIC can use for maximum likelihood fits

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

$BIC = \frac{1}{n\hat{\sigma}^2}(RSS + \log(n)d\hat{\sigma}^2)$

choose model w/ low value, since $\log(n) > 2$ for $n \geq 7$ => heavier penalty on models w/ many variables
=> results in smaller models.

Adjusted R^2 (least squares models)

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

$Adj\ R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$

choose model w/ highest adj R^2.

Validation and Cross-Validation

- Directly estimate test error w/ validation or CV and choose model w/ lowest est. error.
- Very general, can be used w/ any model, even when it's not clear how many "predictors" we have.

Now have fast computers => 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.

↳ shrink 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.

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 $x_{i1} = \ldots = x_{ip} = 0$)

$\lambda \geq 0$ tuning parameter (determined separately of the fitting procedure).

trades off 2 criteria: minimize RSS to fit data well

$\lambda \sum_{j=1}^{p} \beta_j^2$ shrinkage penalty small when β_j's close to zero $\Rightarrow$ shrinks estimates towards zero.

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

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

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

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

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

The standard least squares coefficient estimates are scale invariant.

Multiplying X_j by a constant c leads to a scaling of least squares estimates by a factor of $\frac{1}{c}$.

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

In contrast, the ridge regression coefficients $\hat{\beta}^R_\lambda$ 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.

① = 1000 × ②

due to the sum of squared coef. terms this change will not simply result in the coefficient estimate to change by a factor of 1000.

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

(may even depend on the 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. standard deviation of one.

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

s.t. dev. of j^{th} predictor.

nice in workflow, recipe

① standardize data

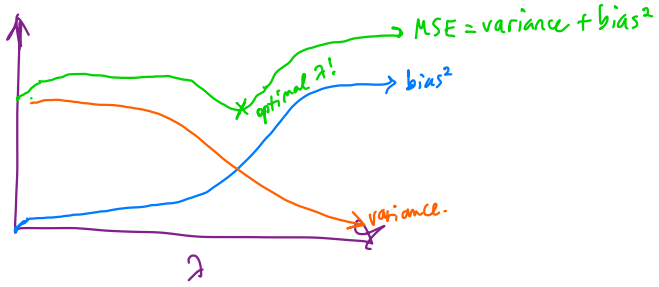
1.5 tune model to choose λ (using CV)

② fit ridge regression on training data using chosen λ.

Why does ridge regression work?

Because of the bias-variance trade-off!

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



In situations where relationship between response and predictors $\approx$ linear least squares estimate will have low bias.

When p is almost as large as n $\Rightarrow$ least squares has 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 decrease in variance.

$\Rightarrow$ ridge regression works very well in high variance scenarios.

Also
Cost advantage over subset selection
b/c for fixed λ, only fit one model! (very fast model to fit).

Ridge regression improves predictive performance.

Does it also help us w/ interpretation? No

2.2 The Lasso

Ridge regression does have one obvious disadvantage.

Unlike best subset, forward/backward selection (generally select model w/ a subset of variables) ridge regression will include all p variables in the final model.

penalty $\lambda \sum_{j=1}^{p} \beta_j^2$ will shrink $\beta_j \to 0$, 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 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

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

$\|\beta\|_1$ ℓ_1 norm.

ℓ_1 penalty

vs.

$\left(\sum_{j=1}^{p} \beta_j^2 = \text{"}\ell_2 \text{ penalty"} \right)$.

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 sufficiently large!

$\Rightarrow$ much like best subset selection lasso perform 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.

variable selection

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 other formulations w/ λ.

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

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

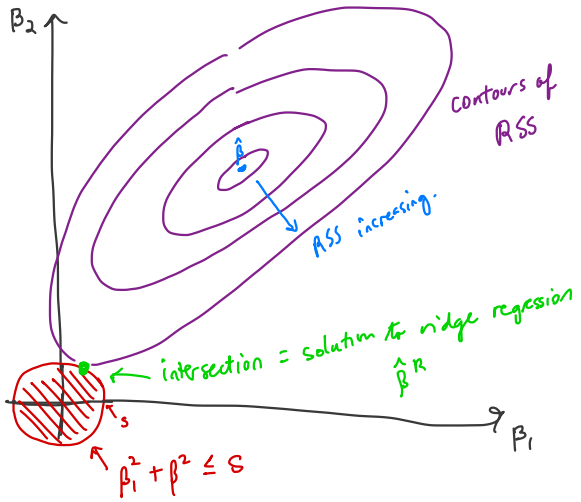
Constraints

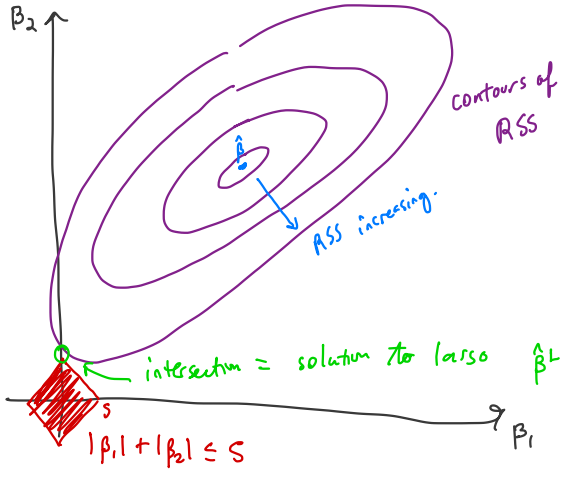
} constrained optimization problems.

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 very large. Similar for Ridge as well.

But why does the lasso result in coef. estimates exactly $= 0$? Let $p = 2$.





Solution to both ridge and lass is the first point the ellipses (RSS) contact the constraint regions.
Since ridge is a circle (no sharp points), intersection doesn't generally occur on the axis.
Lasso has corners on each axis $\Rightarrow$ ellipse often will intersect at the axis $\Rightarrow$ at least one of the coefficients equal to 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 than ridge (bias + variance)

If not (everything is important), ridge regression will perform better.

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

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

λ: penalization parameter

How? CV!

1. choose a grid of λ values.
2. Compute CV error (LOOCV or k-fold CV) for each λ
3. select λ for which ^{CV}error is smallest
4. refit model using all available training data and selected λ.

NOTE: still very important to scale variables $x_1, \ldots, x_p$ for lasso to all 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, 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 fit least squares on 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 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 $p + 1$ coefficients to the problem of estimating $M + 1$ coefficients where $M < p$.

$\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ $\qquad$ $\theta_0, \theta_1, \ldots, \theta_M$

Note:

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

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 bias coefficient estimates (trade-off lower variance, hopefully).

↳ If $p > n$ ($p \approx n$) selecting $M \ll p$ can reduce variance.

All dimension reduction methods work in two steps.

(1) transformed predictors are obtained (get ϕ_{jm}'s).

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

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

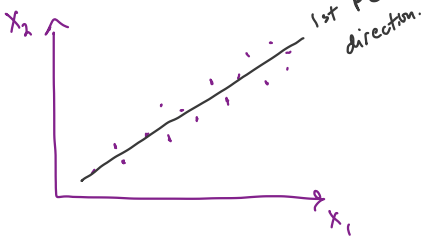
3.1 Principle Component Regression

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 an $n \times p$ data matrix.

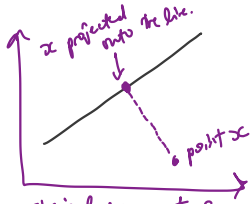
The *first principal component* directions of the data is that along which the obervations vary the most.



The 1st principal components are obtained by projecting the data onto the 1st 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 $\phi_{21}^2 = 1$ choose so that $\text{Var}\left(\phi_{11}(X_1 - \bar{x}_1) + \phi_{21}(X_2 - \bar{x}_2)\right)$ is maximized.



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

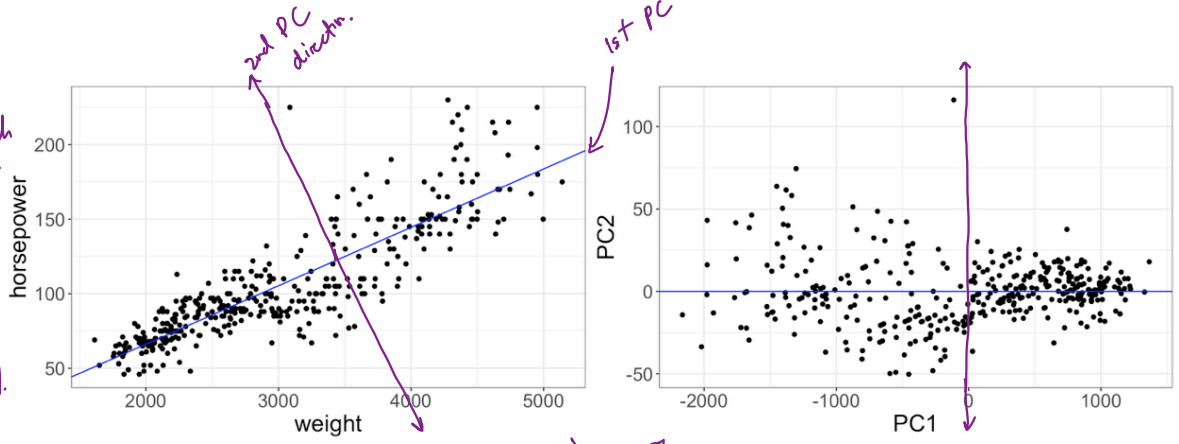
We can construct up to p principal components, where the 2nd principal component is a linear combination of the variables that are uncorrelated to the first principal component and has the largest variance subject to this constraint.

$\Rightarrow$ perpendicular to 1st PC direction!

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

=

line "closest" to all observations (least squares line!).



The 1st 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 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 X (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 on $X_1, \ldots, X_p$, because we can mitigate overfitting (lower variability).

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 $\longrightarrow$ least squares $\Rightarrow$ bias $\downarrow$ but variance $\uparrow$, will see the Ushape 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 features!

$\Rightarrow$ while PCR works well to reduce variance, it does not produce a sparse model.
(More like ridge regression than the lasso).

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

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 the directions).

Consequently, PCR suffers from a drawback

There is no guarantee that the directions that best explain the predictors will also be the best directions to explain the response.

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

(1) identify new features $Z_1, \ldots, Z_M$ linear combinations of features
(2) fit linear model (least squares) using transformed predictors.

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

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

The first PLS direction is computed,

(0) standardize the p predictors (all have st. dev $=1$).
(1) set each ϕ_{j1} equal to coefficient from simple linear regression $Y \sim X_j$

Since the coefficients from SLS of $Y \sim X_j$ $\propto$ ("proportional to") $\mathrm{Cor}(Y, X_j)$, PLS places highest weight on variables most strongly related 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 take residuals $(r_{ji} = x_{ji} - \hat{x}_{ji}, \; \substack{j=1,\ldots,n \\ j=1,\ldots,p})$.
(2) Compute Z_2 by setting each ϕ_{j2} equal to the coefficient from simple linear regression $Y \sim r_j$ ← residuals from (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 ⇒ CV!

Generally, standardize predictors AND response before performing PLS.

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

↳ supervised nature of problem does reduce bias, but also often increases variance ⇒ not always an improvement.

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 history of the field, the bulk of scientific problems requiring statistics have been low dimensional.

e.g. Think about predicting a person's BP based on age, gender, and BMI.

$p=3$, could have thousands of patients, $n \gg p$.

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 be limited due to cost.

e.g. rather than predicting BP on age, gender, BMI might also collect measurements for $\frac{1}{2}$ million SNPs $\rightarrow$ individual DNA mutations common in population

Now $p \approx 500{,}000$, but they are expensive to collect so might only have ≈ 200 of them available!

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

But we might only have access to a few hundred users who have consented to share their search history.

For a given user, features could be absense (0) or presence (1) of each potential search term.

Data sets containing more features than observations are often referred to as *high-dimensional*. p large! $n \approx 300$.

Classical approaches (like least squares) are not appropriate in this setting.

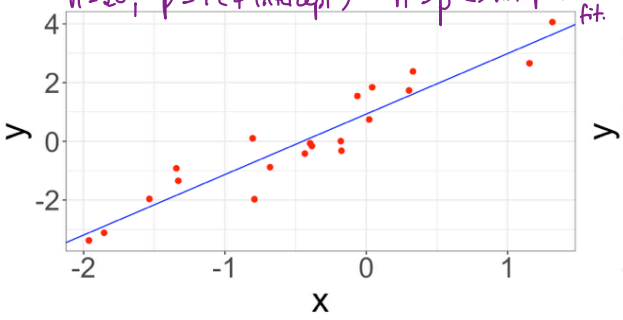
Why?

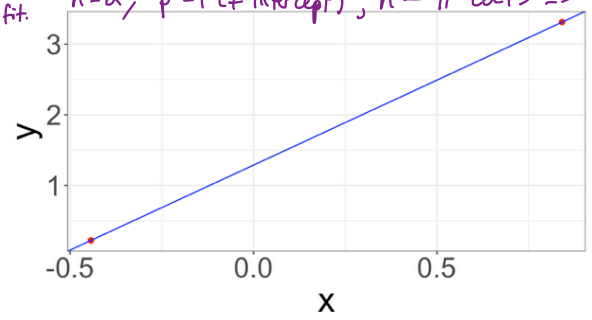
bias-variance trade off and overfitting.

$\Rightarrow$ We need to be extra careful when $n \approx p$ or $n > p$.

What can go wrong in high dimensions? (going to talk about least square, but same issues arise for logistic regression and LDA, etc)

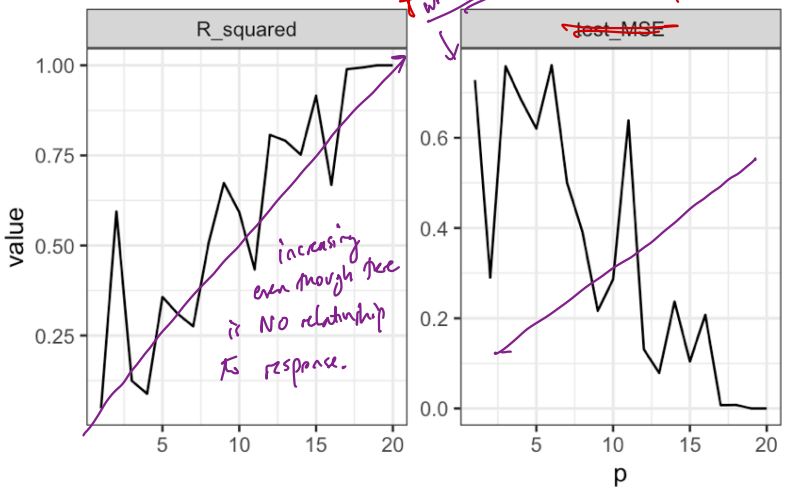
If p is as large as or larger than n, regardless of if there is a relationship between Y and X, least squares will yield a set of coefficients that result in a perfect fit to the data (residuals = 0!).





the resulting model will most likely perform poorly on training data.

Simulated data n=20 and regression performed with between 1 and 20 features. features were generated w/ NO relationship to response!



training MSE decreasing w/ more predictors even though NO relationship to response.

test MSE will show not good result because there is no relationship!

=> we must be very careful when analyzing data with many predictors.

- Always evaluate performance on independent test set (or CV).
- consider regularization, variable subset selection, dimension reduction.

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

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 increase as $p \uparrow$ unless the additional features are truly associated w/ response.
   - this is due to the curse of dimensionality
     - adding additional signal features will improve a fitted model but adding noise will deteriorate the 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 high dimensional setting, it is more likely variables will be correlated
$\Rightarrow$ some variables in the model could be written as linear combination of other variables in the model.

This means we can never really know if any vars are truly predictive of the response
$\Rightarrow$ we can't 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.

Use CV!

- $\Rightarrow$ When we use lasso / feature selection, etc. we should be clear we have identified one of many possible models for predicting the response.
  - ideally: validated on many independent test sets
- Also important to report test errors (not R^2, training error, etc). Because we know $R^2 \uparrow$ as $p \uparrow$ but this doesn't mean necessarily have a good model.