COSC 3337 : Data Science I

N. Rizk

College of Natural and Applied Sciences

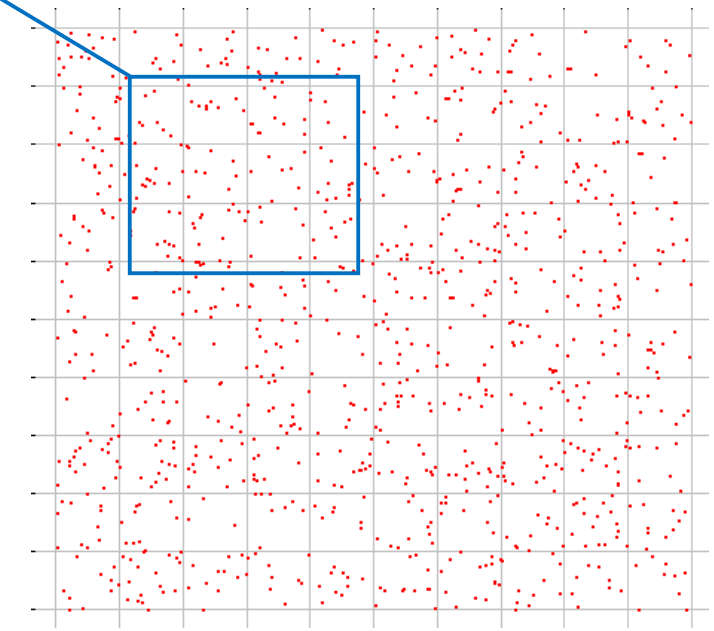
Department of Computer Science

University of Houston

Mastering
the trade-off between
bias and variance
is necessary to become a
data science champion.

Recall:Predicting from Samples

- Most datasets are **samples** from an **infinite population**.
- We are most interested in **models of the population**, but we have access only to a **sample** of it.



For datasets consisting of (X,y)

- features X + label y

a model is a prediction y = f(X)

We train on a training sample D and we denote the model as $f_D(X)$

Bias and Variance

Data-generated model $f_D(X)$ is a **statistical estimate** of the true function $f(X)$.

Because of this, it's subject to bias and variance:

Bias: if we train models $f_D(X)$ on many training sets D, bias is the expected difference between their predictions and the true y's.

i.e. $$Bias = \mathrm{E}[f_D(X) - y]$$

E[] is taken over points X and datasets D

Variance: if we train models $f_D(X)$ on many training sets D, variance is the variance of the estimates:

$$Variance = \mathrm{E}\left[\left(f_D(X) - \bar{f}(X)\right)^2\right]$$

Where $\bar{f}(X) = \mathrm{E}[f_D(X)]$ is the average prediction on X.

Bias and Variance Tradeoff

There is usually a bias-variance tradeoff caused by model complexity.

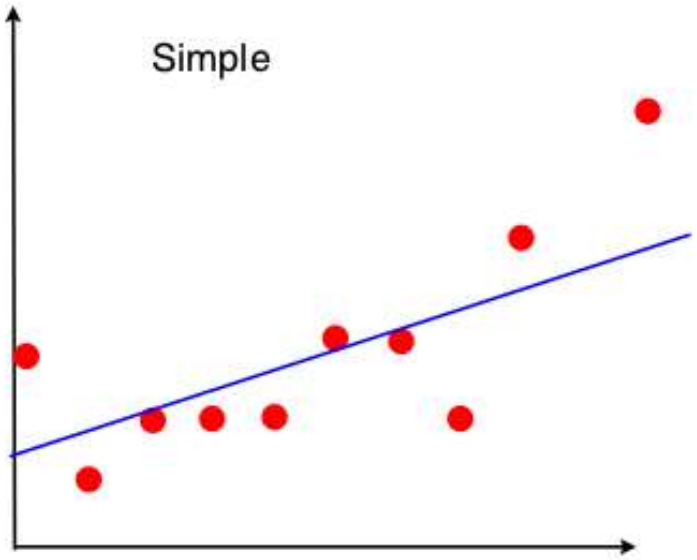
Complex models (many parameters) usually have lower bias, but higher variance.

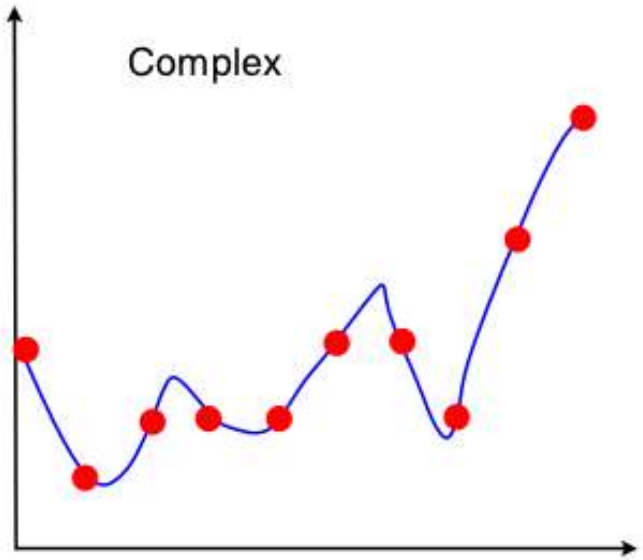
Simple models (few parameters) have higher bias, but lower variance.

Under-fitting **Over-fitting**.

Bias and Variance Tradeoff

e.g. a linear model can only fit a straight line. A high-degree polynomial can fit a complex curve. But the polynomial can fit the individual sample, rather than the population. Its shape can vary from sample to sample, so it has high variance.





higher bias, but lower variance

lower bias, but higher variance

Bias and Variance Tradeoff

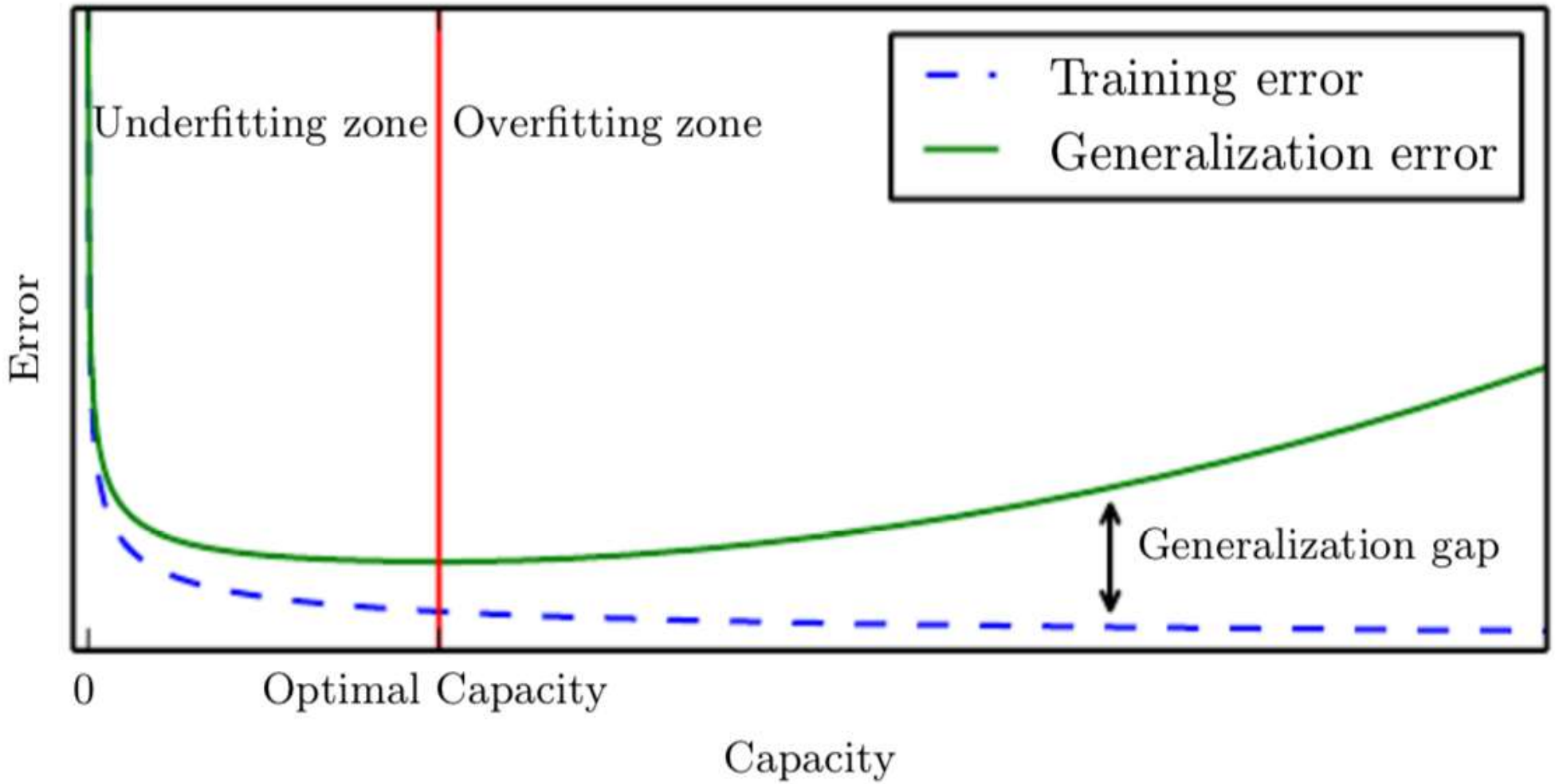
The total expected error is

$$Bias^2 + Variance$$

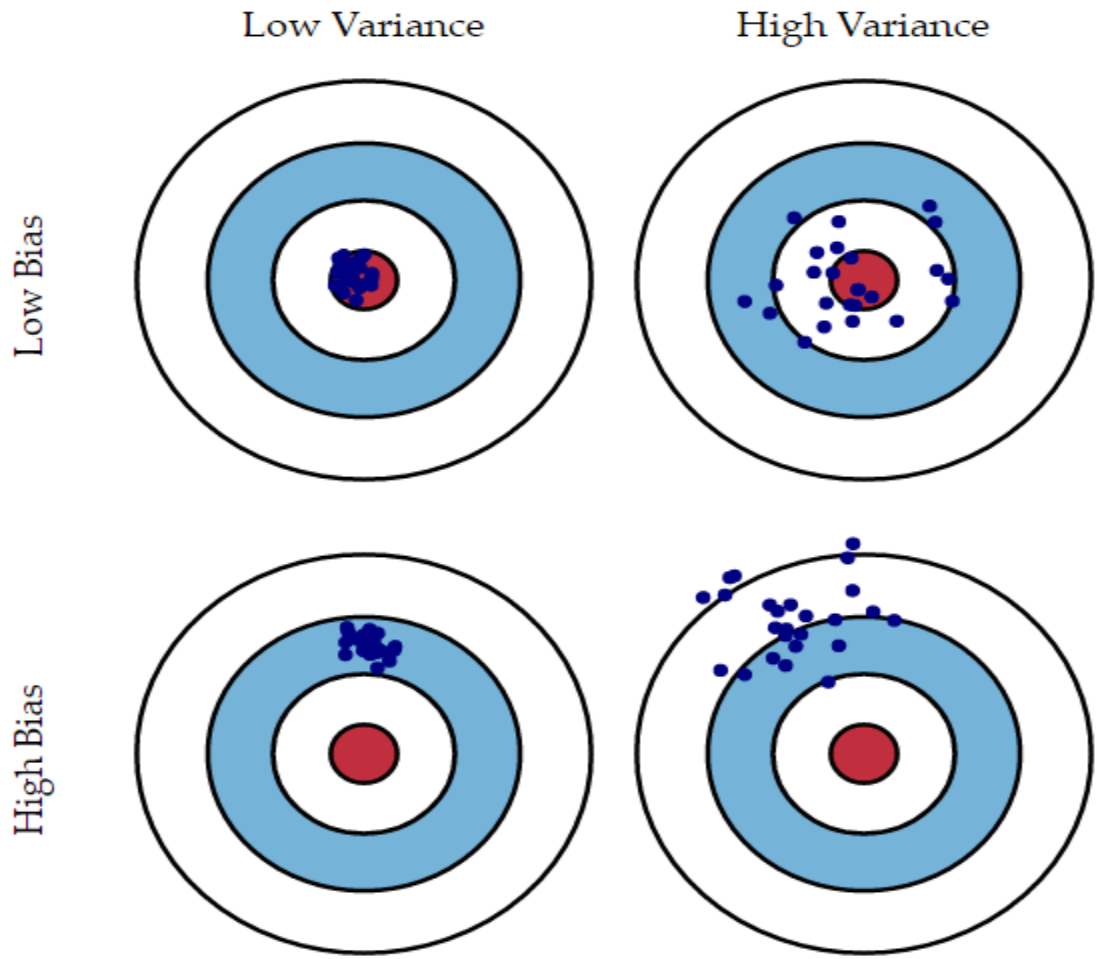
Because of the bias-variance trade-off, we want to **balance** these two contributions.

If $Variance$ strongly dominates, it means there is too much variation between models. This is called **over-fitting**.

If $Bias$ strongly dominates, then the models are not fitting the data well enough. This is called **under-fitting**.

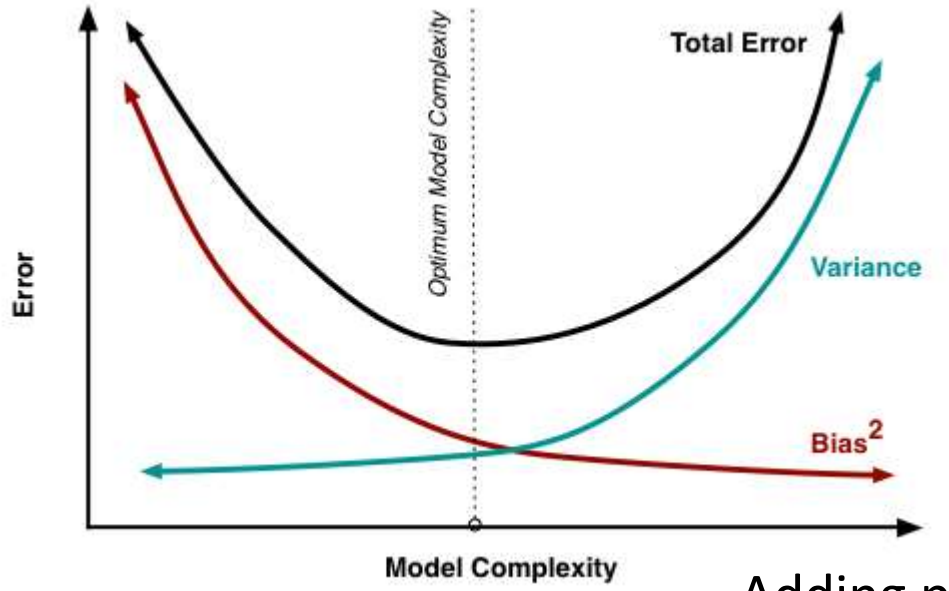

Underfitting zone
Overfitting zone
Training error
Generalization error
Error
Generalization gap
0
Optimal Capacity
Capacity

Goal ? Low variance, low Bias



The challenge lies in finding a method for which both the variance and the squared bias are low.

How ?=➔



Adding parameters to our model, its complexity increases, ➔increasing variance and decreasing bias, i.e., overfitting.

In practice, there is no analytical way to find out one optimum point in our model where the decrease in bias is equal to increase in variance.

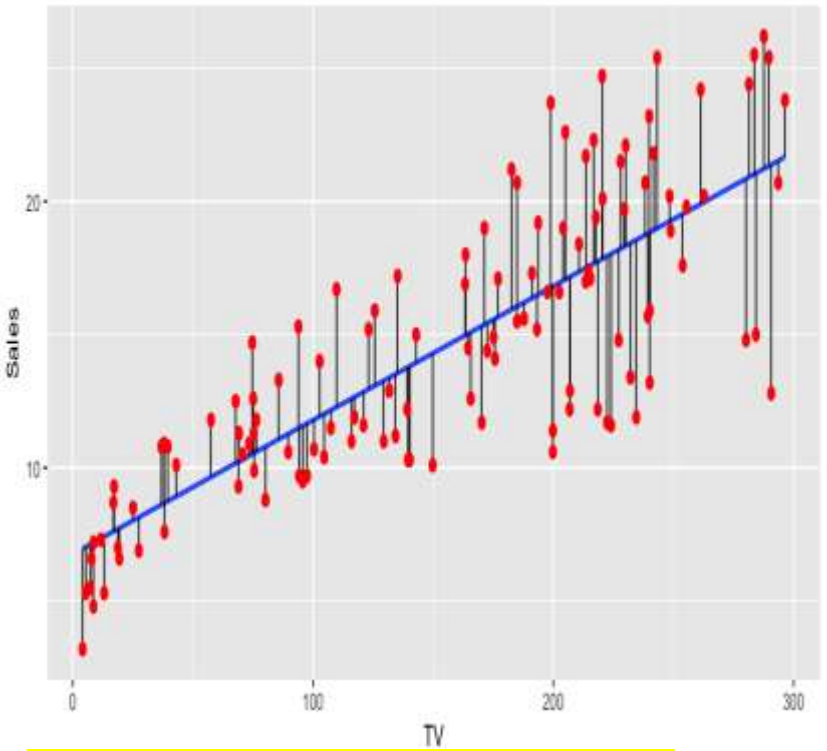
Basically there are two methods to overcome overfitting,

1. Reduce the model complexity (e.g PCA)
2. Regularization

Regression models and evaluation of the fit

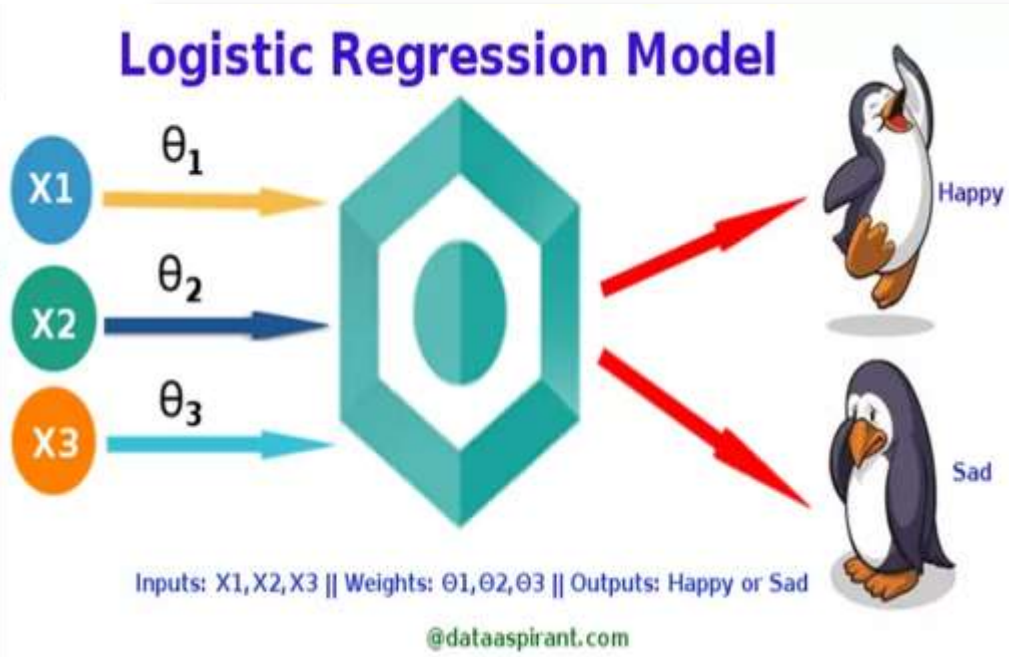
Ridge_Lasso

LINEAR REGRESSION



Best Linear Unbiased Estimators

- There must be linear relationship between independent and dependent variables
- Multiple regression suffers from multicollinearity, autocorrelation, heteroskedasticity.
- Linear Regression is very sensitive to Outliers. It can terribly affect the regression line and eventually the forecasted values.
- Multicollinearity can increase the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model. The result is that the coefficient estimates are unstable
- In case of multiple independent variables, we can go with forward selection, backward elimination and step wise approach for selection of most significant independent variables.



•it is widely used for **classification problems**

•Logistic regression doesn't require linear relationship between dependent and independent variables. It can handle various types of relationships because it applies a non-linear log transformation to the predicted odds ratio

•To avoid overfitting and underfitting, we should include all significant variables. A good approach to ensure this practice is to use a step wise method to estimate the logistic regression

•It requires **large sample sizes** because maximum likelihood estimates are less powerful at low sample sizes than ordinary least square

•The independent variables should not be correlated with each other i.e. **no multicollinearity**. However, we have the options to include interaction effects of categorical variables in the analysis and in the model.

•If the values of dependent variable is ordinal, then it is called as **Ordinal logistic regression**

•If dependent variable is multi class then it is known as **Multinomial Logistic regression**.

Assessing the Fit of Regression Models

A well-fitting regression model results in predicted values close to the observed data values.

Three statistics are used in Ordinary Least Squares (OLS) regression to evaluate model fit:

1. R-squared **and Adjusted R-squared:**
2. the overall F-test,
3. and the Root Mean Square Error (RMSE).

All three are based on two sums of squares: Sum of Squares Total (SST) and Sum of Squares Error (SSE). SST measures how far the data are from the mean, and SSE measures how far the data are from the model's predicted values.

R-squared **and Adjusted R-squared:**

R-squared is Dividing The difference between SST and SSE by SST. It is the proportional improvement in prediction (the goodness of fit of the model)
R-squared has the useful property that its scale is intuitive: it ranges from zero to one (0= No improvement in prediction, 1=perfect prediction)

One pitfall of R-squared can only increase as predictors are added to the regression model.
What if adding predictors is not actually improving the model's fit➔ Adjusted R-squared (proportion of total variance explained by the model)

Adjusted R-squared will decrease as predictors are added if the increase in model fit does not make up for the loss of degrees of freedom.
It will increase as predictors are added if the increase in model fit is worthwhile. Adjusted R-squared should always be used with models with more than one predictor variable.

The overall F-test

F-test evaluates the null hypothesis that all regression coefficients are equal to zero versus the alternative that at least one is not.

An equivalent null hypothesis is that R-squared equals zero. A significant F-test indicates that the observed R-squared is reliable.

the F-test determines whether the proposed relationship between the response variable and the set of predictors is statistically reliable

The Root Mean Square Error (RMSE).

The RMSE is the square root of the variance of the residuals.

It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values.

Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit.

RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.

Mixed models, generalized linear models, and event history models, use maximum likelihood estimation.

Assessing the Fit of Regression: Maximum likelihood estimation (MLE)

is a method of **estimating** the parameters by maximizing a **likelihood** function, so that under the assumed statistical model the observed data is most probable. ...

Likelihood as a function of θ given the data observed

$$L_x(\theta) = P(x|\theta).$$

Coin: product of probability

$$L_X(\theta) = \prod_{x \in X} f(x|\theta)$$

Log-likelihood is the sum over the log of the likelihood for each point.

$$l_X(\theta) = \sum_{x \in X} log[f(x|\theta)].$$

The purpose of MLE is to find the maximum of that function(derivative of this function=0) ➔ the parameters which are most likely to have produced the observed data.

Maximum Likelihood Estimation is a powerful technique for fitting models to data

Assume when x=1 (heads), the probability is p, and when x=0 (tails), the probability is (1-p).
And the following sequence of flips: X = heads, heads, tails, heads, tails, tails, tails, heads, tails, tails.

Likelihood of a sequence of flips.

$$L_X(p) = P(X|p) = \prod_{x \in X} p^x (1-p)^{1-x}.$$

L(p) = p · p · (1-p) · p · (1-p) · (1-p) · (1-p) · p · (1-p) · ´´

Likelihood for n coin flips with h heads.

$$L(p) = p^h \cdot (1-p)^{n-h}.$$

We want to find the p that maximizes this function.

Apply log of both sides. This will bring the exponents down, and will turn the product into a sum. Taking the derivative of sums is easier than products (another convenience of log-likelihood). Remember, we can do this because the p that maximizes the log-likelihood is the same as the p that maximizes the likelihood. Our log-likelihood is:

$$l(p) = h \cdot \log(p) + (n-h) \cdot \log(1-p).$$

Log-likelihood of n coin flips with h heads.

To find the maximum we're going to take the derivative of this function with respect to p. If you're not comfortable with calculus, the important thing is that you know the derivative is the rate of change of the function. In this case, the derivative is:

$$l'(p) = \frac{h}{p} - \frac{n-h}{1-p}.$$

Derivative of the log-likelihood with respect to p.

$$l'(p) = \frac{h}{p} - \frac{n-h}{1-p}.$$

We set the derivative equal to 0 to find the maximum of the function (where the rate of change is 0). Setting the above equation equal to 0 and solving for p

$$p = \frac{h}{n}$$

The MLE estimate of p is the number of heads divided by the number of flips!

It turns out that the Maximum Likelihood Estimate for our coin is simply the number of heads divided by the number of flips!

This makes perfect intuitive sense, if you flipped a fair coin (p = 0.5) 100 times, you'd expect to get about 50 heads and 50 tails.

Ridge and Lasso Regression models

Problem with Linear regression

Multicollinearity can increase the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model

Model:

Spending on food = B+ B1salary+B2payingtaxes +error

Salary and paying taxes are highly related(Multicollinearity)➔ high variance

Samples from Houston ➔ different HoustonBetas

Samples from New York ➔ different NYbetas

High variance between Houstonbetas and Nybetas !

➔which betas are the optimized ones?

The need of Ridge regression which is Best Linear ~~Unbiased~~ Estimators

The average of houstonbetas and Nybetas is a not minimal but low variance

Ridge put constraints on Betas so that they will not get too large

RIDGE REGRESSION

Ridge Regression is a technique used when the data suffers from multicollinearity (independent variables are highly correlated).

In multicollinearity, even though the least squares estimates (OLS) are unbiased, their variances are large which deviates the observed value far from the true value. By adding a degree of bias(PENALTY/SHRINKAGE) to the regression estimates, ridge regression reduces the standard errors.

Model the error

It's very common to model the error as being drawn from a Gaussian distribution with mean zero and variance σ².

$$\epsilon \sim N(0, \sigma^2)$$

$$y = \theta_1 x + \theta_0 + \epsilon$$

A mean of zero distributes the error equally on both sides of the line. **The larger the variance, the larger the deviations.** ➔ add a Gaussian noise term to the model. The resulting model has three parameters: the slope, the intercept, and the variance of the Gaussian

Shrinkage

➔fitting a model involving all *p* predictors, however, the estimated coefficients are shrunken towards zero relative to the least squares estimates.

This shrinkage, aka *regularization* has the effect of reducing variance.
Depending on what type of shrinkage is performed, some of the coefficients may be estimated to be exactly zero.

Shrinkage also performs variable selection. The two best-known techniques for shrinking the coefficient estimates towards zero are the *ridge regression* and the *lasso*.

RIDGE REGRESSION

y=a+b*x+e (error term), (LINEAR REGRESSION +ERROR TO REGULATE THE VARIANCE)

[error term is the value needed to correct for a prediction error between the observed and predicted value]

=> y=a+y= a+ b1x1+ b2x2+....+e, for multiple independent variables.

- The assumptions of this regression is same as least squared regression except normality is not to be assumed
- It shrinks the value of coefficients but doesn't reaches zero, which suggests no feature selection feature
- This is a regularization method and uses l2 regularization.

The parameter λ is a tuning parameter. It modulates the importance of fit vs. shrinkage.

Use cross-validation with many values of λ to choose the best value of Lambda

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

RIDGE REGRESSION vs linear regression

Ridge regression is similar to least squares except that the coefficients are estimated by minimizing a slightly different quantity. Ridge regression, like OLS, seeks coefficient estimates that reduce RSS, however they also have a shrinkage penalty when the coefficients come closer to zero. This penalty has the effect of shrinking the coefficient estimates towards zero.

Ridge regression shrinks the features with the smallest column space variance. Like in principal component analysis, ridge regression projects the data into *d* directional space and then shrinks the coefficients of the low-variance components more than the high variance components, which are equivalent to the largest and smallest principal components.

Ridge and Lasso provide better prediction accuracy and model interpretability.

Prediction Accuracy

The curse of dimensionality:

Given that the true relationship between Y and X is approx. linear, the ordinary least squares estimates will have low bias. OLS also behaves well went n >> p.

Yet if n is not much larger than p, then there can be a lot of variability in the fit, resulting in overfitting and/or poor predictions.

If p > n, then there is no longer a unique least squares estimate, and the method cannot be used at all.

P large ➔

- Observations x start to become closer to the boundaries between classes than the nearby observations ➔major problems for predicting.
- The training samples are often sparsely populated, making it difficult to identify trends and predict

Model Interpretability.

•***Model Interpretability***: **Often in multiple regression, many variables are not associated with the response. Irrelevant variables leads to unnecessary complexity in the resulting model. By removing them (setting coefficient = 0) we obtain a more easily interpretable model. However, using OLS makes it very unlikely that the coefficients will be exactly zero.**

By constraining and shrinking the estimated coefficients, we can often substantially reduce the variance as the cost of a negligible increase in bias, which often leads to dramatic improvements in accuracy.

Approach for automatically excluding features

•***Subset Selection***: This approach identifies a subset of the *p* predictors that we believe to be related to the response. We then fit a model using the least squares of the subset features.

- *Shrinkage*. This approach fits a model involving all *p* predictors, however, the estimated coefficients are shrunken towards zero relative to the least squares estimates.
- *Dimension Reduction*: This approach involves projecting the *p* predictors into an *M*-dimensional subspace, where $M < p$. This is attained by computing *M* different *linear combinations*, or *projections*, of the variables. Then these *M* projections are used as predictors to fit a linear regression model by least squares(PCA).

Shrinkage Methods

fit a model containing **all** *p* predictors using a technique that *constrains* or *regularizes* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero.

The two best-known techniques for shrinking the coefficient estimates towards zero are the *ridge regression* and the *lasso*.

Note that the shrinkage does not apply to the intercept.

Why Penalize the Magnitude of Coefficients?

What is the impact of model complexity on the magnitude of coefficients?

The **size of coefficients increase exponentially with increase in model complexity**

What does a large coefficient signify? It means putting a lot of emphasis on that feature, i.e. the particular feature is a good predictor for the outcome. When it becomes too large, the algorithm starts modelling intricate relations to estimate the output and ends up overfitting to the particular training data.

Shrinkage Methods..Penalty ?

Suppose you have two ways of fitting your data, such as

$y=2x_1+0x_2$

or

$y=x_1+x_2$

Sum of betas squared $=2^2+0^2=4$

Sum of betas squared $=1^2+1^2=2$

In general the effect of each weight on the prediction will be linear but its penalty will be quadratic. Thus it will pay off to put lots of small values instead of just a few big ones.

***Penalizing the flexibility of a model** is a technique that discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.*

Recall Overfitting : regularization

- A regularizer is an additional criterion to the loss function to avoid overfitting
- It's called a regularizer since it tries to keep the parameters more normal/regular
- It is a bias on the model which forces the learning model to prefer certain types of weights over others

$$argmin_{(\beta 0\,,\,\beta i\,)} \Sigma \text{loss} + \lambda \text{regularizer}\,(\beta_0\,,\,\beta_i\,)$$

The values of Betas for which the loss function is minimized

Regularizers

$$\beta_0 + \Sigma \beta_i X_i$$

- Generally, huge weights are not encouraged: If weights are large, a small change in a feature can result in a large change in the prediction
- Better strategy is to assign weights of 0 for features that aren't useful

How do we encourage small weights? or penalize large weights?

Regularizers

$$\beta_0 + \Sigma\beta_i X_i$$

How do we encourage small weights? or penalize large weights?

$$argmin_{(\beta 0\,,\,\beta i\,)} \Sigma \text{loss} + \lambda \text{regularizer}\ (\beta_0\,,\,\beta_i\,)$$

➔Learning Algorithm

Common regularizers

sum of the weights $r(\beta_i, \beta_0) = \Sigma|\beta i|$

sum of the squared weights $r(\beta_i, \beta_0) = \sqrt{\Sigma|\beta i|2}$

What's the difference between these?

Common Regularizers

sum of the weights $r(\beta_i, \beta_0) = \Sigma|\beta i|$

sum of the squared weights $r(\beta_i, \beta_0) = \sqrt{\Sigma|\beta i|2}$

Sum of weights will penalize small values more

Squared weights penalizes large values more

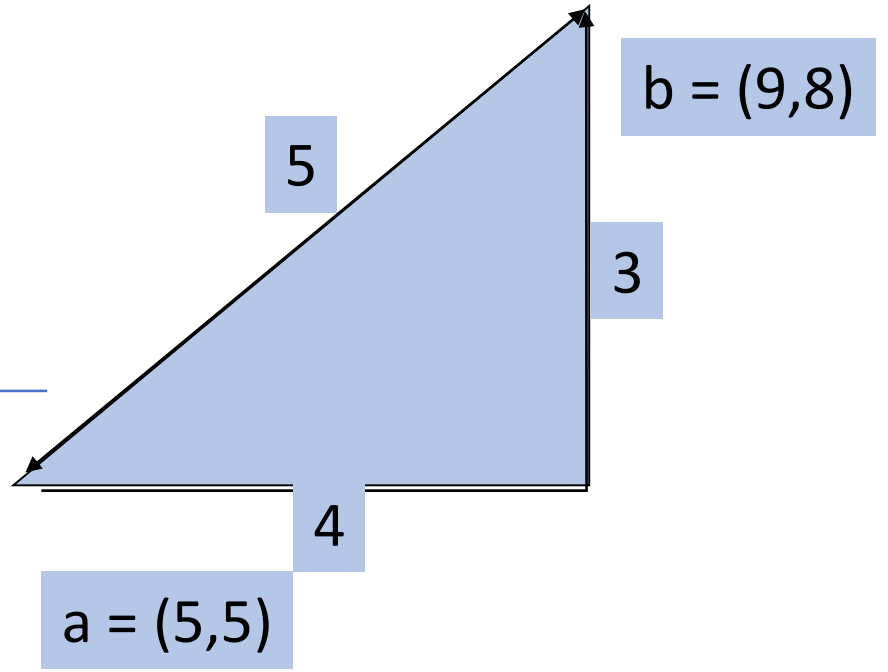
P-norm

L_1-norm: dist(x,y) = 4+3 = 7

L_2-norm: dist(x,y) = $\sqrt{(4^2+3^2)}$ = 5

sum of the weights (1-norm) $r(\beta_i, \beta_0) = \Sigma|\beta i|$

sum of the squared weights (2-norm) $r(\beta_i, \beta_0) = \sqrt{\Sigma|\beta i|2}$



p-norm $r(\beta_i, \beta_0) = \sqrt{\Sigma|\beta i|^p} = \|\beta\|^p$

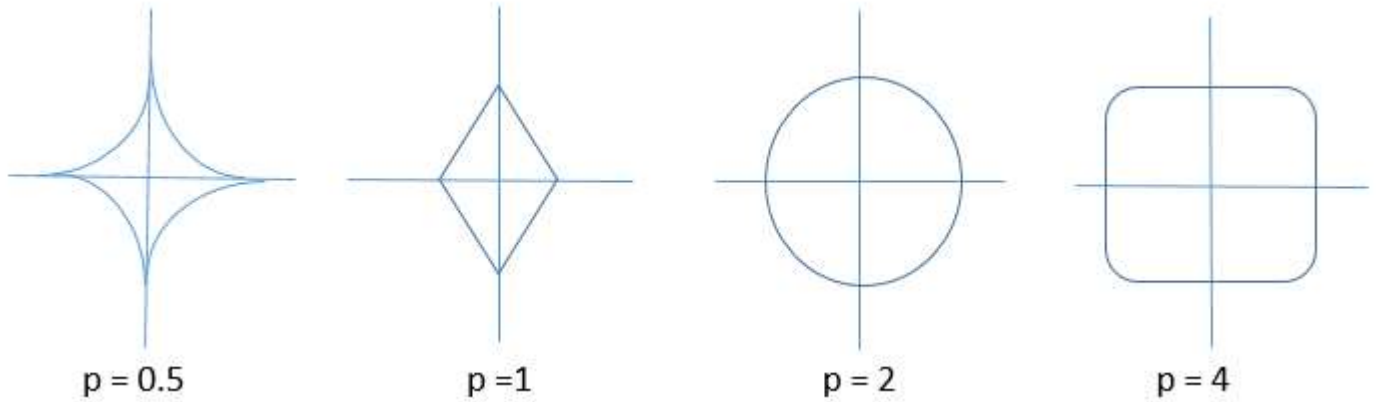
Smaller values of p (p < 2) encourage sparser vectors

Larger values of p discourage large weights more

Recall Euclidean Distances

Lp regularizer.

$$(\Sigma|\beta i|^p)^{1/p}$$



For p=0.5, we can only get large values of one parameter only if other parameter is too small.

For p=1, we get sum of absolute values where the increase in one parameter Θ is exactly offset by the decrease in other.

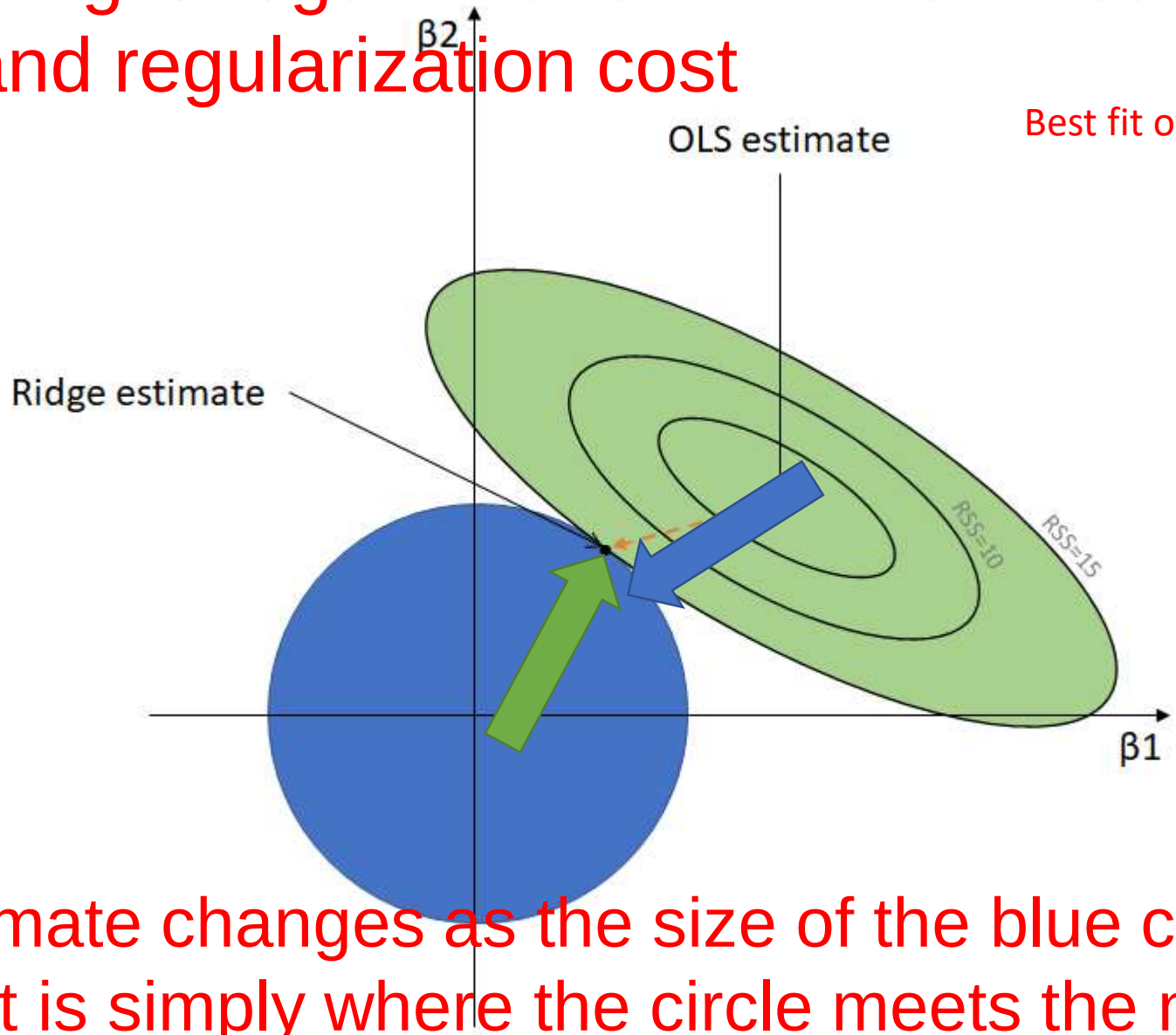
For p =2, we get a circle

and for larger p values, it approaches a round square shape.

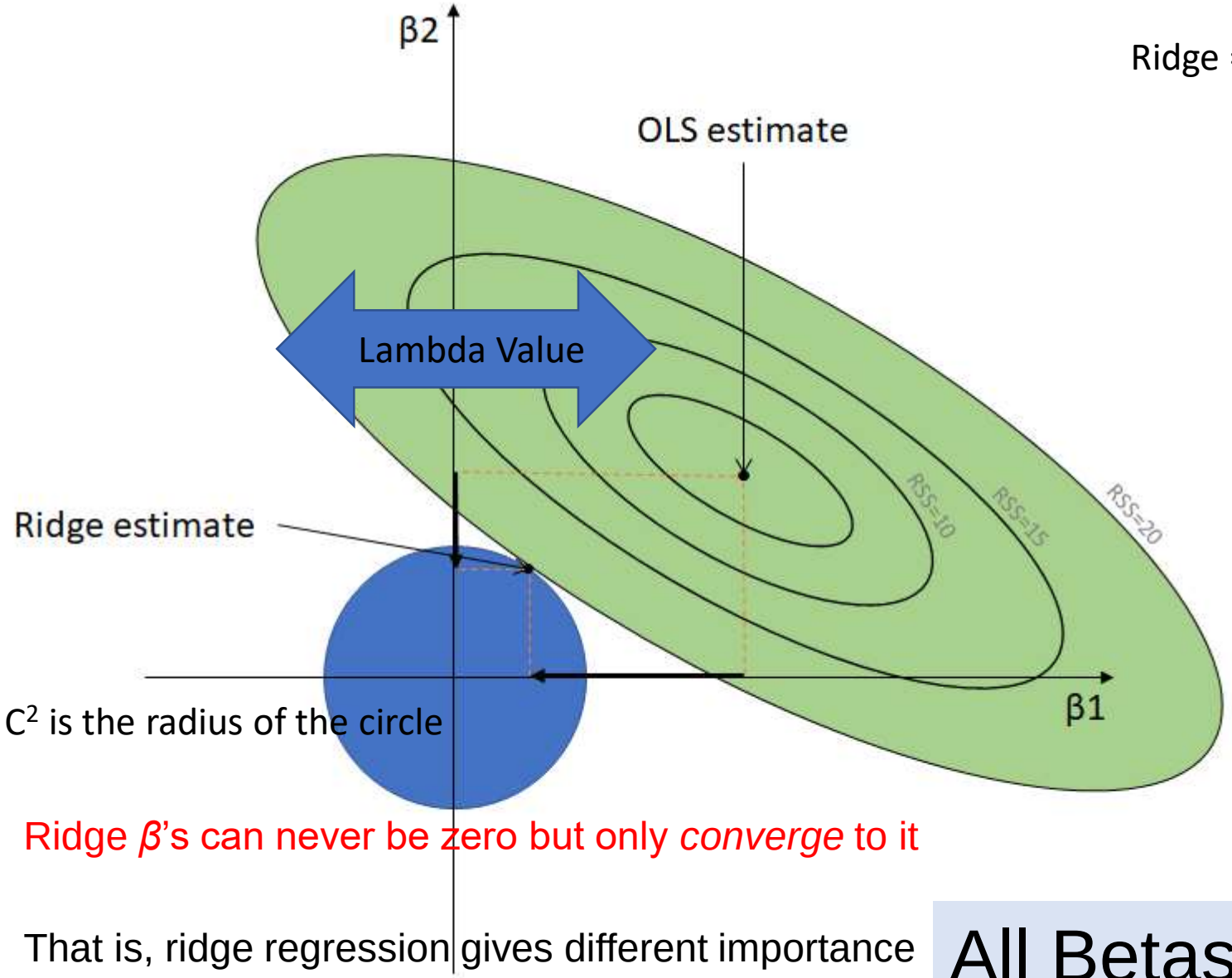
The two most commonly used regularization are in which we have p=1 and p=2, more commonly known as L1 and L2 regularization.

Optimizing is regularization between mean square error and regularization cost

Best fit of the training set ➔Lowest RSS



Ridge estimate changes as the size of the blue circle changes. It is simply where the circle meets the most outer contour



Ridge = Ols + $\beta_0^2 + \beta_1^2 + \cdots + \beta_p^2 \leq C^2$

Ridge =Ols + Lambda *(norm)

Norm is the penalty
Lambda is how severe this penalty is

C^2 is the radius of the circle

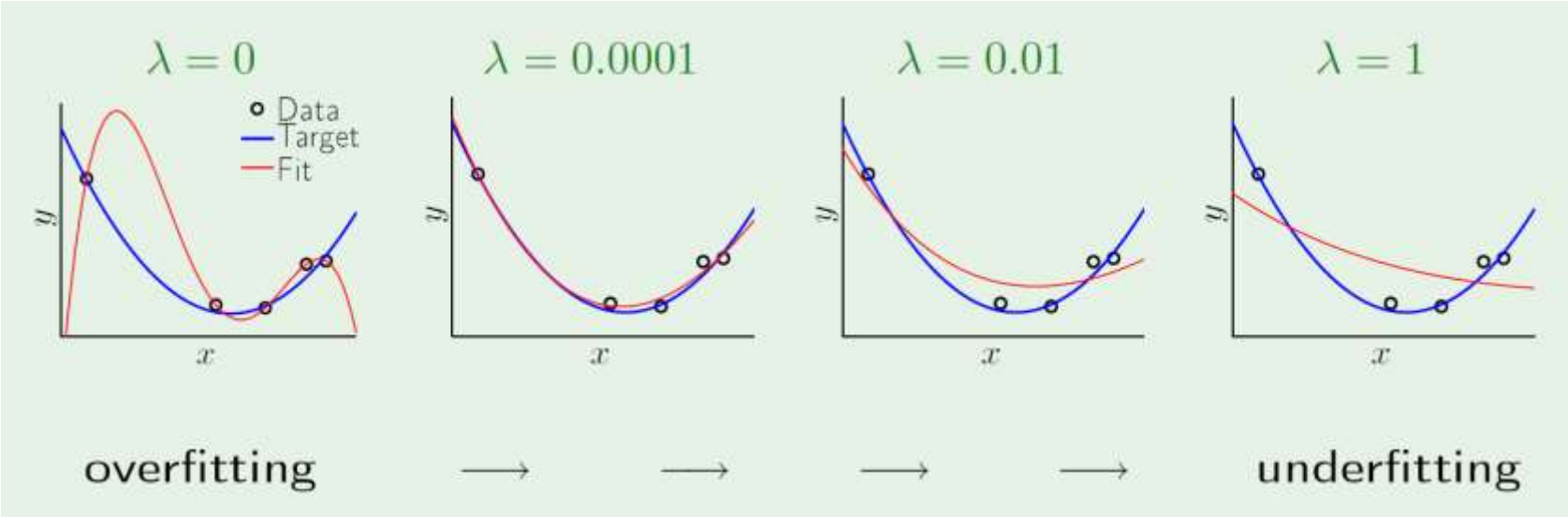
Ridge β1 relatively drops more quickly to zero than ridge β2 does as the circle size changes (compare the two figures). The reason why this happens is because the β's change differently by the RSS.

Ridge β's can never be zero but only *converge* to it

That is, ridge regression gives different importance weights to the features but does not drop unimporta features.

All Betas are limited inside the circle and on the circumference

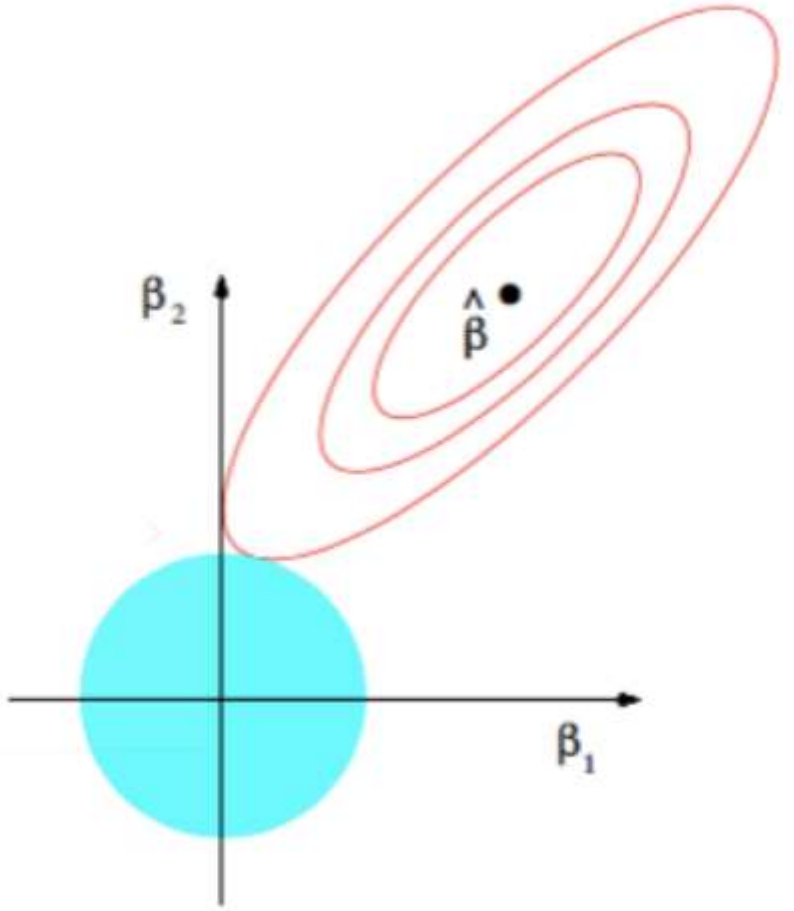
Lambda's influence on a model



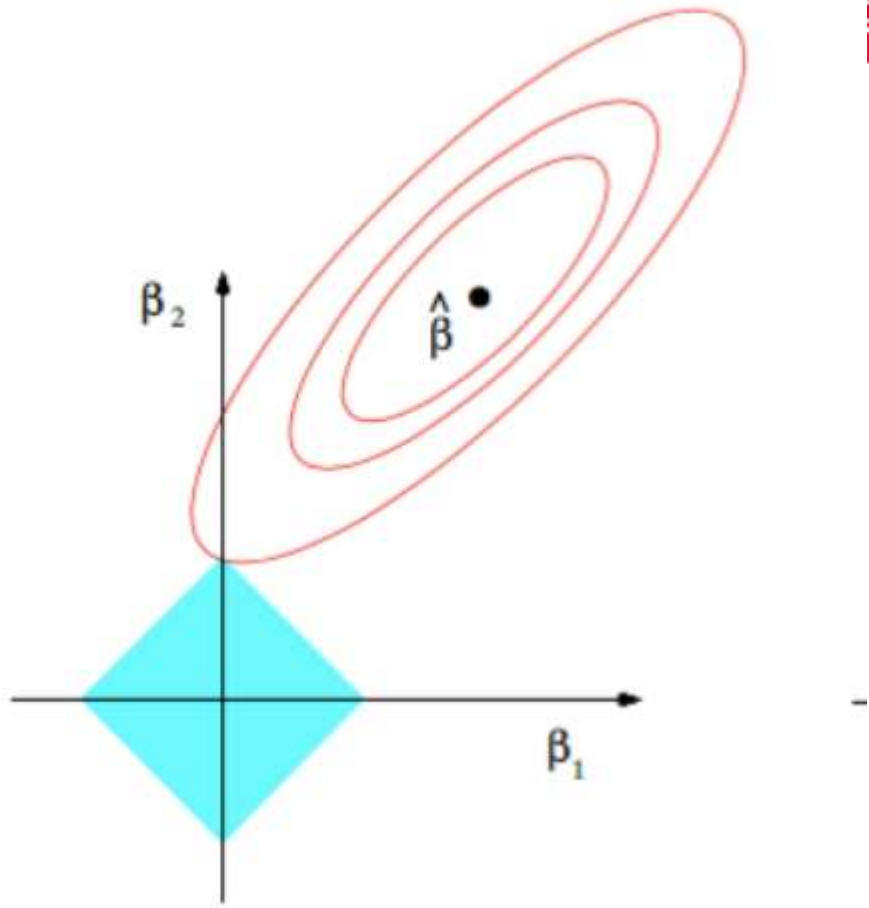
- *Cross-validation* provides a simple way to select the best lambda. We choose a grid of λ values, and compute the cross-validation error rate for each value of λ.
- We then select the tuning parameter value for which the cross-validation error is smallest.
- Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

Since the shape formed by L2 regularizer is a circle, it increases quadratically as we move away from it.

The L2 optimum(which is basically the intersection point) can fall on the axis lines only when the minimum MSE (mean square error or the black point in the figure) is also exactly on the axis.



The black point denotes that the least square error is minimized at that point and as we can see that it increases quadratically as we move from it and the regularization term is minimized at the origin where all the parameters are zero .



At what point will our cost function be minimum? The answer will be, since they are quadratically increasing, the sum of both the terms will be minimized at the point where they first intersect.

L1 or L2?

The L1 optimum can be on the axis line because its contour is sharp and therefore there are high chances of interaction point to fall on axis. Therefore it is possible to intersect on the axis line, even when minimum MSE is not on the axis. If the intersection point falls on the axes it is known as sparse.

Therefore L1 offers some level of sparsity which makes our model more efficient to store and compute and it can also help in checking importance of feature, since the features that are not important can be exactly set to zero.

Advantage of RIDGE REGRESSION

Why is ridge regression better than least squares?

The advantage is apparent in the bias-variance trade-off. As λ increases, the flexibility of the ridge regression fit decreases. This leads to decrease variance, with a smaller increase in bias. Regular OLS regression is fixed with high variance, but no bias. However, the lowest test MSE tends to occur at the intercept between variance and bias.

- In Ridge ➔ by properly tuning λ and acquiring less variance at the cost of a small amount of bias➔find a lower potential MSE.
- Ridge regression works best in situations for least squares estimates have high variance.
- Ridge regression is much more computationally efficient that any subset method, since it is possible to simultaneously solve for all values of λ.

Disadvantage of RIDGE REGRESSION

it includes all *p* predictors in the final model. The penalty term will set many of them close to zero, but never *exactly* to zero.

This isn't generally a problem for prediction accuracy, but it can make the model more difficult to interpret the results.

Lasso overcomes this disadvantage and is capable of forcing some of the coefficients to zero granted that *s* is small enough. Since *s* = 1 results in regular OLS regression, as *s* approaches 0 the coefficients shrink towards zero. Thus, Lasso regression also performs variable selection.

LASSO REGRESSION

y=a+b*x+e (error term), (LINEAR REGRESSION +ERROR TO REGULATE THE VARIANCE)

[error term is the value needed to correct for a prediction error between the observed and predicted value]

The assumptions of this regression is same as least squared regression except normality is not to be assumed

It shrinks coefficients to zero (exactly zero), which certainly helps in feature selection

This is a regularization method and uses l1 regularization

If group of predictors are highly correlated, lasso picks only one of them and shrinks the others to zero

`

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

When we use regularization?

Find out if there is **Multicollinearity**

1-Begin by studying pairwise scatter plots of pairs of independent variables, looking for near-perfect relationships. Study the correlation matrix for high correlations.

2. Consider the variance inflation factors (VIF). VIFs over 10 indicate collinear variables.

3. Eigenvalues of the correlation matrix of the independent variables near zero indicate multicollinearity. Instead of looking at the numerical size of the eigenvalue, use the condition number. Large condition numbers indicate multicollinearity.

4. Investigate the signs of the regression coefficients. Variables whose regression coefficients are opposite in sign from what you would expect may indicate multicollinearity.