STEVENS INSTITUTE of TECHNOLOGY
School of Business

Multivariate Data Analytics

Model Comparison and Selection

Prof. Feng Mai
School of Business

Key concepts

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Objective of least squared method
- Also called **sum of squared estimate of errors** (**SSE**)

$$MSE = \frac{RSS}{n - p}$$

- **Mean squared error**
- n is the sample size, p is the number of parameters (number of x variables + 1)

$$SSTO = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

- **Total sum of squares**
- Total amount of variance in y

$$R^2 = 1 - \frac{SSE}{SSTO}$$

- **Coefficient of Determination, R-squared**
- The proportion of variance explained by the model, from 0 to 1.
- Always increases when more variables are added in the model.

Other model comparison criteria

- Adjusted R2

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-P-1}$$

- Higher the better

- AIC (Akaike information criterion)

$$\mathrm{AIC} = 2P + n\log(\mathrm{RSS}/n)$$

- Lower the better

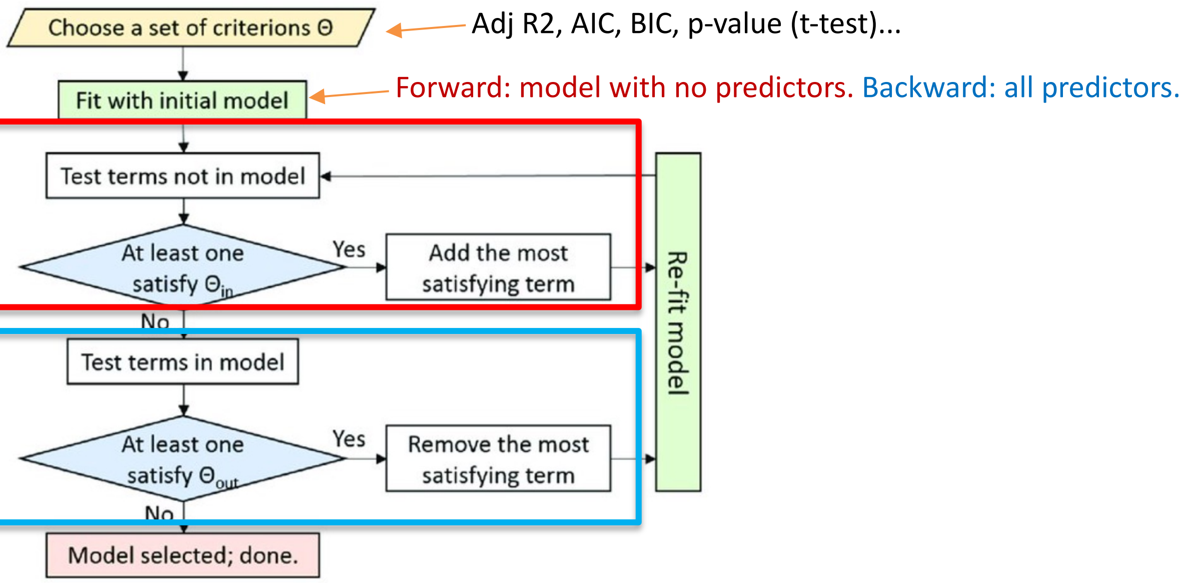
- BIC (Bayesian information criteria)

$$\mathrm{BIC} = P\log(\mathrm{n}) + \mathrm{n}\log(\mathrm{RSS}/\mathrm{n})$$

- Lower the better

Model selection

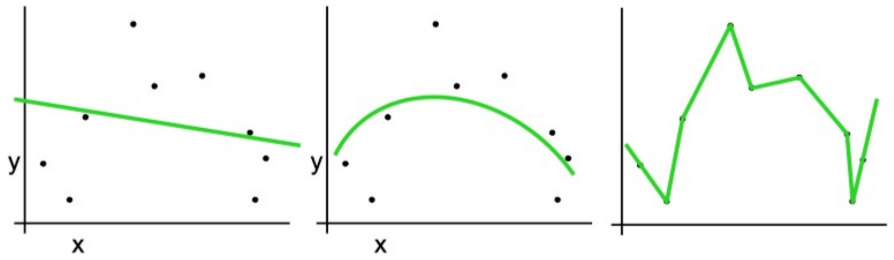
- **Best subset regression**: fit all the possible models from all of the possible combinations of the predictors, then compare adj R2 / AIC /BIC
    - 2^k possible models for k predictors
    - Over 1 million models with 20 predictors
- **Stepwise regression:** forward selection, backward elimination, stepwise


Choose a set of criterions Θ
Adj R2, AIC, BIC, p-value (t-test)...
Fit with initial model
Forward: model with no predictors. Backward: all predictors.
Test terms not in model
At least one satisfy Θ_{in}
Yes
Add the most satisfying term
No
Test terms in model
At least one satisfy Θ_{out}
Yes
Remove the most satisfying term
No
Model selected; done.
Re-fit model

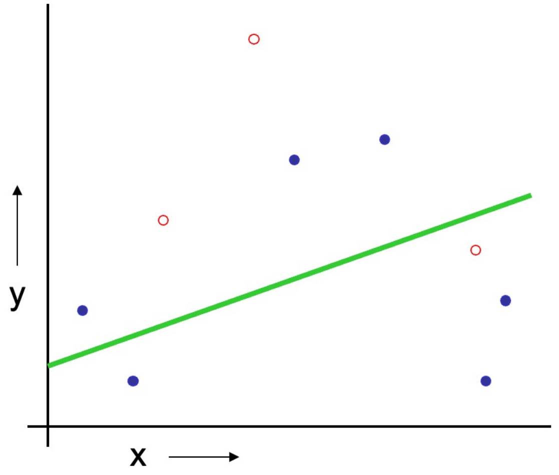
Source: Yang et al 2017. 10.3390/nano7100319

Model Comparison Revisited

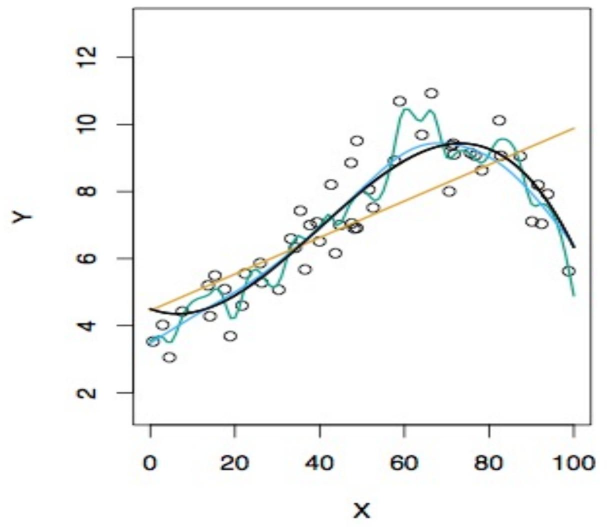
- A good model should be generalizable (useful for prediction)
- **Adj R^2, AIC, BIC**: penalizes complicated models after fitting
- **Test Set Approach:** Find the best model by explicitly testing models' ability to generalize

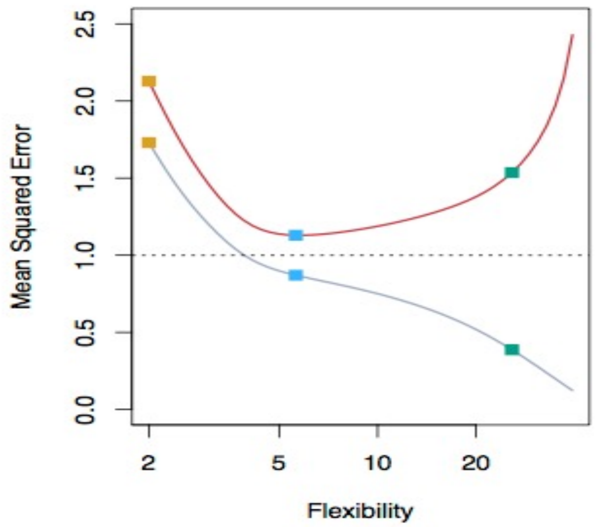


1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Fit regression models on the training set
4. For each model, estimate future performance (MSE) with the test set
5. Use the model with the best test set performance



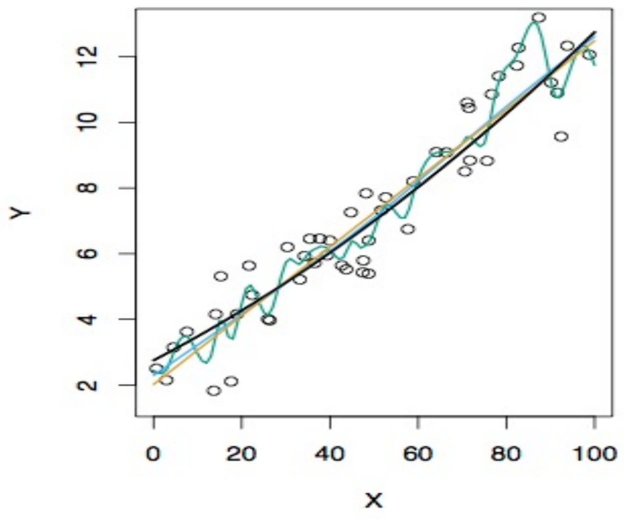
Source: Adapted from Andrew W. Moore's Tutorial

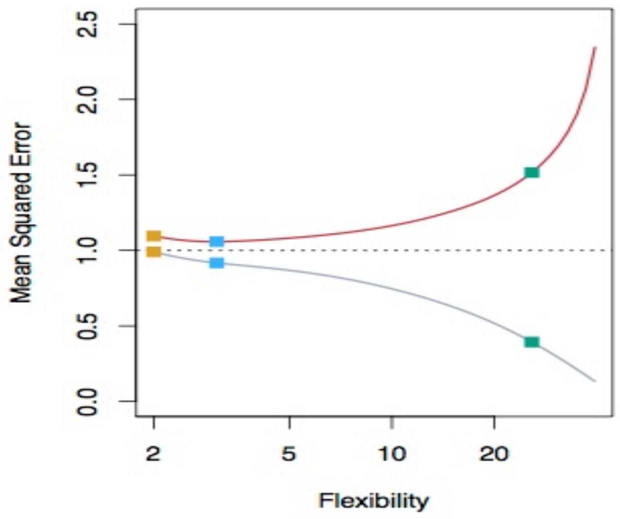




LEFT
Black: Truth
Orange: Linear Estimate
Blue: smoothing spline
Green: smoothing spline (more flexible)

RIGHT
RED: Test MES
Grey: Training MSE
Dashed: Minimum possible test MSE (irreducible error)





<u>LEFT</u>

Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)

<u>RIGHT</u>

RED: Test MES

Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

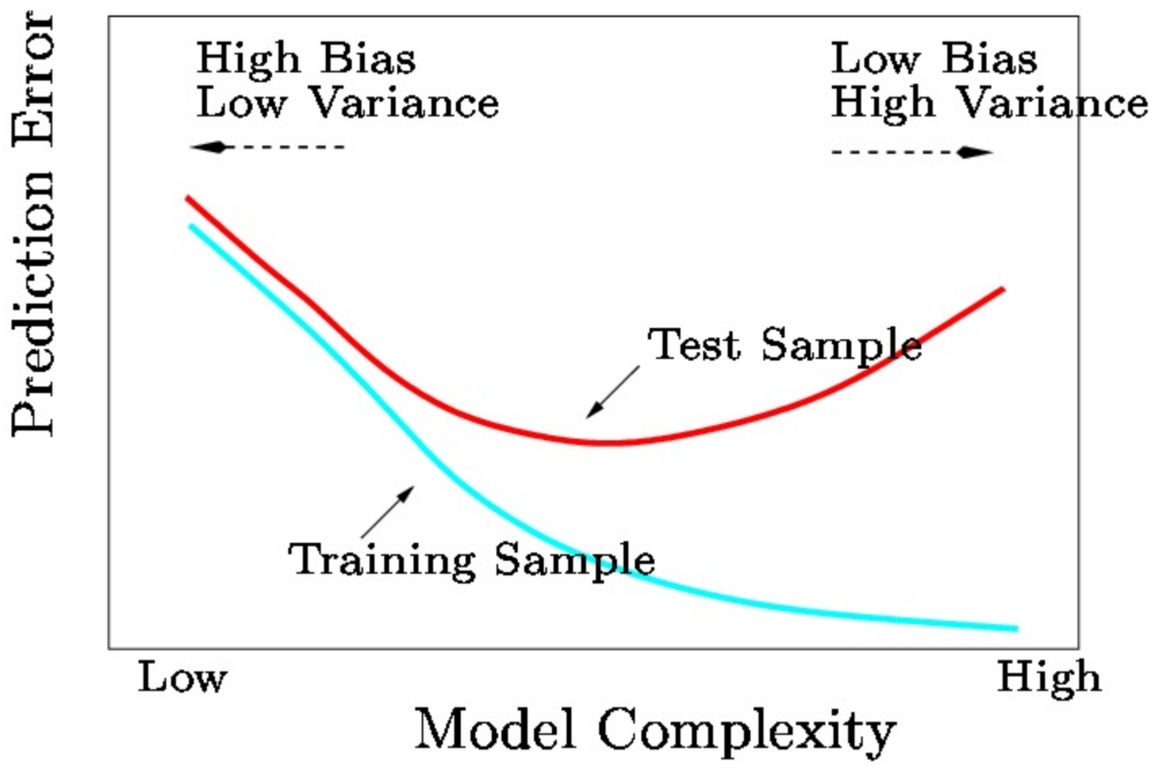
Bias/ Variance Tradeoff

- The previous graphs of test versus training MSE's illustrates a very important tradeoff that governs the choice of statistical learning methods.
- There are always two competing forces that govern the choice of learning method i.e. bias and variance.
- Bias refers to the error that is introduced by modeling a real life problem (that is usually extremely complicated) by a much simpler problem.
    - For example, linear regression assumes that there is a linear relationship between Y and X. It is unlikely that, in real life, the relationship is exactly linear so some bias will be present.
    - The more flexible/complex a method is the less bias it will generally have.
- Variance refers to how much your estimate for f would change by if you had a different training data set.
    - Generally, the more flexible a method is the more variance it has.
- The Trade-Off

$$Expected\,Test\,MSE = E\left(Y - f(x_0)\right)^2 = Bias^2 + Var + \underbrace{\sigma^2}_{\text{Irreducible Error}}$$

- As a method gets more complex the bias will decrease and the variance will increase but expected test MSE may go up or down!

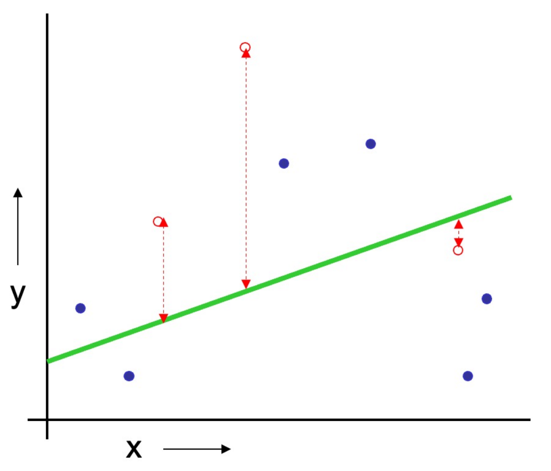
- In general training errors will always decline.
- However, test errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate).

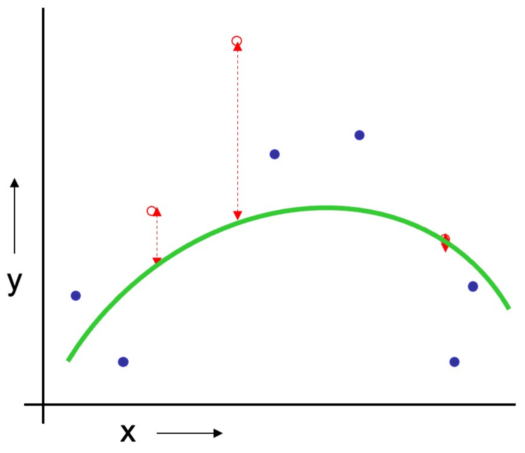


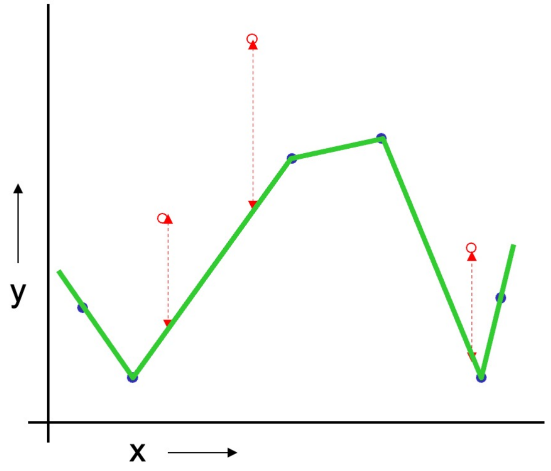
We must always keep this picture in mind when choosing a learning method. More flexible/complicated is not always better!

Test Set Performance

All models are fitted using train set (blue points) but evaluated on the test set (red points)







(Linear regression example)

Mean Squared Error = 2.4

(Quadratic regression example)

Mean Squared Error = 0.9

(Join the dots example)

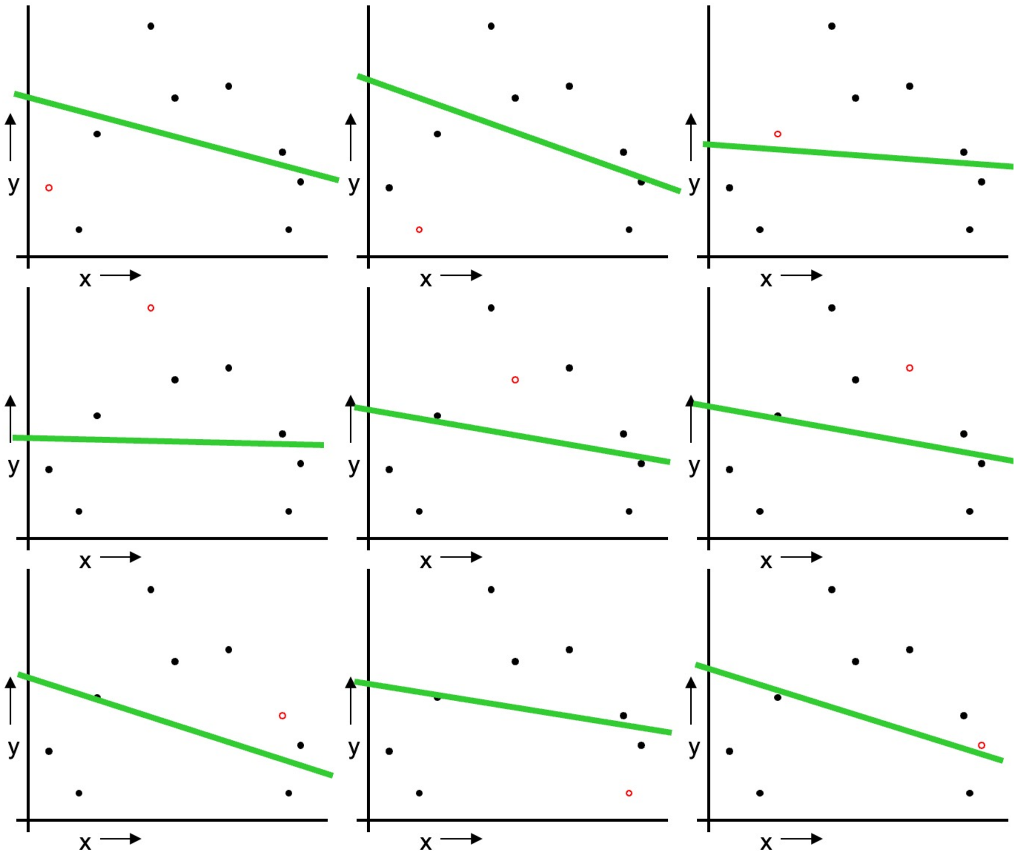
Mean Squared Error = 2.2

Best Model!

But...

- We wasted data.
- Performance estimation on small test set has high variance

LOOCV (Leave-one-out Cross Validation)



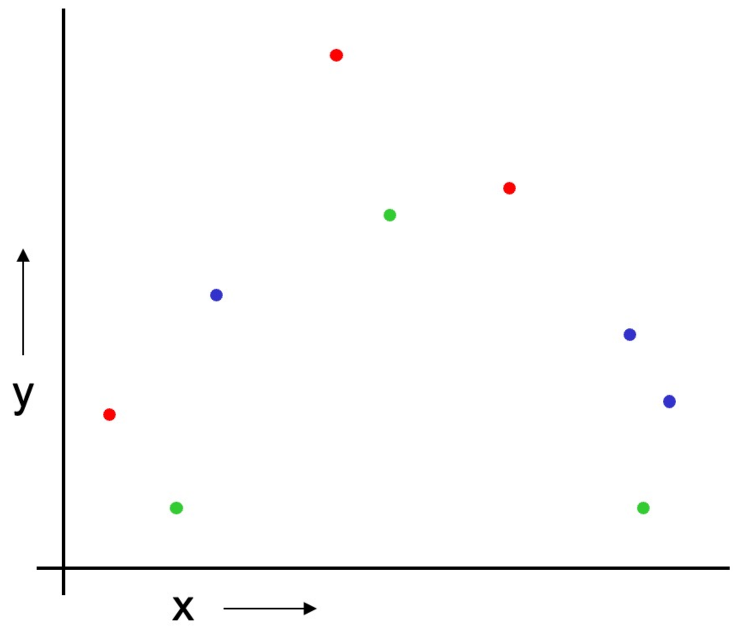
For k=1 to n

1. Let *(x_k,y_k)* be the *k*th record
2. Temporarily remove *(x_k,y_k)* from the dataset (use this single point as the test set)
3. Train on the remaining n-1 datapoints
4. Note your error *(x_k,y_k)*

When you've done all points, report the average error for all points.

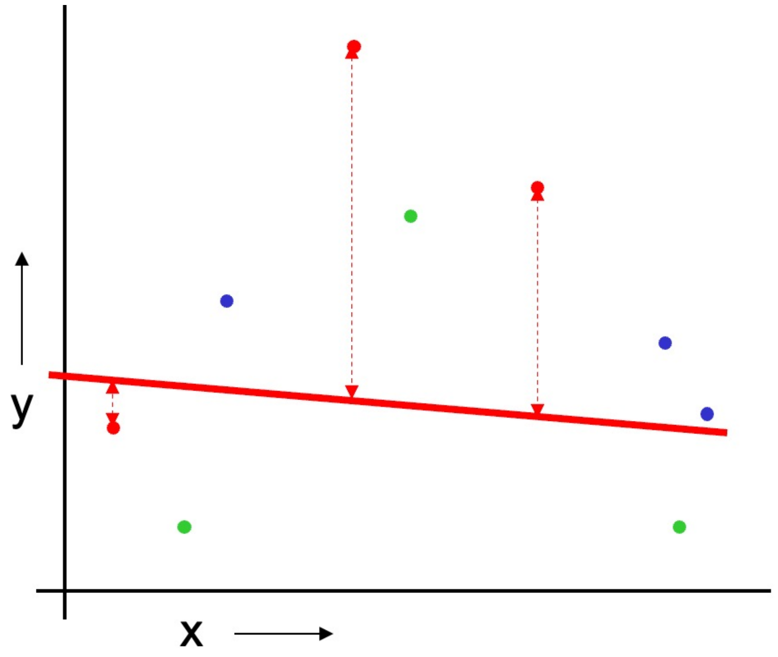
Pros: Wasted less data and more stable
Cons: Time consuming

k-Fold Cross Validation



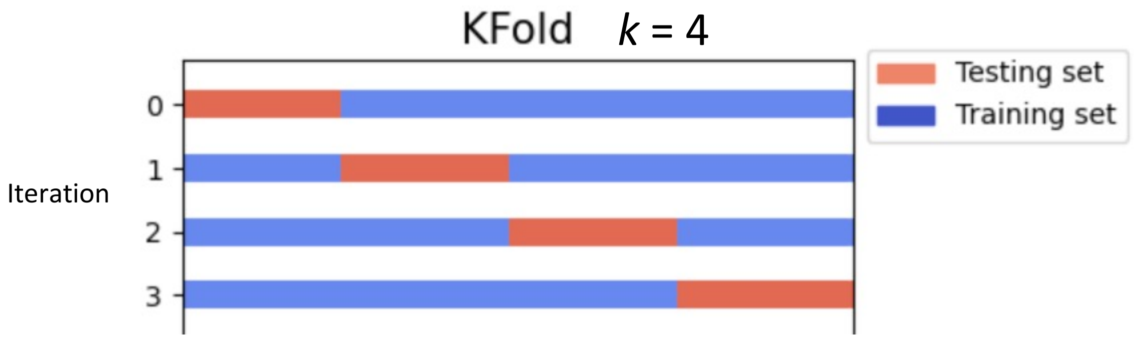
- Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red, Green and Blue)
  - For the red partition: Train on all the points not in the red partition (2/3 of the data). Find the test-set sum of errors on the red points (1/3 of the data).
  - For the green partition: Train on all the points not in the green partition (2/3 of the data). Find the test-set sum of errors on the green points (1/3 of the data).
  - For the blue partition: Train on all the points not in the blue partition (2/3 of the data). Find the test-set sum of errors on the blue points (1/3 of the data).
  - Then report the average error

k-Fold Cross Validation

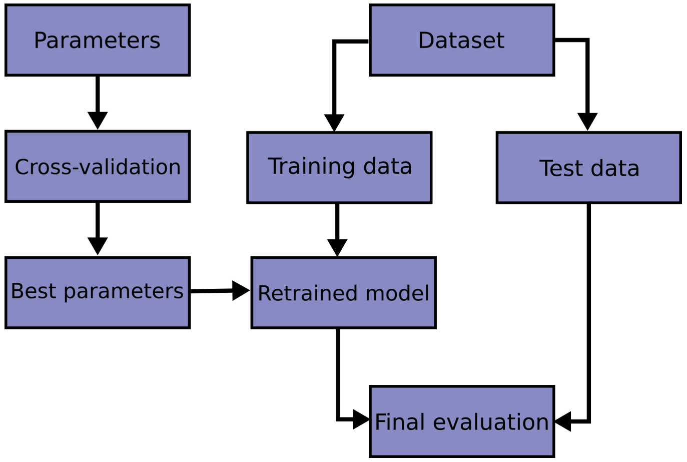


- Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red, Green and Blue)
  - For the red partition: Train on all the points not in the red partition (2/3 of the data). Find the test-set sum of errors on the red points (1/3 of the data).
  - For the green partition: Train on all the points not in the green partition (2/3 of the data). Find the test-set sum of errors on the green points (1/3 of the data).
  - For the blue partition: Train on all the points not in the blue partition (2/3 of the data). Find the test-set sum of errors on the blue points (1/3 of the data).
  - Then report the average error

k-Fold Cross Validation



- Widely used to compare models and tune model parameters
- Less computationally expensive than LOOCV
  - Still more computationally expensive than AIC/BIC
- More stable than test set approach (average performance of the k partitions)
- Typically choose *k* = 10 or 5



Source: https://scikit-learn.org/stable/modules/cross_validation.html

Review

- In a linear regression (OLS), the goal is to minimize the residual sum of squares:

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

- If a dataset has many predictors, we can fit a full model with all predictors.
  - Not all variables are useful
  - More complicated models are less generalizable (overfit)
- We can use best subset regression or stepwise regression to find a simpler and more generalizable model with a subset of predictors.
  - best subset regression: slow
  - stepwise regression: inconsistent results with different model selection algorithms

Ridge Regression

- Modifying the objective

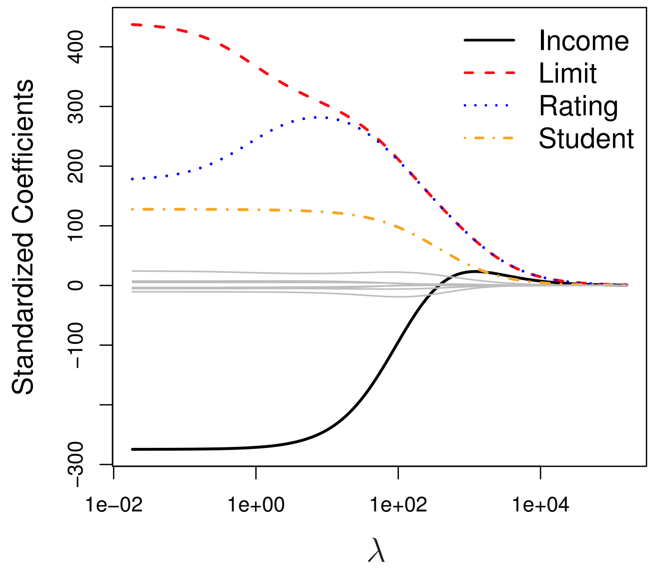
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\overset{\text{OLS objective}}{\text{RSS}}+\overset{\text{Tuning parameter}}{\lambda}\underbrace{\sum_{j=1}^{p}\beta_j^2}_{\text{Model complexity penalty}},$$

Tuning parameter λ is positive. When λ = 0, we get the OLS objective

- Since we are minimizing the objective, model with high βs = bad!
- This has the effect of "shrinking" large values of βs towards zero.
- Improve the fit, because shrinking the coefficients can significantly reduce their variance
- Also handles multicollinearity
- Also called L2 regularization

Tuning Parameter λ for Ridge

- When λ = 0, we get the OLS objective
- As λ increases, the standardized coefficients shrinks towards zero.
- Small λ → More complicated models (high variance, low bias).
- Large λ → Simpler models (low variance, high bias).
- Choose the λ to minimize BIC/AIC, or use cross-validation to choose λ



Source: James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

LASSO (least absolute shrinkage and selection operator)

- Modifying the objective

OLS objective

Tuning parameter

Absolute value instead of squared!

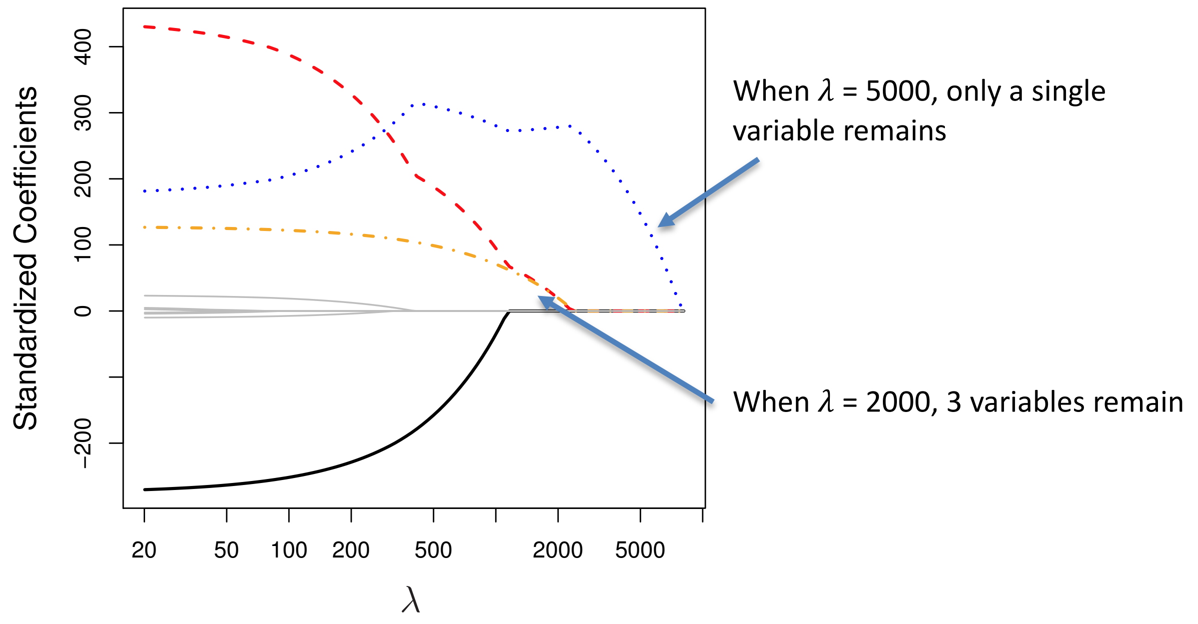
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS}+\lambda\underbrace{\sum_{j=1}^{p}|\beta_j|}_{\text{Model complexity penalty}}.$$

Similar idea to Ridge Regression, with a big difference in results

- Using this penalty, some coefficients end up being set to exactly zero
- LASSO does variable selection! Variables with β coefficients = 0 are eliminated.
- Also called L1 regularization

Tuning Parameter λ for LASSO

- When λ = 0, we get the OLS objective
- As λ increases, some coefficients become zero.
- Small λ → Fewer variables eliminated, more complicated models (high variance, low bias).
- Large λ → More variables eliminated, simpler models (low variance, high bias).
- Choose the λ to minimize BIC/AIC, or use cross-validation to choose λ



Source: James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

One can show that the lasso and ridge regression coefficient estimates solve the problems

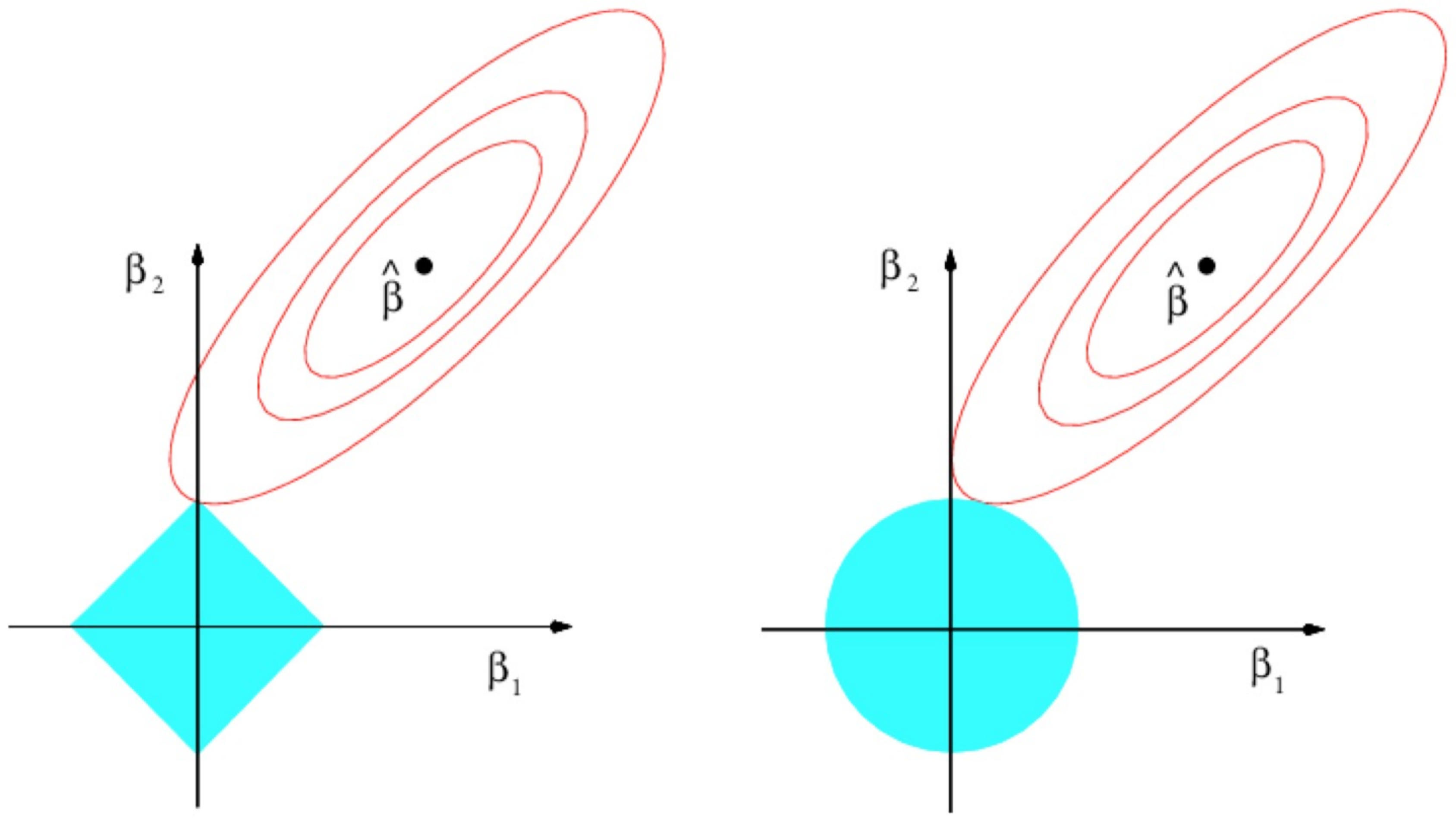
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \le s$$

and

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

respectively.

The Lasso Picture



STEVENS
INSTITUTE of TECHNOLOGY
School of Business

Thank you!

Prof. Feng Mai
School of Business