Model Assessment and Selection (I)

Ke Chen

Reading: Sects. 2.2, 5.1, 7.1 [Intro Stat Learn Python]

https://www.statlearning.com/

https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print10.pdf

Lecture Goal

- Understanding the ultimate goal of machine learning
- Model assessment and selection: motivation, tasks and methodology
- Empirical methods: held-out validation, K-fold and leave-one-out cross-validation
- Practical aspects of empirical methods

Learning Model and Ultimate Goal

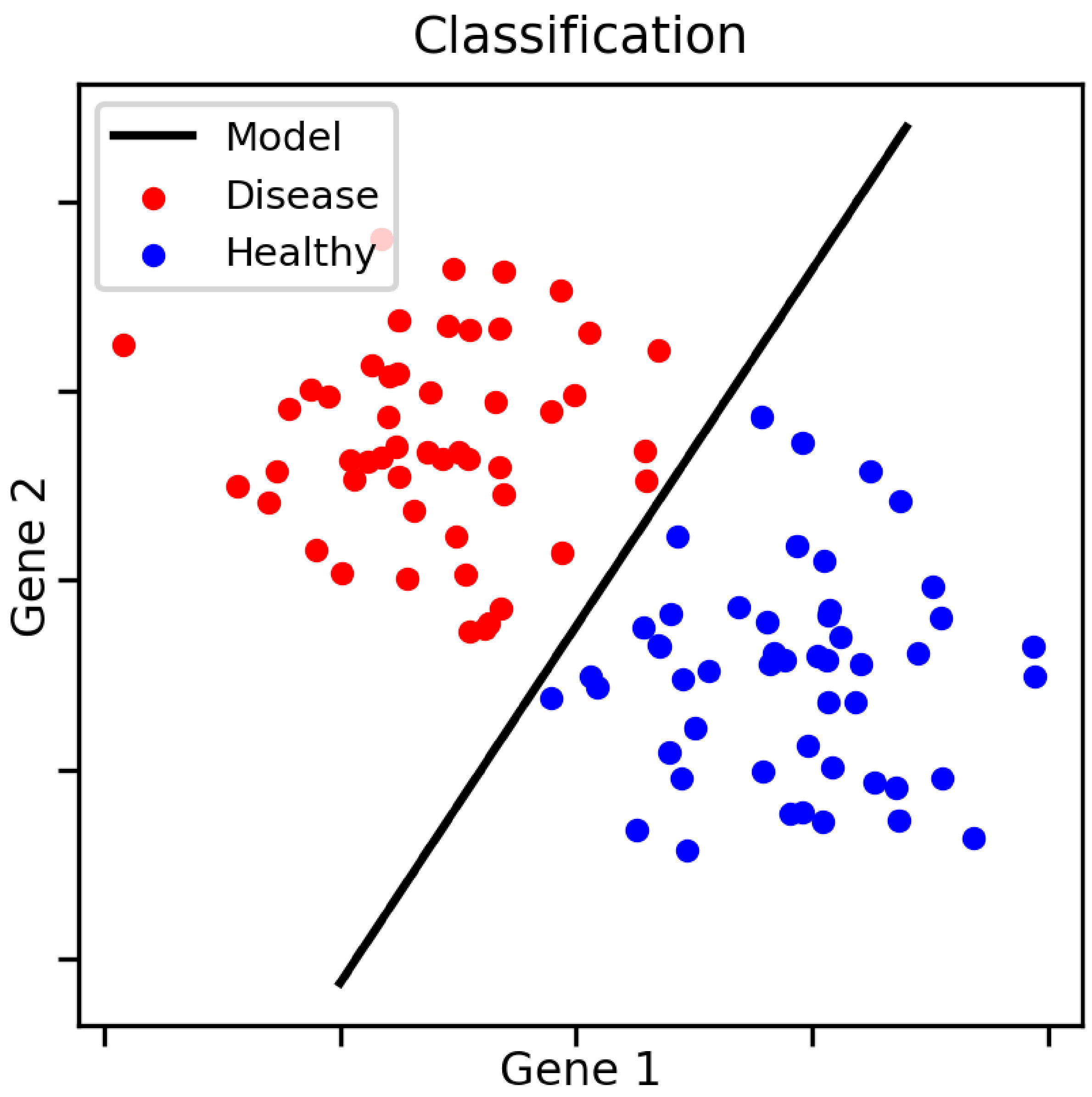
- Learning models are established based on a data-driven principle; a typical process is to learn/discover generic regularities (or distributions from a probabilistic perspective) based on a training data set (a collection of observations – a sample).

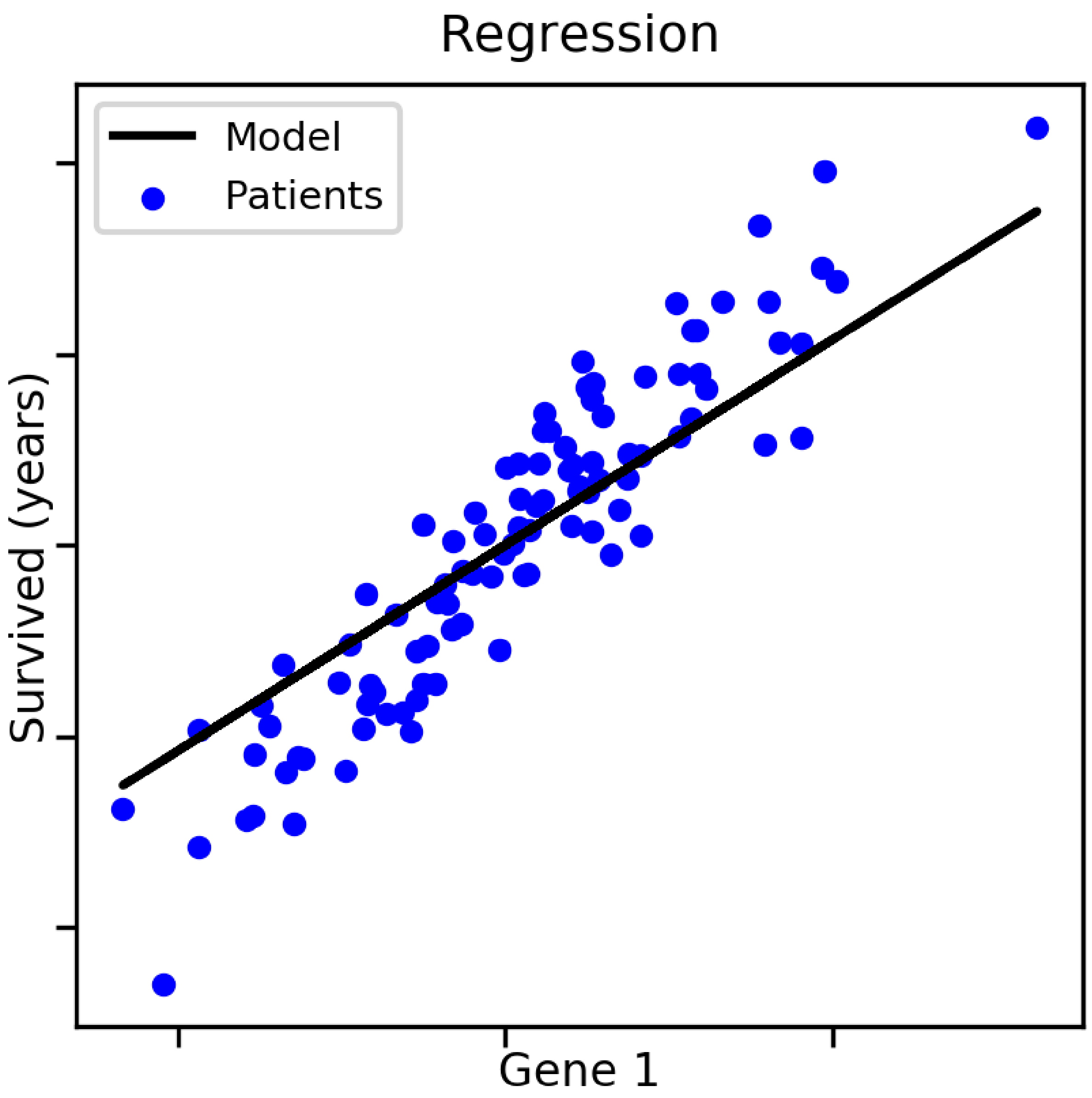
- Statistical learning works in an inductive learning manner; learning/discovery of a generic regularity/rule from a number of examples/instances generated by such a regularity/rule.

- The nature of statistical (machine) learning always limits the training data to only a specific sample of a population to be modelled.

- The ultimate goal of statistical learning is towards inductive bias or generalisation; a learning model can predict output correctly given input that have not been encountered during learning. In other words, a learning model should be able to generalise the regularity/rule learned from observed data to unseen data.

Learning Model and Ultimate Goal





Learning Model and Ultimate Goal

- For a task, different learning models could be employed to learn or discover the regularity or rules behind from training data.

- Different models have different capacities in problem solving via learning, which are often quantified by their complexities or flexibilities.
  - For example, polynomial regression models; a quadratic model of degree 2 has a larger capacity than a linear model of degree 1

    Linear model: $f(x; \boldsymbol{\beta}) = \beta_0 + \beta_1 x$

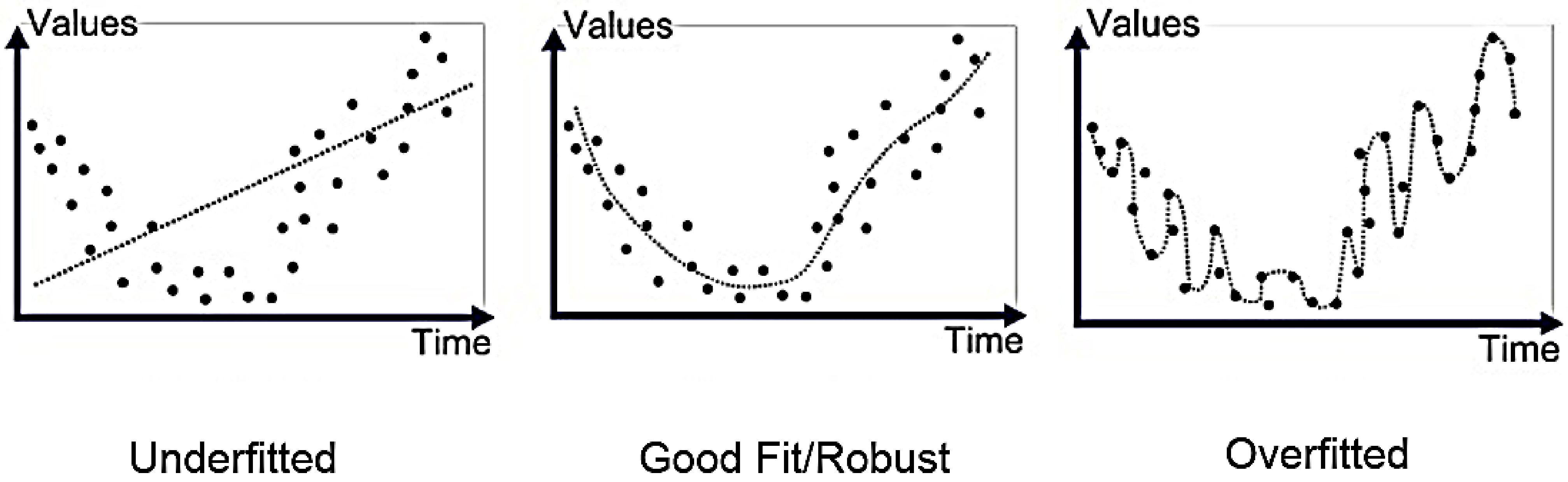
    Quadratic model: $f(x; \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2$

- To attain the ultimate goal of statistical learning (inductive bias or generalisation), a proper model of certain complexity (flexibility) must be used for a specific data set!

Learning Model and Ultimate Goal

- For a specific data set, what happens if a proper model is NOT used ……

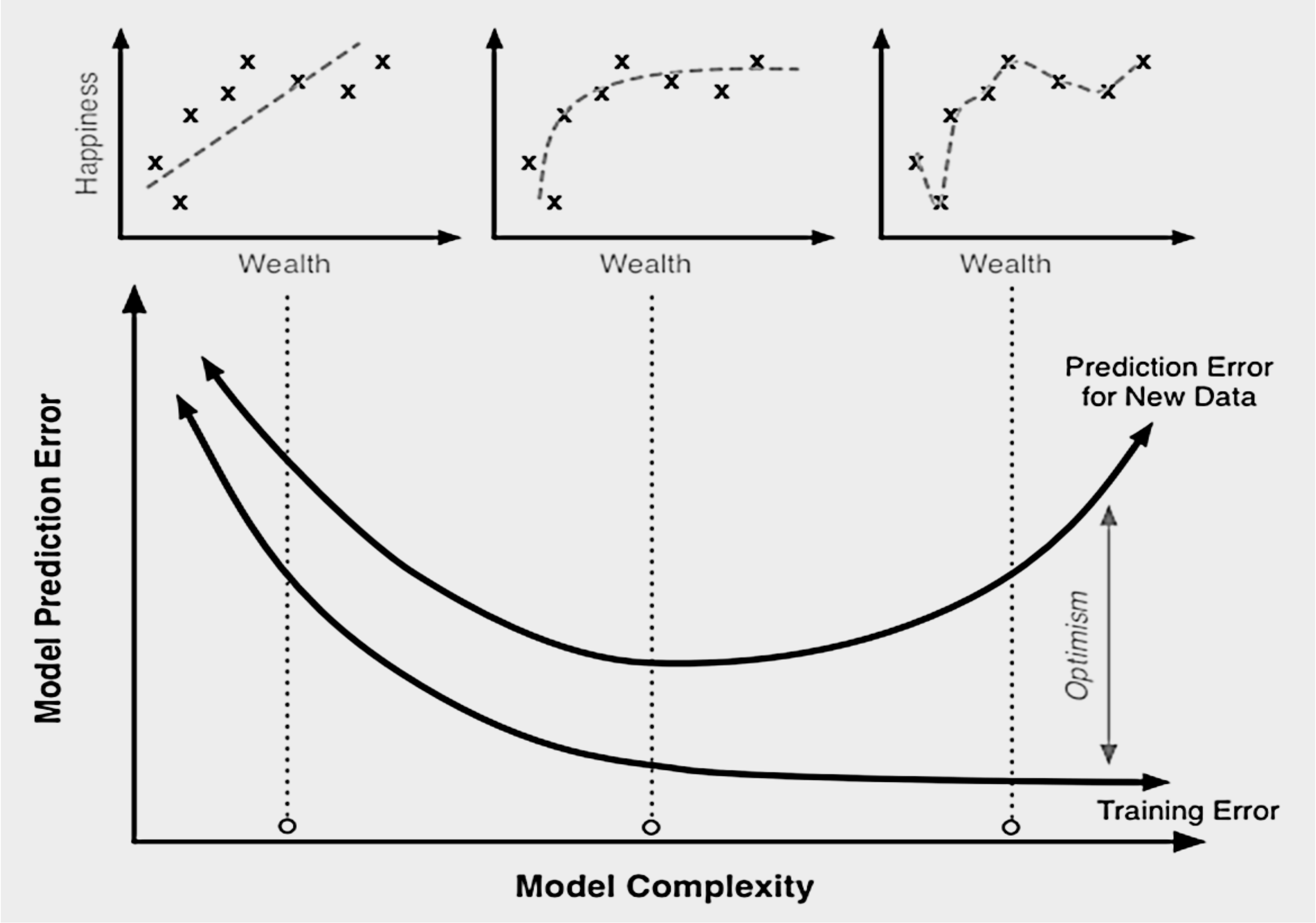
- Under-fitting (Underfitting) vs. Over-fitting (Overfitting)
  - Underfitting: a model has a too limited capacity to capture the underlying regularity of the data
  - Overfitting: a model (of a higher complexity than what is required) closely explain training data but fail to generalise the regularity found to unseen data (against the ultimate goal!)



Bias, Variance and Their Trade-off

- How to quantify those phenomena is essential for statistical learning!

- General observation



Bias, Variance and Their Trade-off

- In general, we can use two “measurements” to quantify the phenomena.

- Regression setting: $Y = f(X) + \varepsilon;\ E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma_\varepsilon^2$
  - Learning an approximation of $f(X)$: $\hat{f}(X, \Theta)$ based on training data (a sample of the population)
  - Different training data sets lead to different $\hat{f}(X, \Theta)$ even when the same learning model is used.

- Bias vs. Variance
  - “Averaging” performance on all samples of a population, can be characterised by two aspects:

    Bias is defined as

$$\text{Bias}^2(\boldsymbol{x}_i) = [E\hat{f}(\boldsymbol{x}_i, \Theta) - f(\boldsymbol{x}_i)]^2$$

Variance is defined as

$$\text{Var}(\boldsymbol{x}_i) = E[\hat{f}(\boldsymbol{x}_i, \Theta) - E\hat{f}(\boldsymbol{x}_i, \Theta)]^2$$

Bias, Variance and Their Trade-off

- Decomposing test error in terms of bias and variance

$$\mathrm{Err}(\boldsymbol{x}_i) = E[y_i - \hat{f}(\boldsymbol{x}_i, \Theta)]^2$$

$$= E[y_i - E\hat{f}(\boldsymbol{x}_i, \Theta) + E\hat{f}(\boldsymbol{x}_i, \Theta) \;-\hat{f}(\boldsymbol{x}_i, \Theta)]^2$$

$$= E[[(f(\boldsymbol{x}_i) + \varepsilon) - E\hat{f}(\boldsymbol{x}_i, \Theta)] + [E\hat{f}(\boldsymbol{x}_i, \Theta) \;- \hat{f}(\boldsymbol{x}_i, \Theta)]]^2$$

$$= E[[(f(\boldsymbol{x}_i) - E\hat{f}(\boldsymbol{x}_i, \Theta)] + [E\hat{f}(\boldsymbol{x}_i, \Theta) \;- \hat{f}(\boldsymbol{x}_i, \Theta)] + \varepsilon]^2$$

$$= [E\hat{f}(\boldsymbol{x}_i, \Theta) \;- f(\boldsymbol{x}_i)]^2 + E[\hat{f}(\boldsymbol{x}_i, \Theta) \;- E\hat{f}(\boldsymbol{x}_i, \Theta)]^2 + E(\varepsilon^2)$$

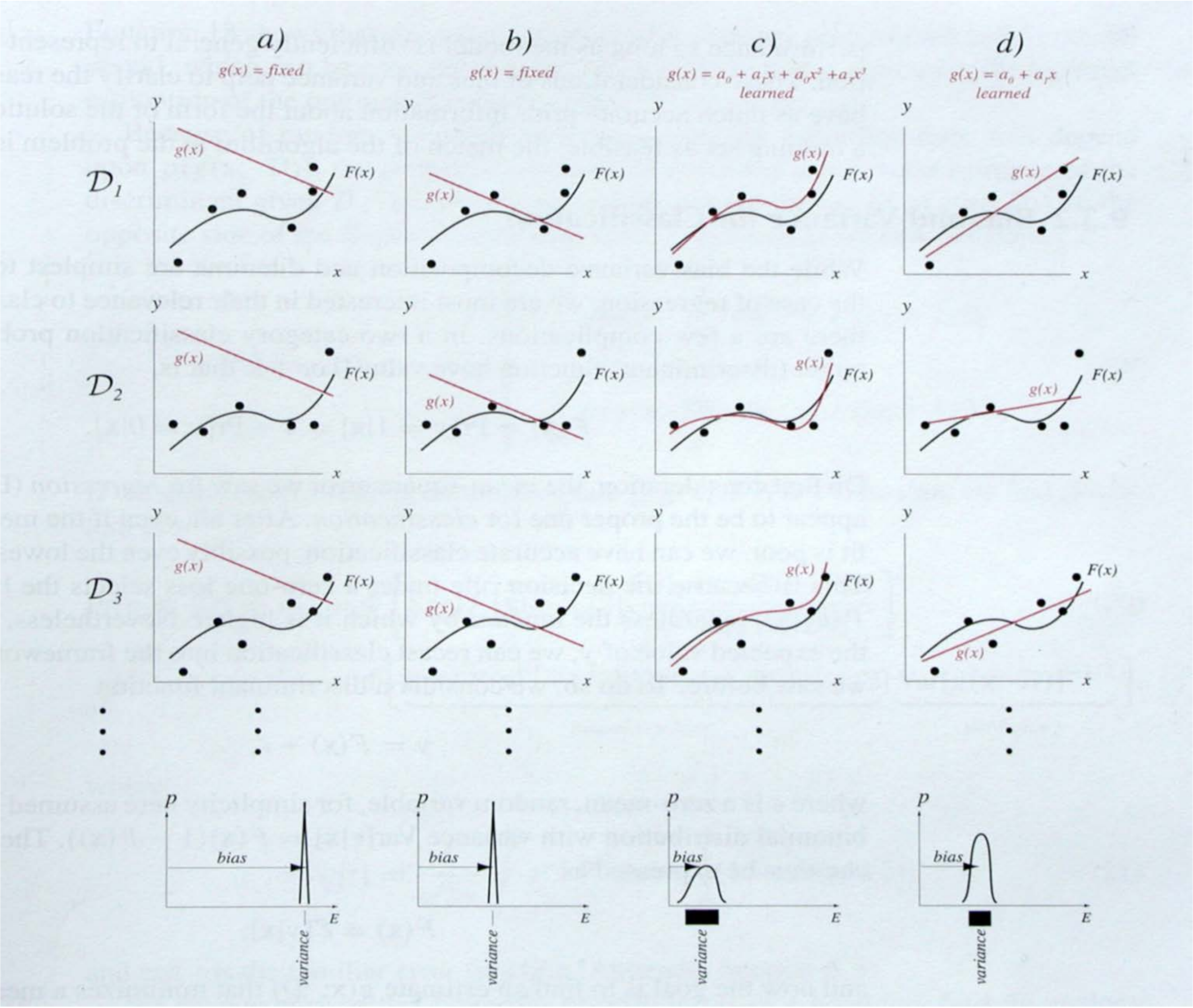
$$= \quad \mathrm{Bias}^2(\boldsymbol{x}_i) \quad + \quad \mathrm{Var}(\boldsymbol{x}_i) \quad + \quad \sigma_\varepsilon^2$$

Amount by which average estimate differs from the true mean

Expected deviation of $\hat{f}(\boldsymbol{x}, \Theta)$ around its mean

Irreducible error

Bias, Variance and Their Trade-off

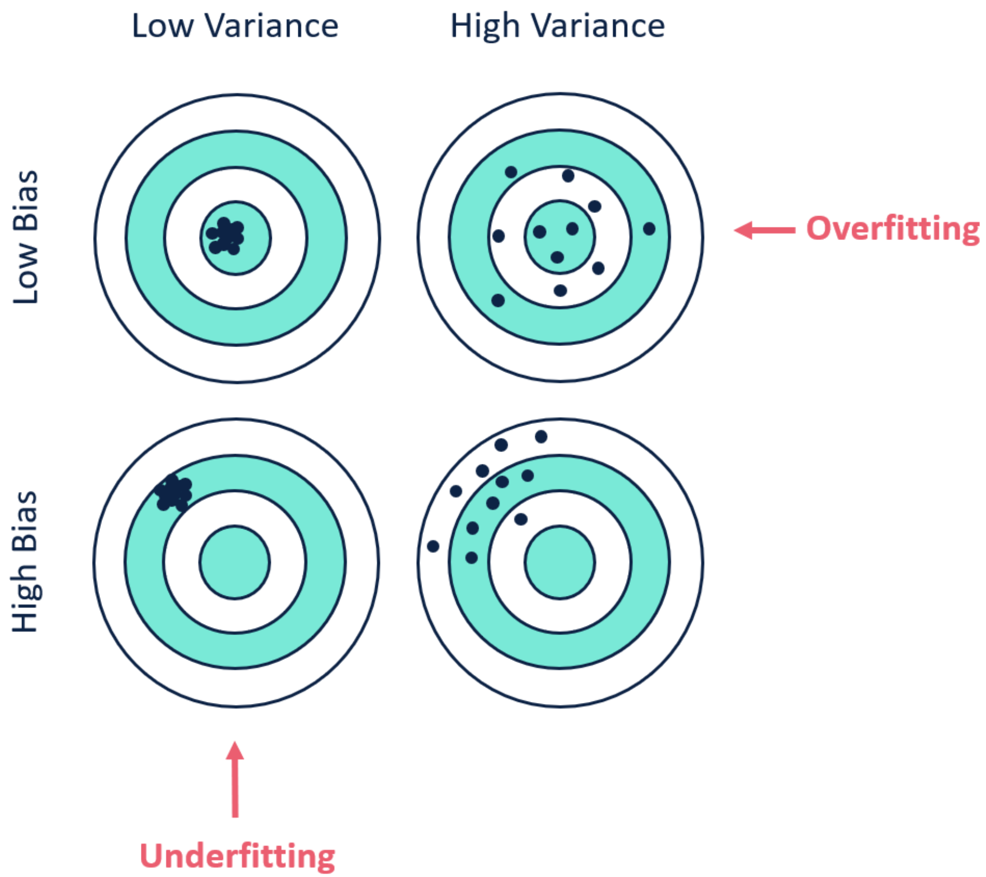


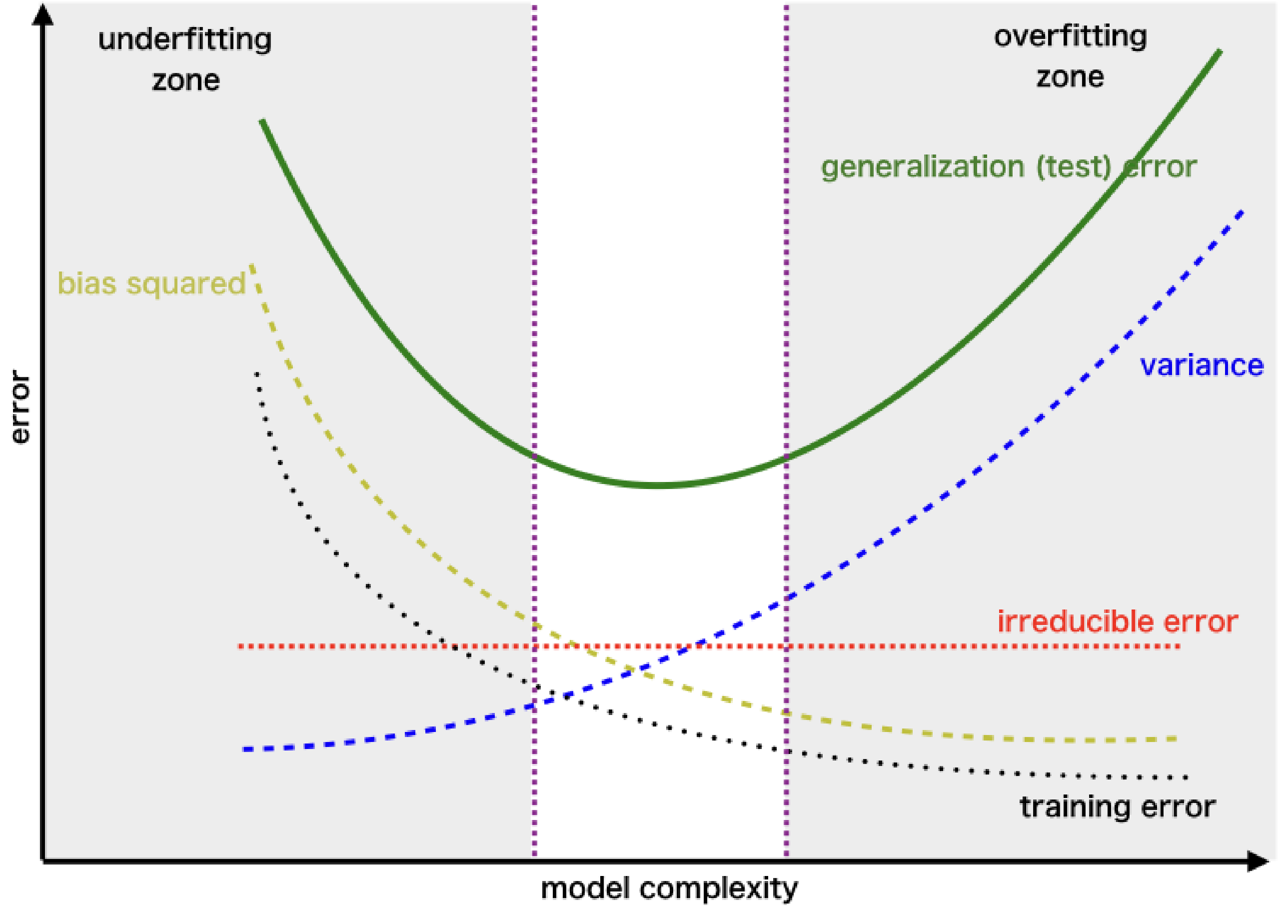
Example: decomposing test error in terms of bias and variance

Duda, Hart & Stork, Pattern Classification, 2nd Ed., 2001.

Bias, Variance and Their Trade-off

- The bias-variance trade-off (bias-variance dilemma)
  - To minimise the test error, both bias and variance should be reduced simultaneously in theory.
  - In reality, however, reducing bias is often at a cost of increasing variance and vice versa. (Dilemma)
  - The complementarity of bias and variance demands a trade-off between them in statistical learning.





Model Assessment and Selection

- To evaluate the performance of learning models in term of the ultimate goal, there are two essential tasks: model assessment and model selection.
  - Model selection: For a number of candidate models trained on a data set, estimating performances of different models to choose the best one that leads to the least prediction error on unseen data.
  - Model assessment: Having chosen a model and trained it on a training data set, estimating the prediction error on new data that are never involved during learning.
- Measuring errors: loss (cost/score) functions

Regression: $l(Y, \hat{f}(X)) = \left(Y - \hat{f}(X)\right)^2$ (residual sum of square loss)

Classification: $l(Y, \hat{f}(X)) = \mathrm{I}\left(Y \neq \hat{f}(X)\right)$ (0-1 loss)

- In reality, however, it is impossible to do model assessment and selection properly by simply measuring training errors ($\overline{\mathrm{err}} = l(Y_{tr}, \hat{f}(Y_{tr}))$). In fact, the test error (Err) is (nearly always) higher than training error (i.e., $\overline{\mathrm{err}} < \mathrm{Err}$).
- Methodologies for model assessment and selection: Empirical vs. Analytical

Empirical Methodology

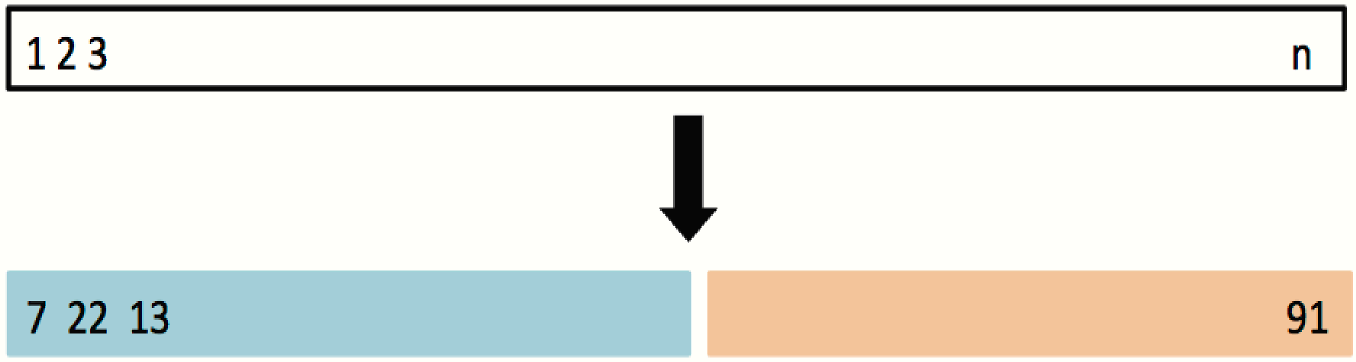
- For model assessment and selection on a sample (available for a learning model), an intuitive idea is "simulating" a training-test scenario via re-sampling techniques.

- Re-sampling techniques allow for splitting a data set available into subsets randomly.

- An empirical method for model assessment and selection would use only some subsets of data for training a learning model while reserving the remaining data as "simulated" test data for validation.
  - Held-out validation
  - Cross-validation
    - *K*-fold cross-validation
    - leave-one-out cross-validation (LOOCV)

Held-out Validation

- Held-out validation is a straightforward manner to “simulate” a training-test scenario.
- A data set is randomly split into two subsets in a specific ratio (e.g., 80% vs. 20%).



Training Subset

Validation Subset

- Training subset is used for training a learning model, while validation subset is used for “model assessment”. For model selection, all the candidate models work on the same condition so model comparison is done based on their “model assessment”.
- For reliability, multiple trials of (independent) held-out validation is often conducted.
- Pro & con: computationally efficient; only a subset of data used in training a model

Held-out Validation

- **Example:** predicting mpg from horsepower

  Polynomial models of different degrees

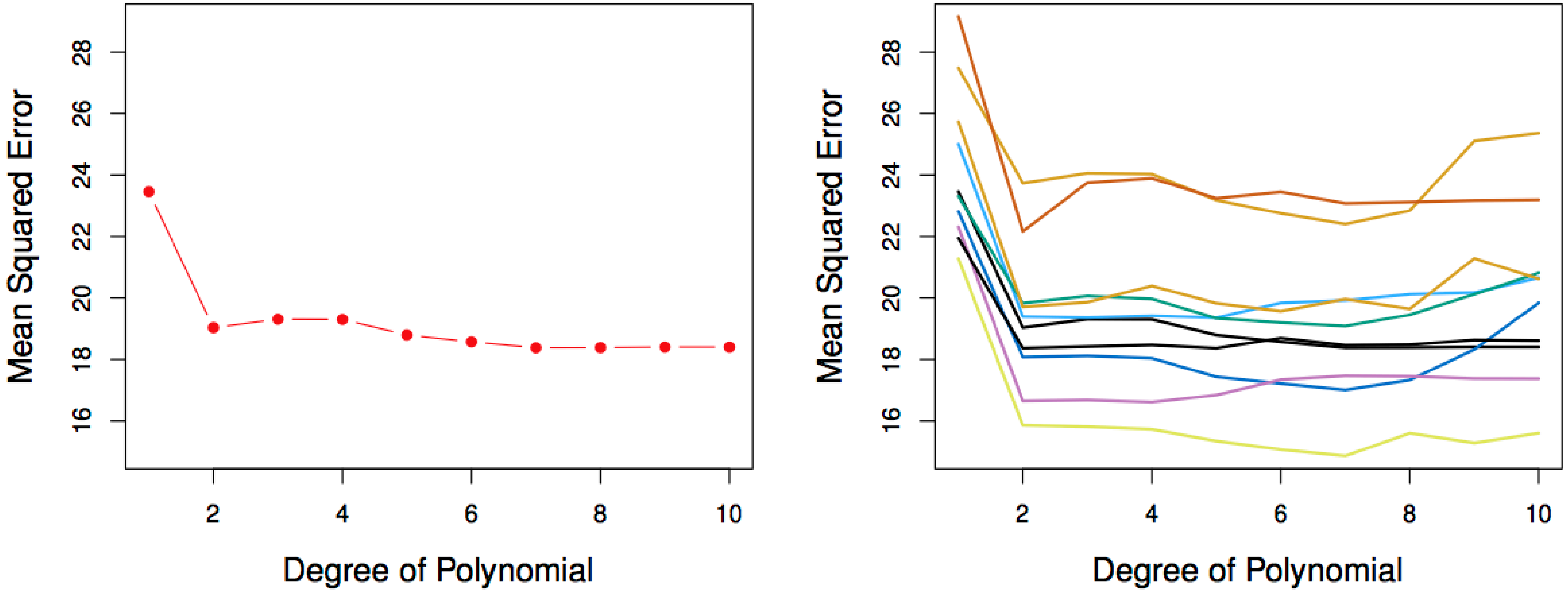
  - Linear model (1 degree): mpg ~ horsepower
  - Quadratic model (2 degree): mpg ~ horsepower + horspower^2
  - ......

  Which model gives a better fit?

  - We randomly split 392 observations into training and validation data sets (50/50), and we fit both models using the training data.
  - Next, we evaluate those models of different degrees using the validation data set.
  - **Winner** = model with the lowest validation MSE

Held-out Validation

- **Example:** predicting mpg from horsepower (cont.)



Left Panel: Validation error estimates for a single split into training and validation data sets.

Right Panel: Validation error estimates for 10 trials (splits); shows the validation MSE is highly variable.

MANCHESTER 1824
The University of Manchester

Cross-Validation

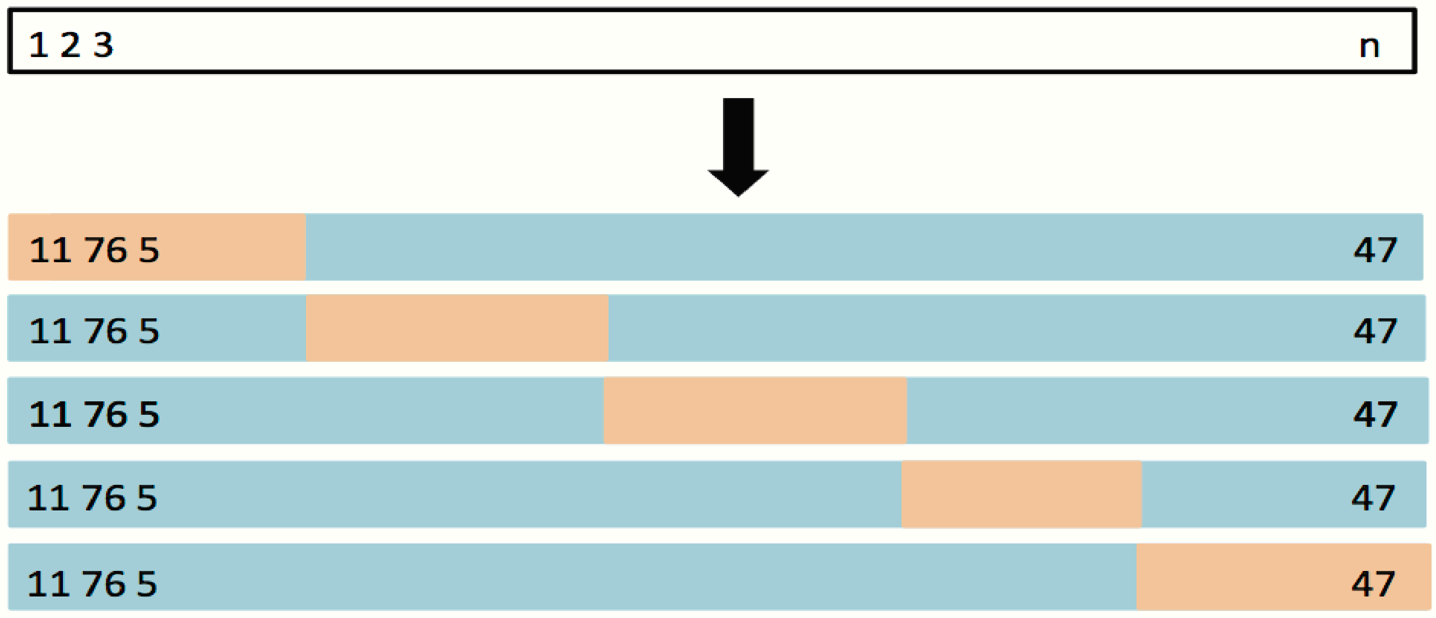
- Cross-validation is going to overcome the limitation of held-out validation
  - Training and validation subsets are exclusive; only a part of data are used for training a model.
  - Random splitting may lead to unstable (highly variable) validation error rates.
  - The above problems are exacerbated when only fewer training data (a small sample) are available. In this case, the validation error rate often tends to be over-estimated.

- Cross-validation is probably the most commonly used empirical method for model assessment, model selection, model comparison, hyper-parameter tuning and so on.

- Unlike the held-out validation, all the data (in the sample available for training) are always used in both training and validation in turn.

K-fold Cross-Validation

- *K*-fold cross-validation randomly splits a data set into *K* subsets of equal size.



  - The first subset is treated as a validation subset, and the model is trained on the remaining $K - 1$ subsets. The error is measured on the validation subset.
  - The process is repeated *K* times, taking out a different subset each time.
  - By averaging the *K* estimates of the validation error, we get an averaging validation error rate for each model. Multiple trials may have to be done when *K* is small.

K-fold Cross-Validation

- Let the K folds be $C_1, \ldots, C_K$, where C_k denotes the equal-sized subsets of the data in fold k. There are n_k data in fold k: if N is a multiple of K, then $n_k = n / K$. (Otherwise, split data into K folds as even as possible).
- Compute: $\mathrm{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k$ in terms of regression

  where $\mathrm{MSE}_k = \frac{1}{n_k} \sum_{i \in C_k} (y_i - \hat{f}(\boldsymbol{x}_i))^2$ and $\hat{Y}_i$ is the fitted value for observation i, obtained from the data with fold k removed.

1	2	3	4	5
Train	Train	Validation	Train	Train

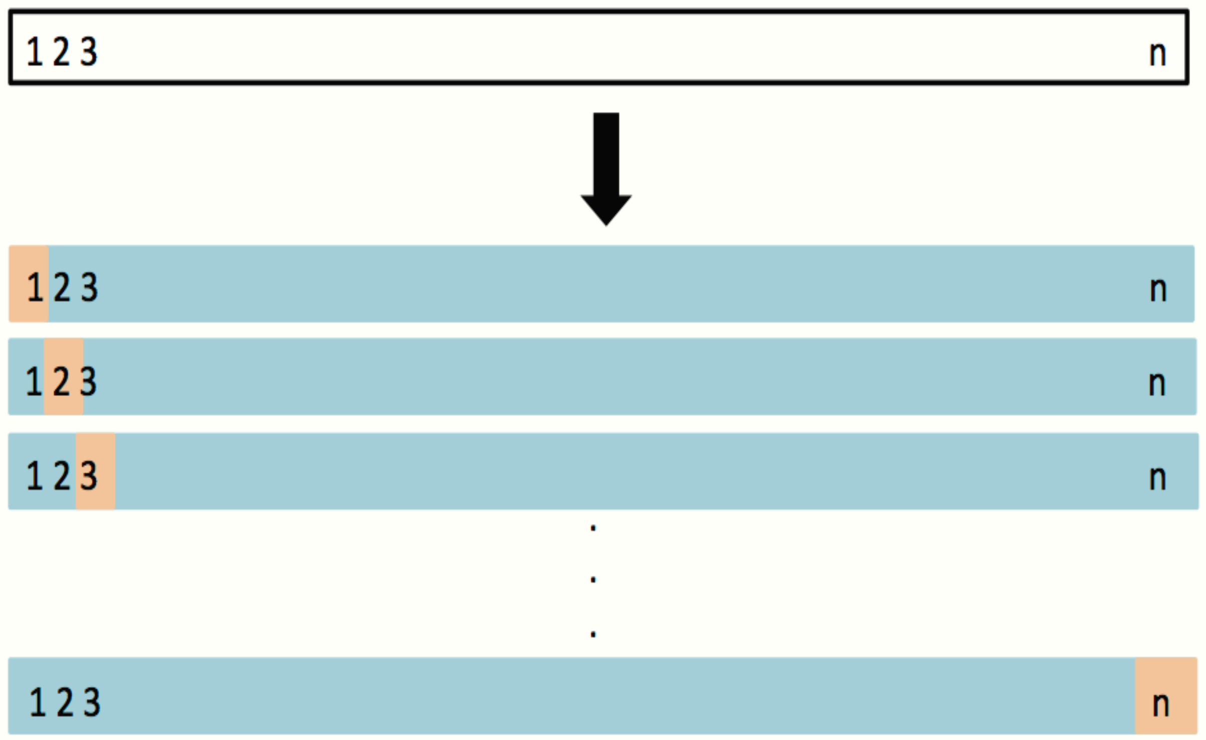
$\boldsymbol{MSE_3}$ **estimated on** $\boldsymbol{C_3}$

Leave-One-Out Cross-Validation (LOOCV)

- LOOCV is a special case of K-fold cross-validation by setting $K = n$.

LOOCV Algorithm

- Split the entire data set of size n into:
  - Blue = training subset of n-1 data
  - Beige = validation subset of 1 datum
- Fit the model using the training subset
- Evaluate the model using validation subset and compute the corresponding MSE.
- Repeat this process n times, producing n validation errors. The average of these n validation errors for each model.



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i$$

Leave-One-Out Cross-Validation (LOOCV)

- LOOCV is a special case of K-fold cross-validation by setting $K = n$ but significantly different from held-out validation and K-fold cross-validation settings when $K << n$.

- LOOCV is far more stable and hence not to overestimate the validation error rate.

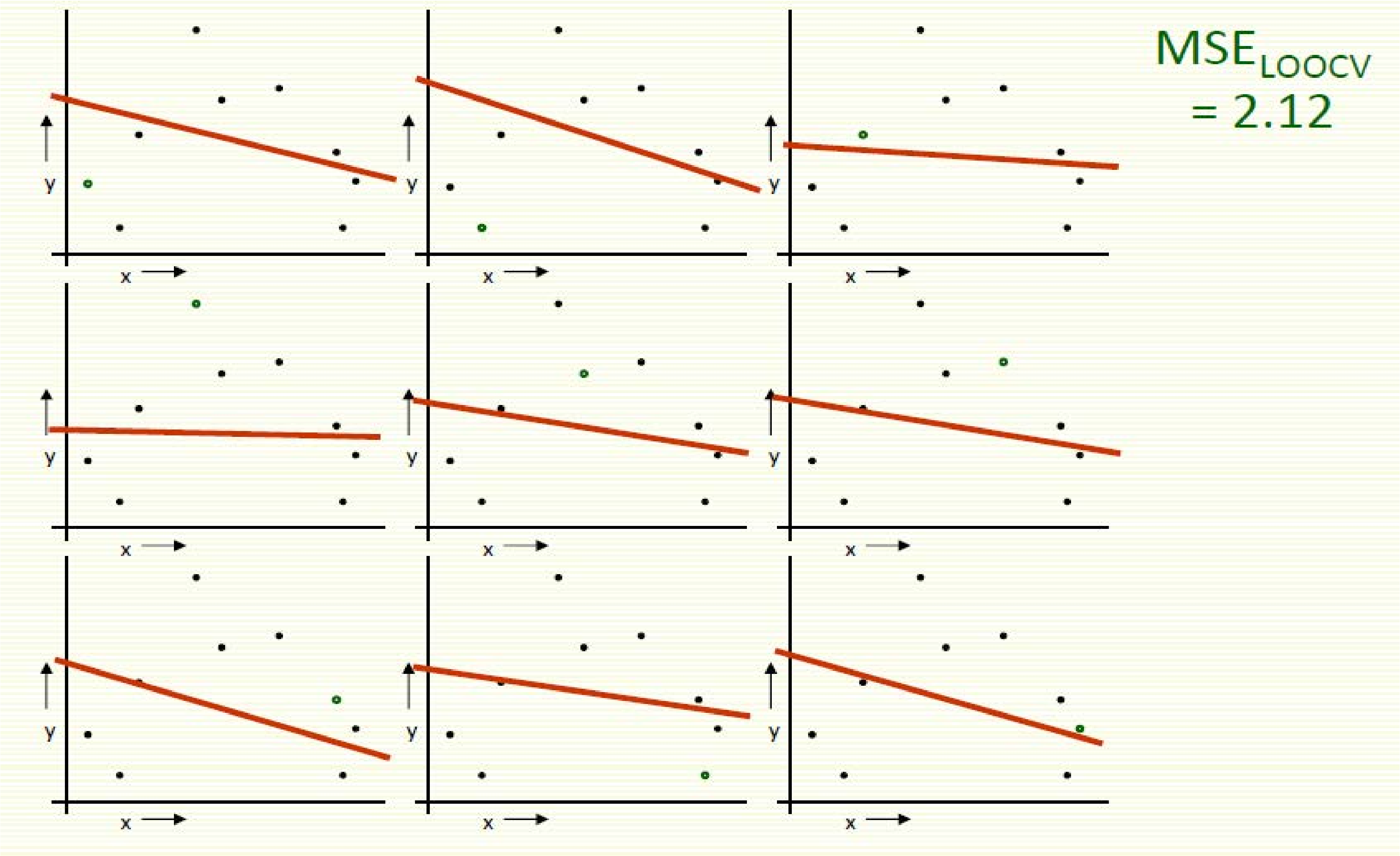
- Performing LOOCV multiple times always yields the same results because there is no randomness in the training/validation splits; no longer a re-sampling method!

- LOOCV is computationally intensive because the model has to be fit n times. For all linear regression models, there is a short-cut via generalised cross-validation (GCV):

$$\mathrm{GCV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}(\boldsymbol{x}_i)}{1-\mathrm{df}(\hat{\boldsymbol{y}})}\right)^2, \quad \text{where } \mathrm{df}(\hat{\boldsymbol{y}}) = \frac{\sum_{i=1}^{n} cov(y_i, \hat{y}_i)}{\sigma_\varepsilon^2}, \ \sigma_\varepsilon^2 = \mathrm{RSS}/(\mathrm{n}\text{-}2).$$

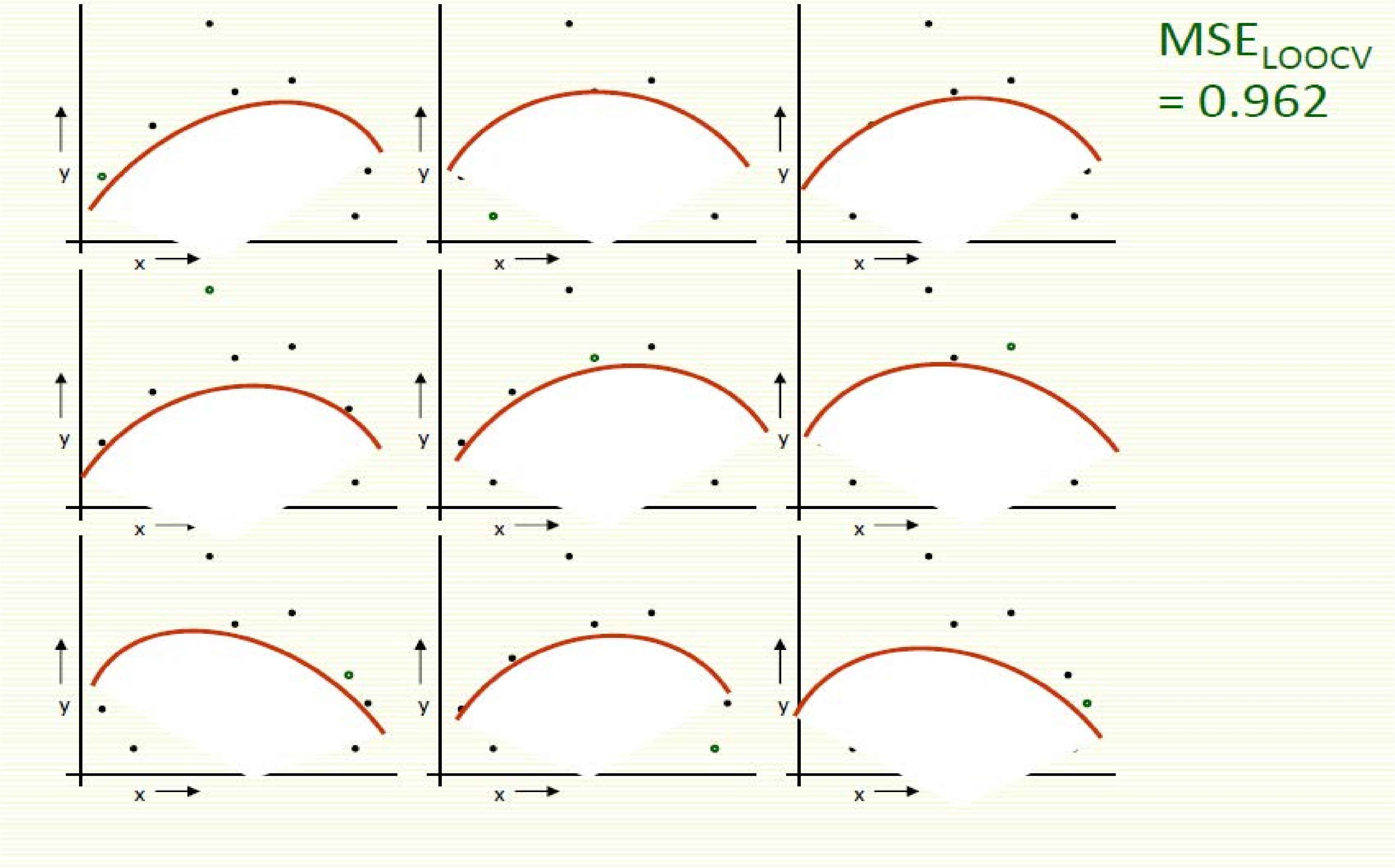
Leave-One-Out Cross-Validation (LOOCV)

- Example: applying LOOCV to a toy data set with linear models



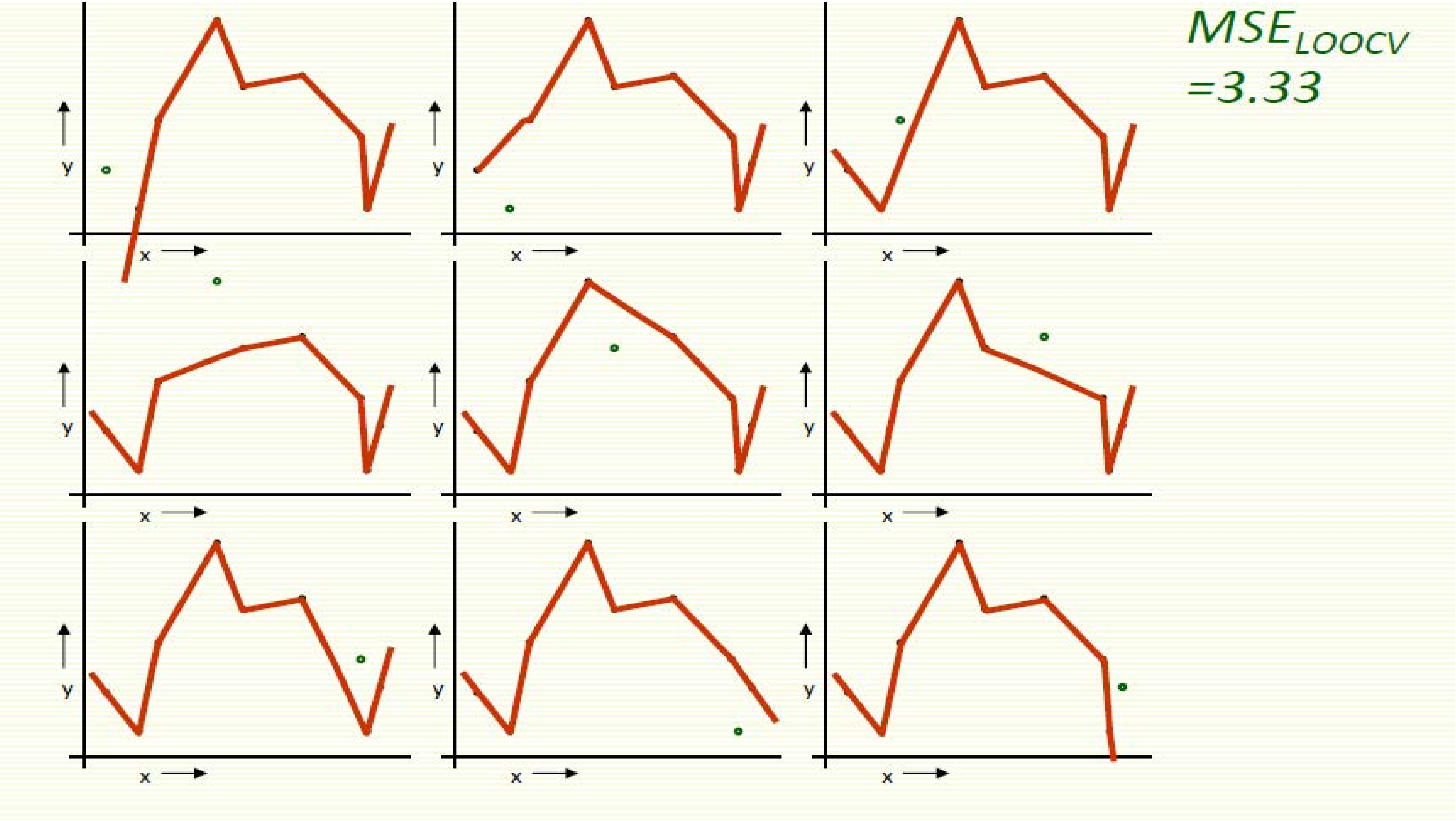
Leave-One-Out Cross-Validation (LOOCV)

- Example: applying LOOCV to a toy data set with quadratic models



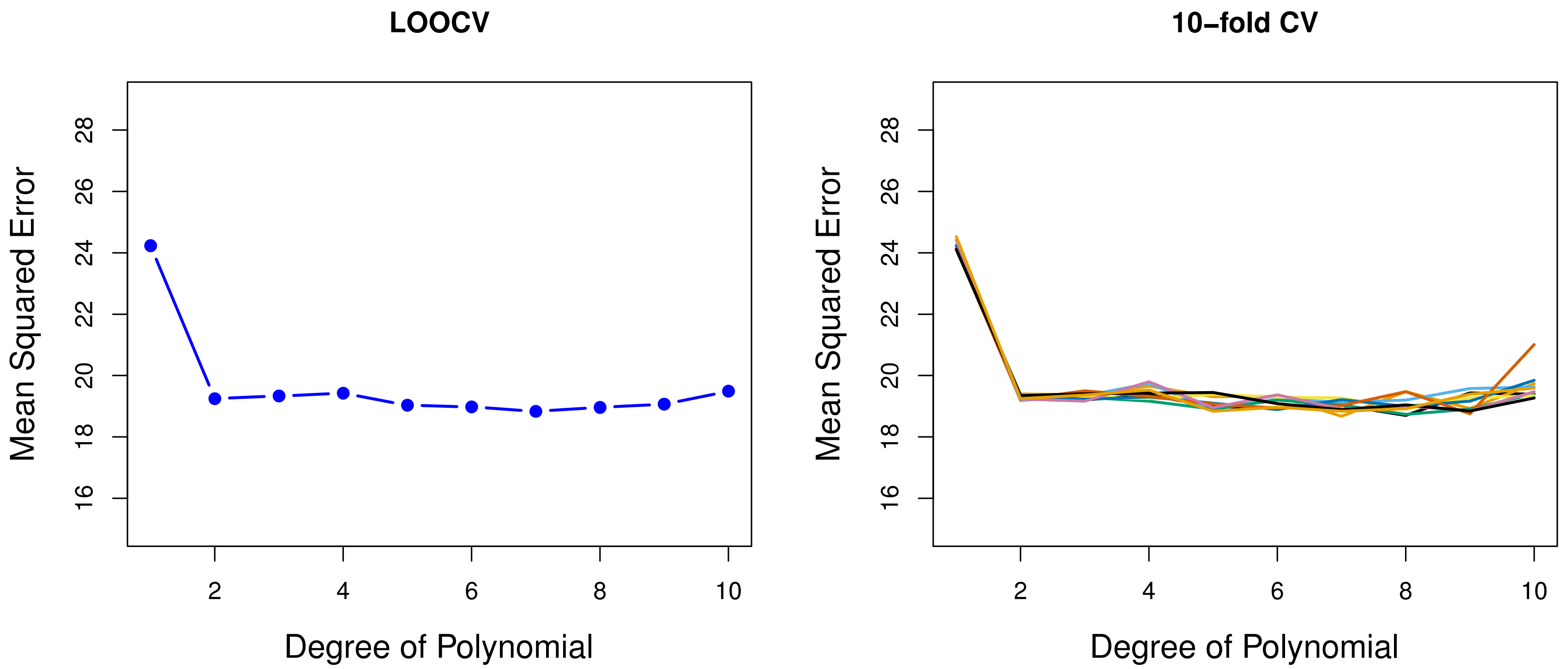
Leave-One-Out Cross-Validation (LOOCV)

- Example: applying LOOCV to a toy data set with [“joint the dots” models]



K-fold Cross-Validation vs. LOOCV

- **Example:** predicting mpg from horsepower



Left Panel: LOOCV Error Curve

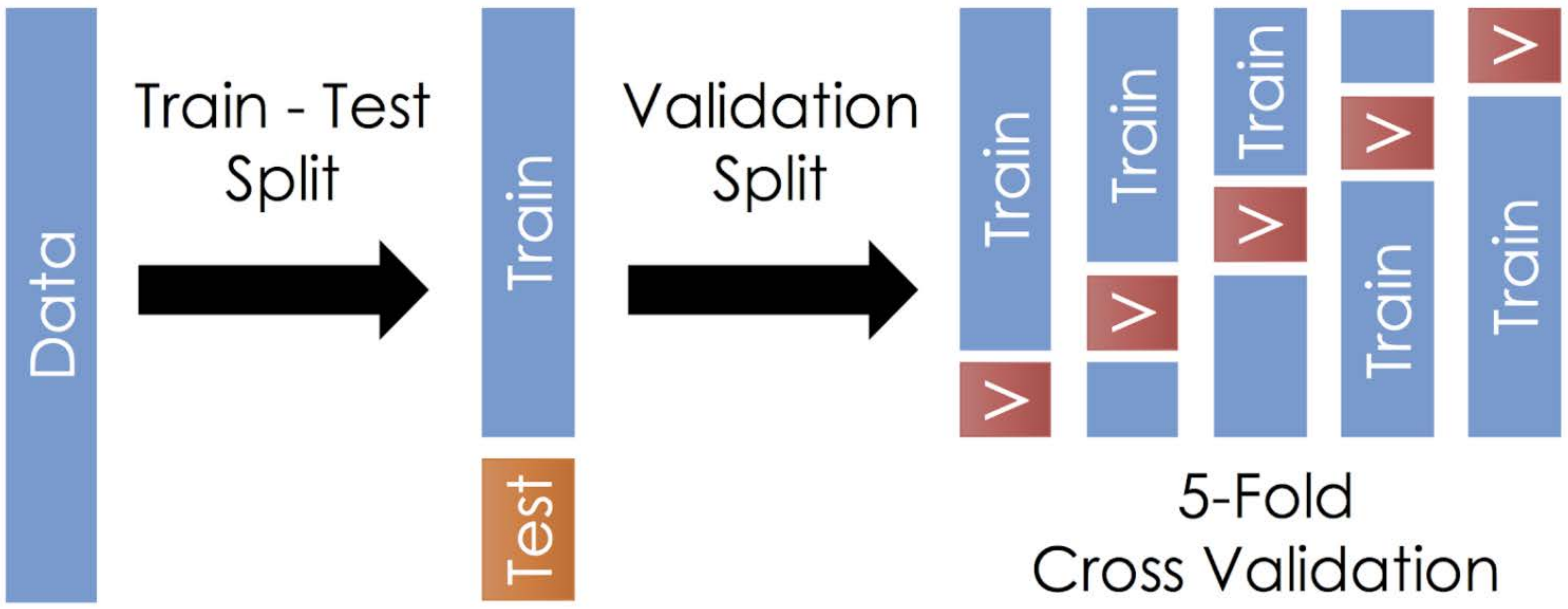
Right Panel: 10-fold CV run 9 separate times (i.e. 9 trials), each trial with a different random split of the data into ten subsets of (almost) equal size.

Practical Aspect of Empirical Methods

- Which is better, LOOCV or *K*-fold cross-validation?
  - LOOCV is more computationally intensive than *K*-fold cross-validation in general.
  - From the perspective of bias reduction, LOOCV is preferred to *K*-fold cross-validation when $K < n$, as more examples are available for training a learning model.
  - However, LOOCV often has higher variance than *K*-fold cross-validation when $K < n$.
  - Thus, we see the bias-variance trade-off between the two cross-validation methods.
- We tend to use *K*-fold cross-validation with $K = 5$ or $K = 10$, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance by considering the bias-variance trade-off.
- The empirical methods are often used for hyper-parameter tuning in a “complex” learning model. The model is trained multiple times with different hyper-parameters on training subset and errors are estimated on validation subset.

Practical Aspect of Empirical Methods

- In real applications, data available, aka development data, for building up a learning model are always split into two data sets for training and test to simulate the real scenarios.
- For example, *K*-fold cross-validation is applied on the development data set as follows:



Summary

- The ultimate goal of statistical learning enables a learning model to work for unseen (new) data instead of only working perfectly on seen (training) data
  - inductive bias and generalisation
  - model complexity (flexibility), underfitting vs. overfitting
  - the bias-variance trade-off in attaining the ultimate goal

- Model assessment and model selection are essential in statistical learning.

- Empirical methods for model assessment and selection
  - held-out validation
  - cross-validation: *K*-fold and leave-one-out
  - practical aspects of empirical methods