# 40.016: Analytics Edge
# Week 5 Lecture 1

## MODEL ASSESSMENT AND MODEL SELECTION: SUBSET SELECTION

Term 6, 2020

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

# Outline

- Model assessment and Model selection
- Bias-Variance trade-off
- Subset Selection

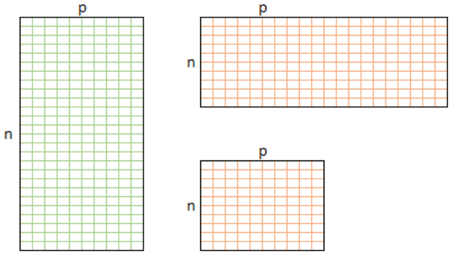
- Cross validation
- LASSO

# Model assessment and Model selection

- We will use recent ideas in regression and classification.
- Mostly developed for large data sets with many predictors.
- GOAL – prediction accuracy
  – model interpretability

# Big data

- $n$ : # observations.
- $p$ : predictors, attributes.
- Classical statistics: $n >> p$.
- But sometimes: $n \sim p$ or $n << p$.
- cancer dataset
  — many many genes (potential predictors)
  — a very small sample
- High flexibility in model selection.



# Bias-Variance trade-off

- Think of a linear regression model. The true model is:

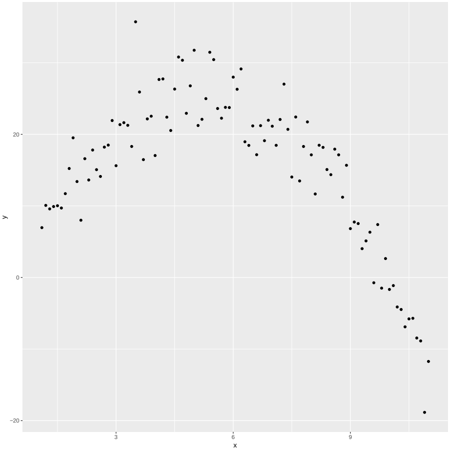
$$Y = f(X) + \epsilon$$

where $\epsilon$ is a random error term with mean 0 and variance $\sigma^2$.

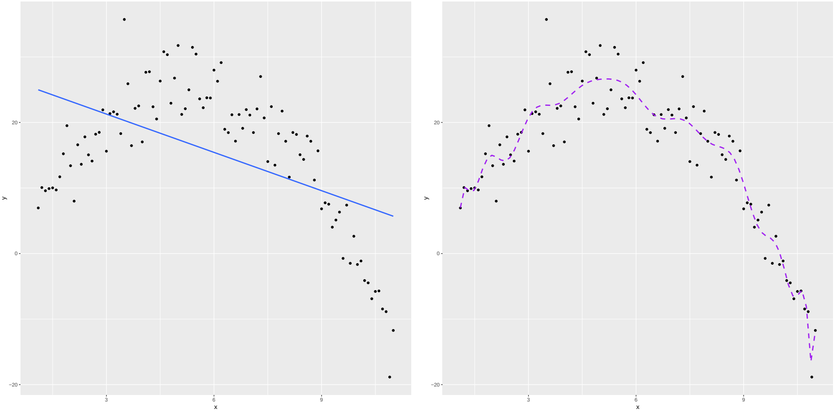
- Suppose we use least squares minimization to find predictor $\hat{f}$ for $f$.
- We use training set data to find $\hat{f}$ but expect good performance out-of-sample.
- Let $(X_0, Y_0)$ be a (test) data point.
- We want low test MSE or low test RSS (residual sum of squares):

$$\text{Test MSE} = \mathbb{E}(Y_0 - \hat{f}(X_0))^2.$$

# Balancing bias and variance

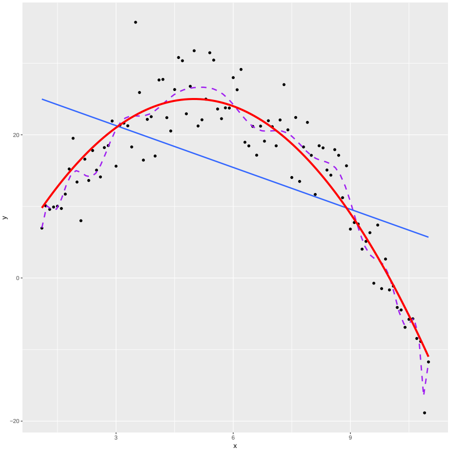


# Balancing bias and variance




Left: Linear regression fit, Right: Higher degree polynomial fit

# Balancing bias and variance




Blue line: linear regression, Purple line: higher degree polynomial, Red curve: actual function.

# Bias and Variance

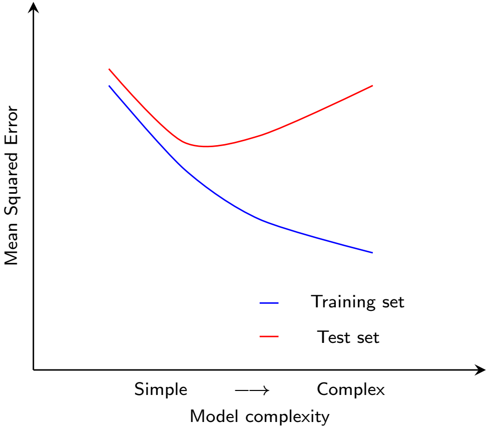
# Bias-Variance trade-off

- We can check that:

$$
\begin{aligned}
\mathbb{E}(Y_0 - \hat{f}(X_0))^2 &= \text{Var}(\hat{f}(X_0)) + \mathbb{E}\left[(f(X_0) - \hat{f}(X_0))^2\right] + \sigma^2 \\
&= \text{Variance of estimator} \\
&\quad + \text{Squared Bias} \\
&\quad\quad + \text{Variance of error term (irreducible errror).}
\end{aligned}
$$

- Complex model: typically high variance and low bias.
- Simple model: Low variance but high bias.
- Find the right balance.

# Training set error and test set error



# Subset selection

- We have $n$ observations, and $p$ predictor variables.
- If $n \approx p$ or $n < p$: risk of overfitting.
- Pick a selection of important explanatory variable from the $p$ available..
- Solution space: $2^p$ subsets.

# Best subset selection

- Let $M_0$ denote a null model with no predictors (only the intercept).
- For $k = 1, \ldots, p$, do:
  - Fit all $\binom{p}{k}$ models that contains exactly $k$ predictors and the intercept.
  - Pick the best among these models by choosing the model with the minimum sum of squared errors for linear regression or maximum log-likelihood for logistic regression. Call this model $M_k$.
- Choose the best model among $M_0$, $M_1$, ..., $M_p$ using cross-validation errors or adjusted R-squared (linear regression) or AIC (logistic regression).

Complexity is $O(2^p)$: we need to solve $2^p$ linear or logistic regressions.

# Forward stepwise selection

- Let $M_0$ denote a null model with no predictors (only the intercept).
- For $k = 1, \ldots, p$, do:
  - Fit all models that augment the predictors in model $M_{k-1}$ with exactly one more predictor (a total of $p - k + 1$ models are fit in step $k$).
  - Pick the best among these models by choosing the model with the minimum sum of squared errors for linear regression or maximum log-likelihood for logistic regression. Call this model $M_k$.
- Choose the best model among $M_0$, $M_1$, ..., $M_p$ using cross-validation errors or adjusted R-squared (linear regression) or AIC (logistic regression).

Complexity is $O(p^2)$: but there is no guarantee that this will always find the optimal solution.

# Backward stepwise selection

- We start with the subset including all variables.
- We drop variables one at a time rather than adding.
- Has the same complexity as forward stepwise selection.
- Solutions from backward & forward methods can be different.
- Solutions of both can differ from the best subset selection solution.

Complexity is $O(p^2)$: no guarantee of optimality.

# Cross-Validation

- Model assessment technique.
- A model is considered good if it has a low *test set error (TEST MSE)* .
- We often do not have a large test set to validate our model.
- One method of model assessment here is *Cross Validation*.

# Validation set approach

1. Divide the data randomly into $2$ subsets (often roughly of equal size): the *training set* and the *validation set* or *hold-out set*.
2. Use the training set to fit the model, and the validation set to predict and then estimate Mean squared error (MSE).

Potential drawbacks:

1. The method depends on the points chosen, hence different choices may lead to starkly different estimated MSEs.
2. Since we are only using a subset of the available data set, the performance of the model is worse than it would be on a larger data set. And the error estimates tend to be larger.

# LOOCV: Leave out one cross validation

Compensates for the drawbacks of the *Validation set approach* yet keeping the same spirit.

1. For every $i \in I = \{1, \ldots, n\}$, train the model on the set $I \setminus \{i\}$.
2. Use this model to predict the $i$th response, say it is $\hat{y}_i$. and compute $\text{MSE}_i = (y_i - \hat{y}_i)^2$.
3. Compute cross validation error

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \text{MSE}_i.$$

Advantages:

1. This method has far less bias, since we are fitting the model to $n-1$ of the points.
2. Does not change depending on the random sample like the validation set method.

The only potential drawback is that it may be computationally intensive: we need to fit $n$ models.

# $k$-fold cross validation

1. Divide the data randomly into $k$ subsets (folds) of (roughly) equal size.
2. Start with the first fold as a validation set and use the remaining $k-1$ folds to fit the model.
3. Compute the error of the fitted model in the held-out fold.
4. Repeat steps 2 and 3 by using the second, third and so on folds as the hold-out fold with the remaining $k-1$ folds to fit the model.
5. Average the error across all the $k$ fitted models to estimate the cross-validation error.

Some of the commonly used choices for $k$ are 5 or 10.
When $k = n$, this reduces to LOOCV.

# $k$-fold cross validation