MONASH University
Information Technology

FIT5201 - Data analysis algorithms

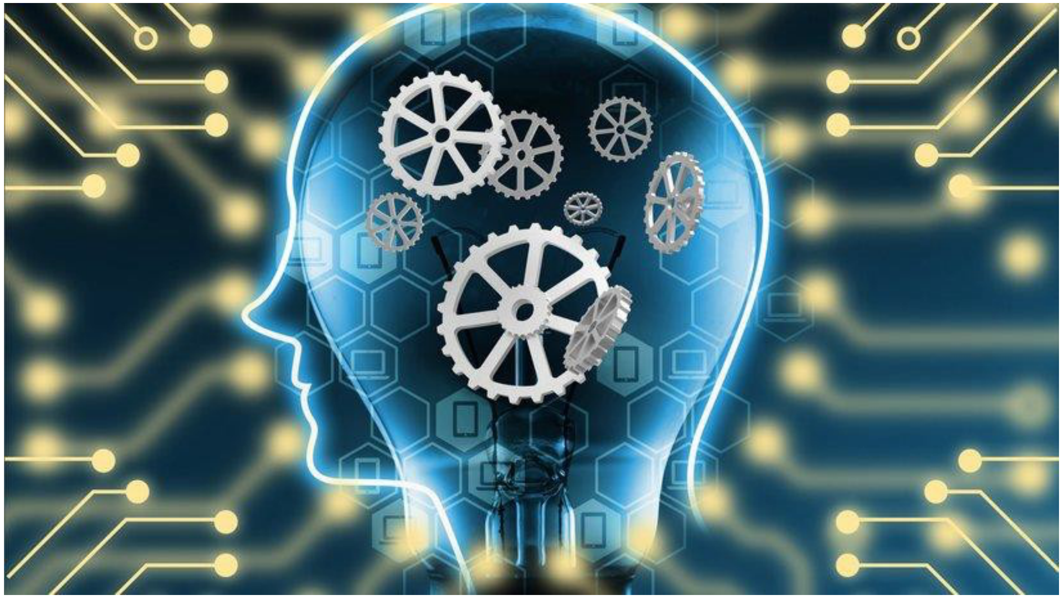
Module 1: Elements of Machine Learning

Objectives

- o Provide an introduction to machine learning theory
- o Part A (Week 1):
- - Understand the machine learning process
- - Understand key concepts of machine learning
- - Understand how to select a good learning model
- o Part B (Week 2):
  - - Another key concept: uncertainty
  - - Understand probabilistic machine learning

Part A

- An Introduction to Machine Learning
- The Fundamental Concepts of Machine Learning and model selection



What is Machine Learning?

Human

Learn from Experience

Machine

Learn from Experience?

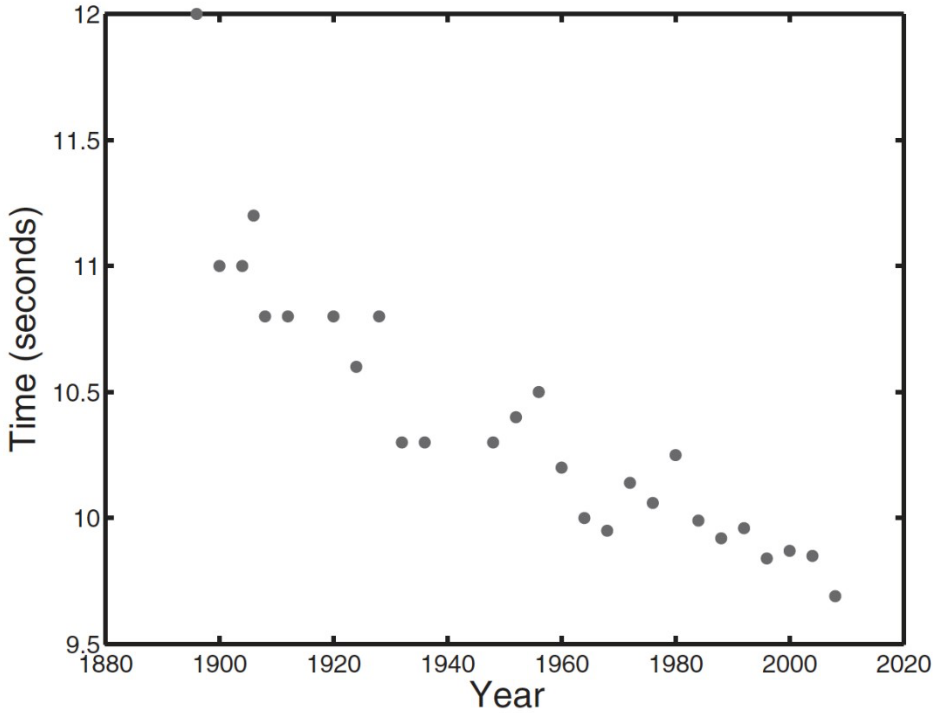
Learn from Data (sensor data, input data, etc)

- Why: automation & learn patterns from large data

What is Machine Learning?

- Process:
  - Learn a functional relationship between a set of attribute (or input variables which can be obtained by data Wrangling) and the associated response or target variables.
- Purpose
  - **Prediction**
    - Predict the target/response value for any (possibly new) values of the attribute variables
    - Main focus
  - Inference
    - Understand the way that the target variable is affected as the input variables change

Predict Olympic gold medal winning time

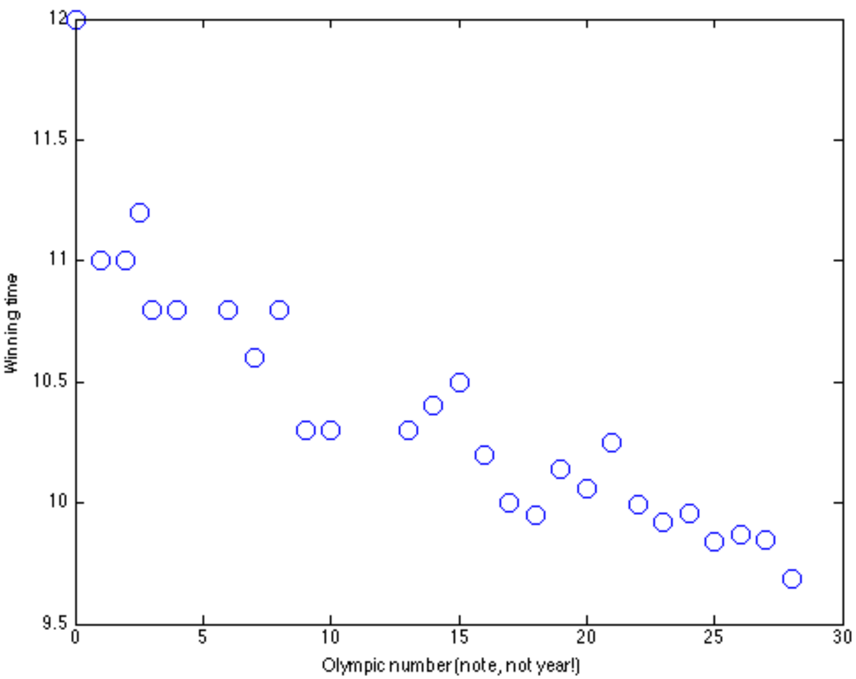


Winning men's 100m times at the Summer Olympics since 1896.

Q: Can we teach a machine to learn from the data and to make predictions about the winning times in future games?

Olympic gold medal winning time problem

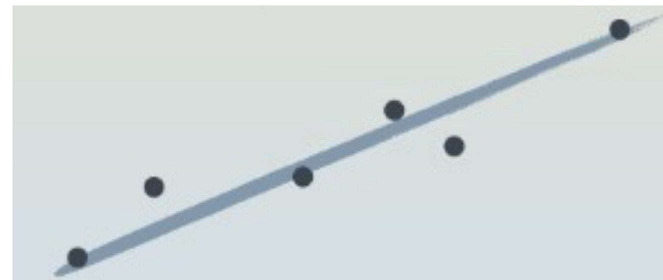
- Learning objective: learn a function between "Olympic Year" and "Winning Time". Then, use this function to make predictions about the winning times in future games.



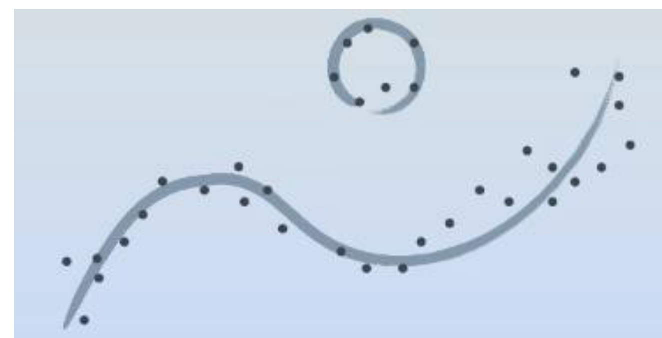
Q: How can we define such a function?

Model Definition

- Types of relationships between x and t
  - Linear relationship:
    - $\mathbf{t} = \mathbf{w_0} + \mathbf{w_1 x}$ ($\mathbf{w_0}$, $\mathbf{w_1}$: model parameters)



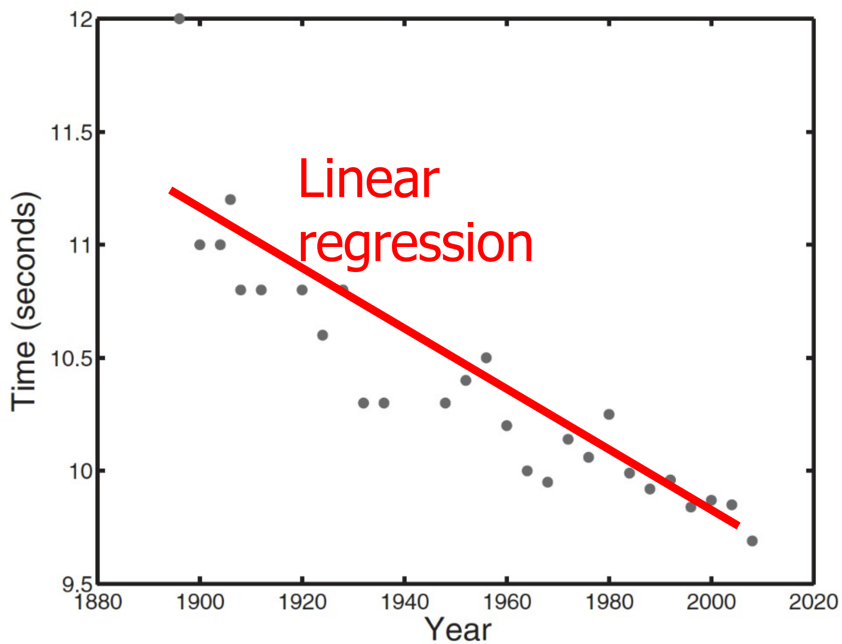
  - Non-linear relationship:
    - $\mathbf{t} = \mathbf{w_0} + \mathbf{w_1 x} + \mathbf{w_2 x^2} + \cdots + \mathbf{w}_M \mathbf{x}^M = \sum_{j=1}^{M} \mathbf{w_j x^j}$



  - Model parameters are something that need to be determined somehow.

Olympic gold medal winning time problem

❑ What could be a good model for this problem?



Q: Any functional relationship between “Olympic Year” and “Winning Time”?

- There is a **statistical dependence** between ”Olympic Year” and “Winning Time”
- The dependence could be adequately modelled with a straight line.
- Standard equation: $\mathbf{t} = \mathbf{w_0} + \mathbf{w_1 x}$ ($\mathbf{w_0}$, $\mathbf{w_1}$: model parameters)
- Learning task involves in using the data to choose suitable values for $\mathbf{w_0}$, $\mathbf{w_1}$.

Parameter Learning

- Training set
  - Used to learn the parameters **w**
- Test set
  - Once the model is trained (e.g., **w** is learned), it can be used to predict the winning time for new Olympic years (i.e., test set)
  - Generalization
    - The ability in predicting the target for new data (or test set) that differ from those used in the training set
  - The ultimate goal of machine learning is to build models that can generalize well to unseen examples.

Other important concepts

- **Supervised learning**
  - In the training set, the target variables of corresponding input variables are given
- **Unsupervised learning**
  - In the training set, only input variables are given
  - Clustering or visualization
- **Regression**
  - The target variables are real-valued and continuous
- **Classification**
  - The target variables are a finite number of discrete categories

Summary

❑ What is the Machine Learning Process?

- o Learn a model of the functional or statistical dependence between input attributes and target values from the training set. Then, use this model to make predictions about unseen examples on the test set.

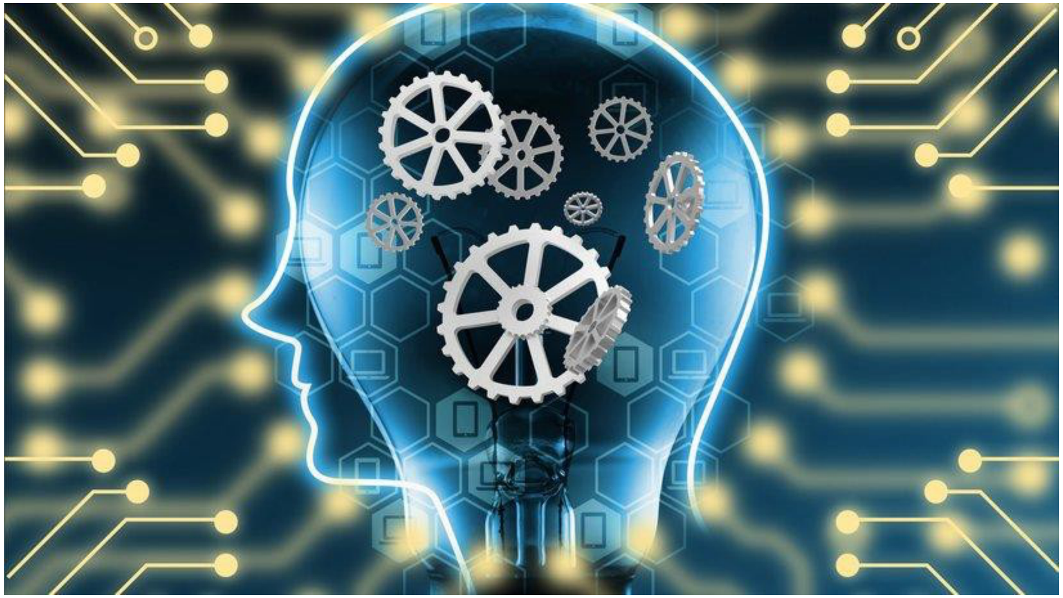
- o key concepts: training set & testing set, generalization, supervised & unsupervised learning, regression & classification

Key activity summary

- Given data, the key activities needed for generating a model for future use are:
  - The choice of model
  - Parameter learning on training data
  - Testing the generalization on test data

Part A

- An Introduction to Machine Learning
- **The Fundamental Concepts of Machine Learning and model selection**

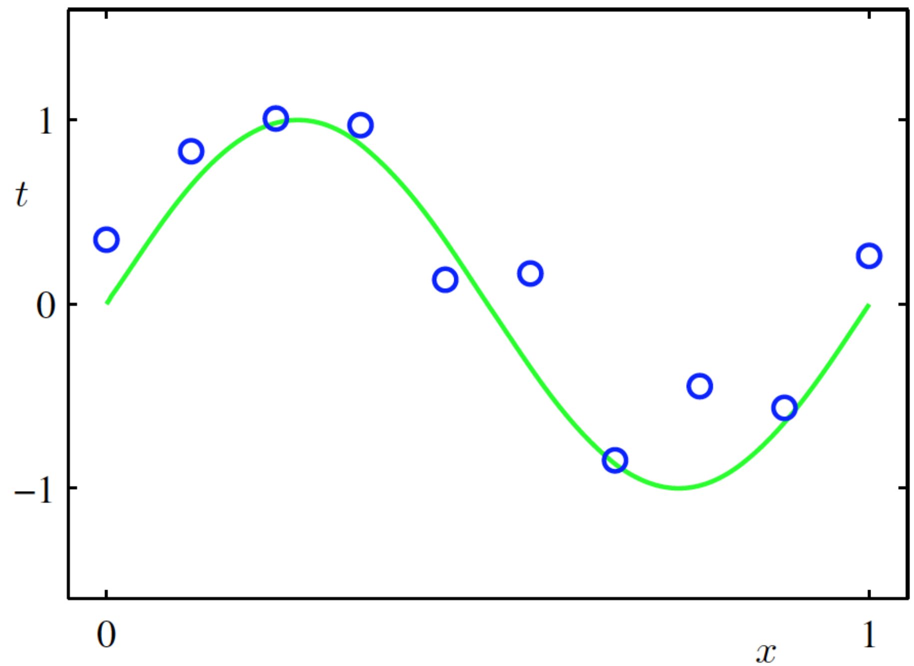


A regression problem example

- Training set: N (N=10) data points of pairs of x and t: $\{(x_1, t_1), \ldots, (x_N, t_N)\}$
- Underlying real function $t = \sin(2\pi x)$
- Noise exist

The figure shows the plot of a training set with N=10.

- Blue circle: 10 training points
- Green curve: the sin function (with noises) used to generate the training data.



A regression problem example

- Test set: 100 data points of pairs of x and t generated by the same process
- Noise exist

Objective in the Regression Problem

- To use the training set to build a model that can predict the value of t for a new input x accurately, without knowledge of the green curve.

- This involves implicitly trying to discover the underlying function

$$t = sin(2\pi x)$$

- To achieve good "generalisation" of the model by making accurate predictions for new data.

- Assess the generalisation of the trained model by comparing the predicted value and original value of for each input in test set.

Objective in the Regression Problem

- Note that, we are not allowed to use the test set while the model is trained. Otherwise, it would be cheating.

Learning a Model

- Challenges
  - Need to generalize from a finite data set.
  - The observed (or training) data are corrupted with noise: uncertainty existence!

- Assuming a model class
  - Parametric model: parameters are fixed regardless of the size of the training set (e.g. Linear regression)
  - Non-parametric model: the number of parameters can grow as the size of training set increases (e.g. k-NN classifier)

Learning a Model

- Consider a simple approach based on curve fitting, a degree M-polynomial function:
  - $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=1}^{M} w_j x^j$, where
    - $\mathbf{w}$ is the vector denoting the model parameters collectively (i.e. polynomial coefficients);
    - M is the order of the polynomial;
    - x^j denotes x raised to the power of j.

  - Note that $y(x, \mathbf{w})$ is a non-linear function of x, but a linear function of the coefficients $\mathbf{w}$. We call the function $y(x, \mathbf{w})$ a linear model

- Question: how to determine $\mathbf{w}$?

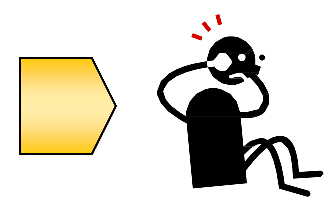
Learning a Model

❑ How to determine $\mathbf{w}$?

- o When fitting the polynomial to the training set, we need to find $\mathbf{w}$ that minimise an error function that measures the misfit between $y(x, \mathbf{w})$, for any given value of $\mathbf{w}$, and the training set data points.

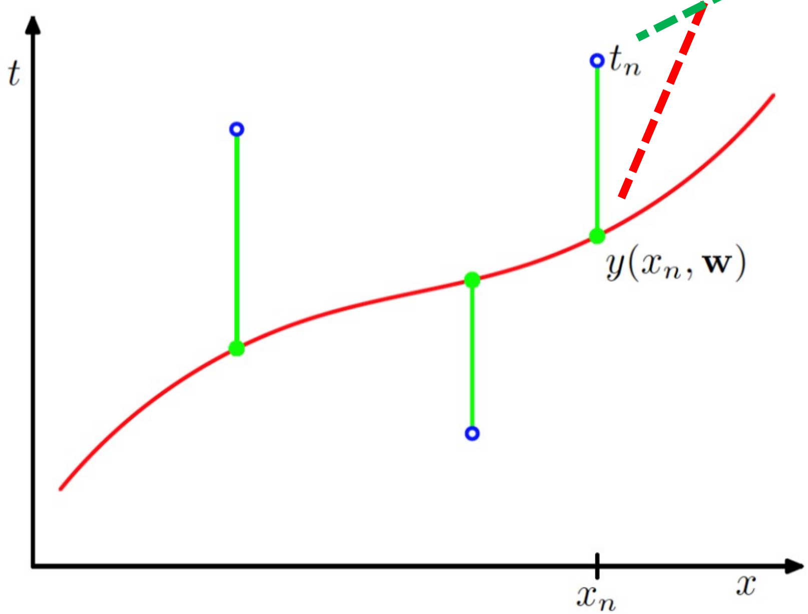
❑ How to define such an error function?

$$E(\mathbf{w}) := \frac{1}{2}\sum_{n=1}^{N}[\underbrace{y(x_n,\mathbf{w})}_{\textit{predictions}} - \underbrace{t_n}_{\substack{\textit{target}\\ \textit{values}}}]^2$$



Learning a Model

- Error Function: $E(\mathbf{w}) := \frac{1}{2}\sum_{n=1}^{N}[y(x_{n,}\mathbf{w}) - t_n)]^2$



$E(\mathbf{w})$ means "the sum of the squares of the errors" between the predictions $y(x, \mathbf{w})$ for each data point x_n and the target values t_n.

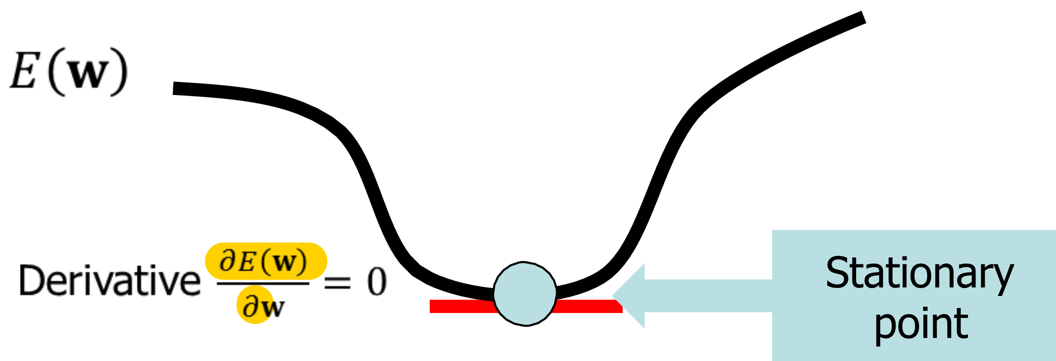
Learning a Model

- Training Objective: find $\mathbf{w}$ $(w_0, \ldots w_M)$ that minimise the error function

$$E(\mathbf{w}) := \frac{1}{2}\sum_{n=1}^{N}[y(x_n, \mathbf{w}) - t_n)]^2$$

$\mathbf{0}$? Perfectly accurate model

- Optimisation Algorithm: Learning problem is solved by choosing the value of $E(\mathbf{w}^*)$: $\mathbf{w}^* := \arg\min_{\mathbf{w}} E(\mathbf{w})$



Learning a Model

- Linear models
  - o The error function for linear models is quadratic of **w**
  - o Its derivatives with respect to **w** is linear
  - o The minimization of the error function has a unique solution
  - o Learning process much easier

Model complexity

- Consider again the polynomial function:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=1}^{M} w_j x^j$$

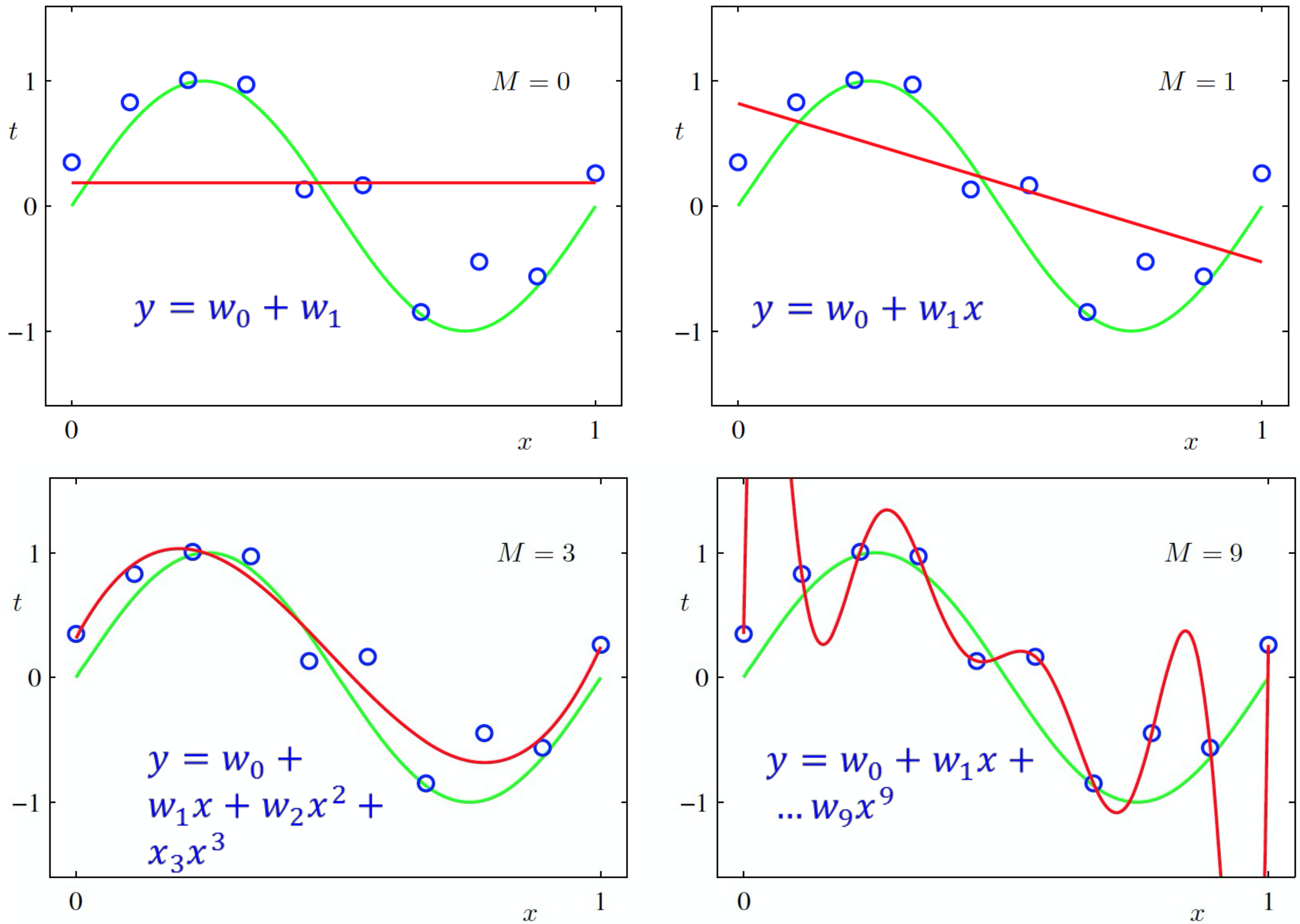
- Determines the complexity of the model

M

The higher the order, the more complex the model.

Fitting polynomial with M to the data

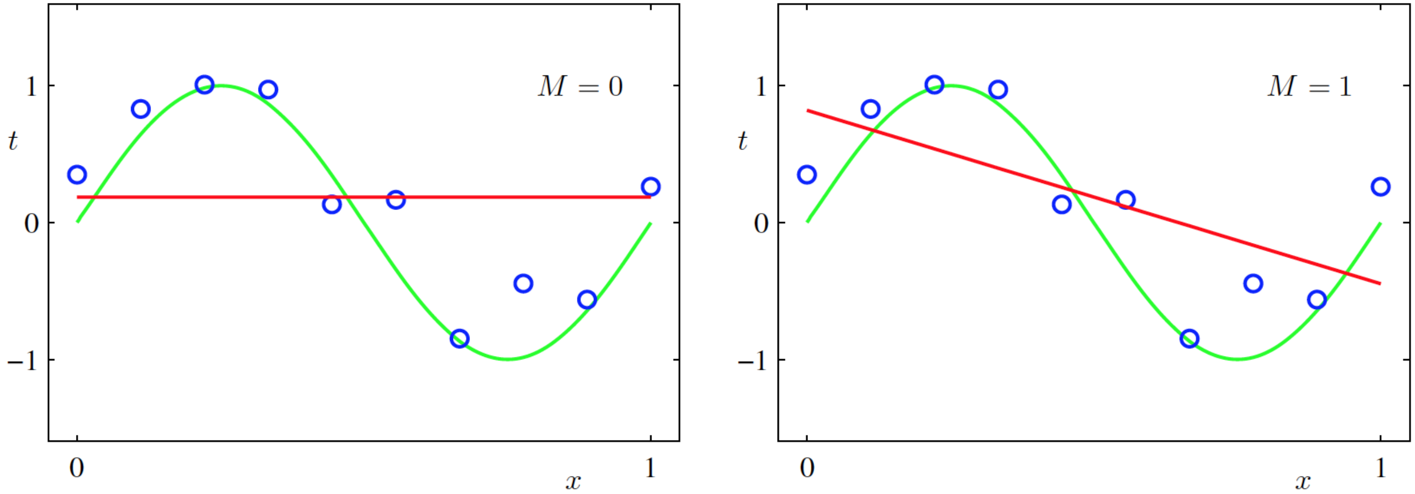
M=0, 1: poor fits to the data, thus poor representation of $\sin(2\pi x)$



M=3: well fit to $\sin(2\pi x)$

M=9: an excellent fit to the training

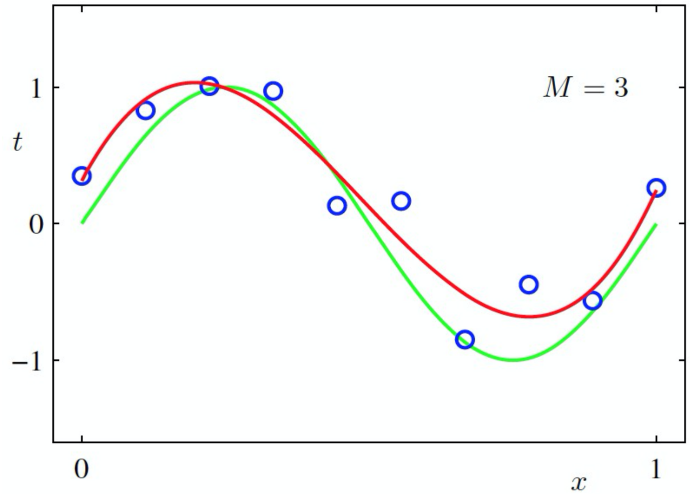
Too simple: Under-fitting

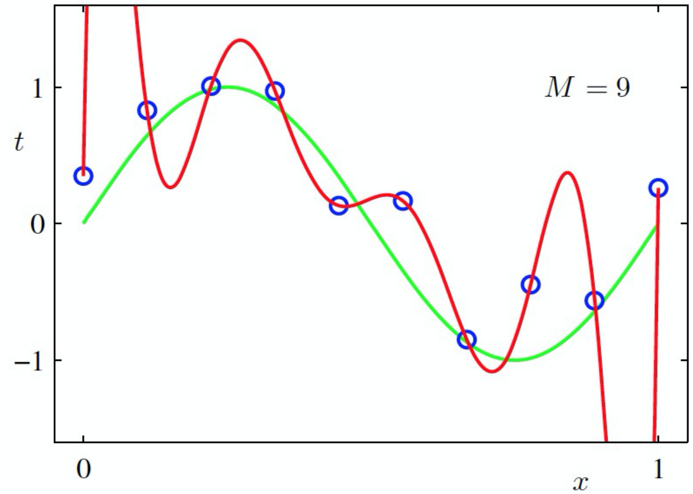


- Both model with M=0 and M=1 have **poor representation of** $\sin(2\pi x)$**: "Under-fitting"**

Too complex: Over-fitting

Which model is better?

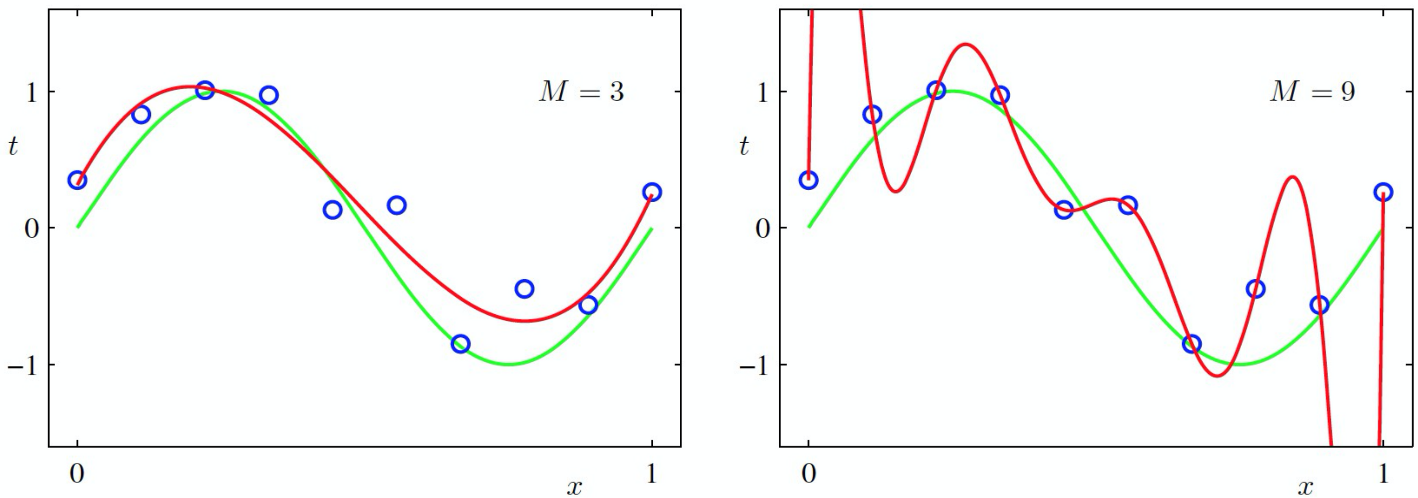




- The model with M=9 has an excellent fit to the training data but **poor representation of $\sin(2\pi x)$:** "Over-fitting"

Too complex: Over-fitting

Which model is better?



- The model with M=9 has an excellent fit to th training data but **poor representation of si**
  “Over-fitting”

Cannot tell from the training error without the real function! Then how?

Generalisation Performance

Recall we need to achieve a good generalisation.

How to measure the generalisation performance on a model with **M**?

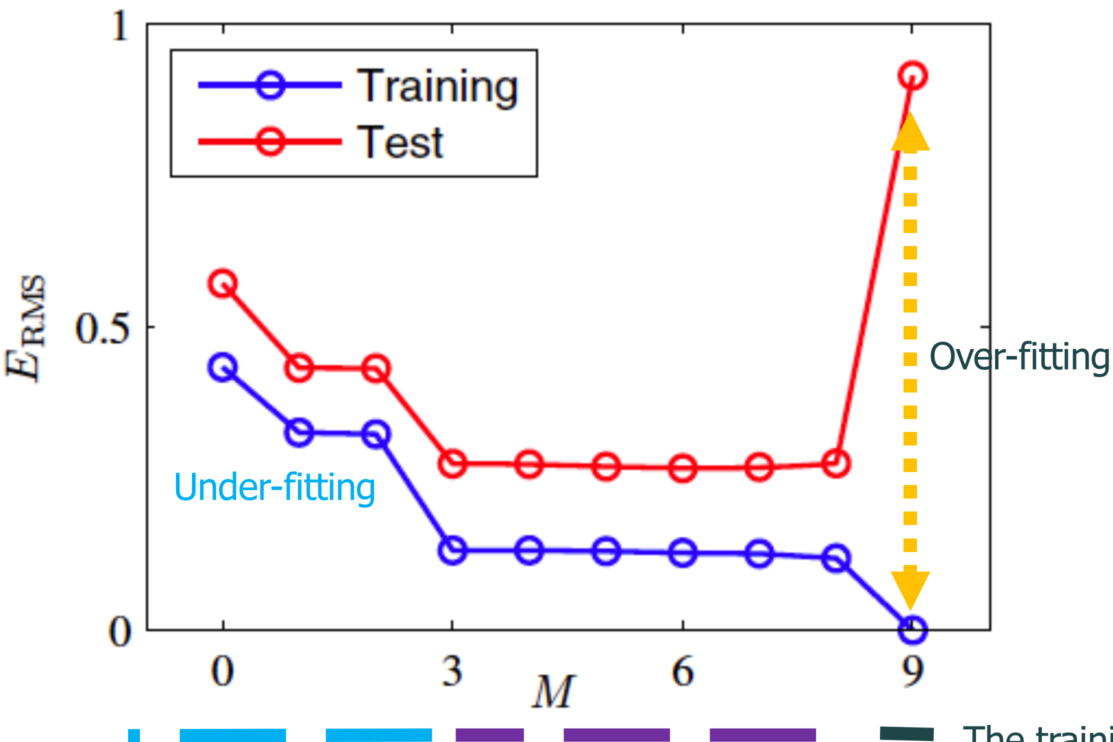
Evaluate $E(\mathrm{w}^*)$ for both training and testing set

Use the **root-mean-square (RMS)** error:

$$E_{RMS} = \sqrt{2E(\mathbf{w}^*)/N},$$

where N is the size dataset (training set and testing set).

Generalisation Performance



Large values of the error

Small values of the error

The training set error goes to 0, but the test set error becomes very large

Paradox on the model with M=9

A power series expansion (e.g. $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$) of $\mathbf{sin(2\pi x)}$ contains all lower order polynomials.

Expect that the test error should decrease gradually as the degree **M** is increased. However, the model with M=9 shows a **large over-fitting problem.**

What's wrong with the model with M=9 then?

Paradox on the model with M=9

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

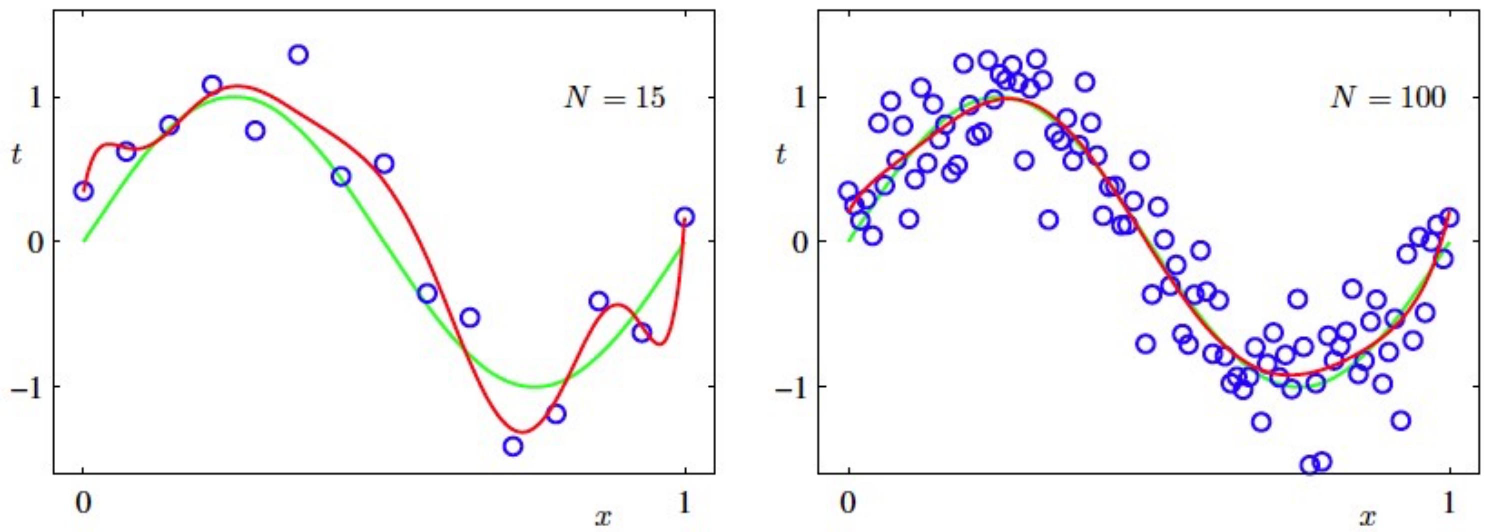
Table of the parameters for polynomials for various orders.

As M increases, the parameters become larger → The models is too flexible; and the parameters were very finely tuned to the 10 training points.

How to reduce high flexibility of the model with M = 9?

This questions is formulated as how to reduce an **over-fitting** problem?

- o **Increase the size of a training set**



Plots of the solutions obtained by minimizing the error function with M = 9 for N = 15 data points (left plot) and N = 100 data points (right plot).

Regularisation

- A technique to control the over-fitting phenomenon
  - **Idea:** Add a penalty term to the error function to discourage the parameters from reaching large values:

$$E(\mathbf{w}) := \frac{1}{2}\sum_{n=1}^{N}[y(x_{n,}\mathbf{w}) - t_n)]^2 + penalty(\mathbf{w})$$

  - **Penalty Term**: the sum of square of all parameters:

$$penalty(\mathbf{w}) := \frac{\lambda}{2}||\mathbf{w}||^2,$$

where $||\mathbf{w}||^2 = \mathrm{w}_0^2 + \mathrm{w}_1^2 + \ldots + \mathrm{w}_M^2$, and λ is the regularization parameter governing the relative importance of the penalty term.

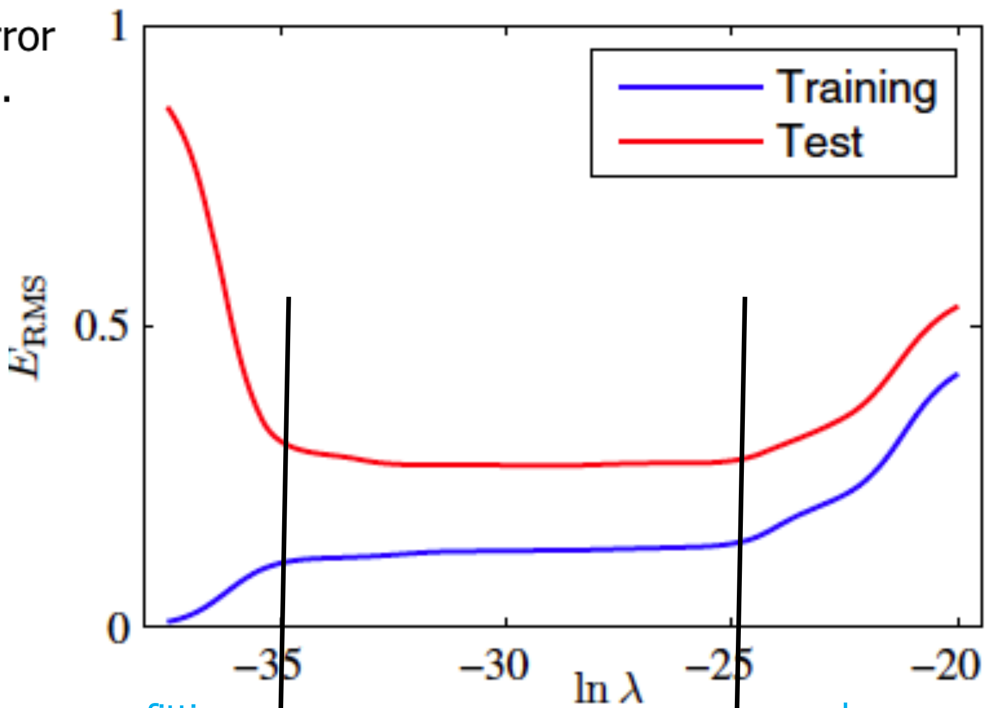
- Revise the training objective
  - tradeoff between the empirical error and the model complexity.

How does regularisation work?

For a large λ, models with high complexity can be ruled out. For a small λ, models with high training errors can be ruled out. The optimal solution lies somewhere in the middle.

Graph of the root-mean-square error vs. $\ln(\lambda)$ for the M = 9 polynomial.

1. λ increases, model becomes less complex, training error increases
2. λ is very small, model too complex, test error high (over fitting)
3. λ increases, model becomes less complex, testing error decreases
4. λ is too large, model too simple, testing error increases (under fitting)



over fitting

under fitting

Model Selection

- Revisit what we've learned from the polynomial curve fitting problem:
  - The order controls the complexity
  - The regularization parameter also controls the complexity
  - For more complex models, more parameters
  - How to train these parameters?
    - No access to the test data when we train the model, remember?
    - Cannot do this based on testing error

Model Selection

What data do I need to use to select a good model?

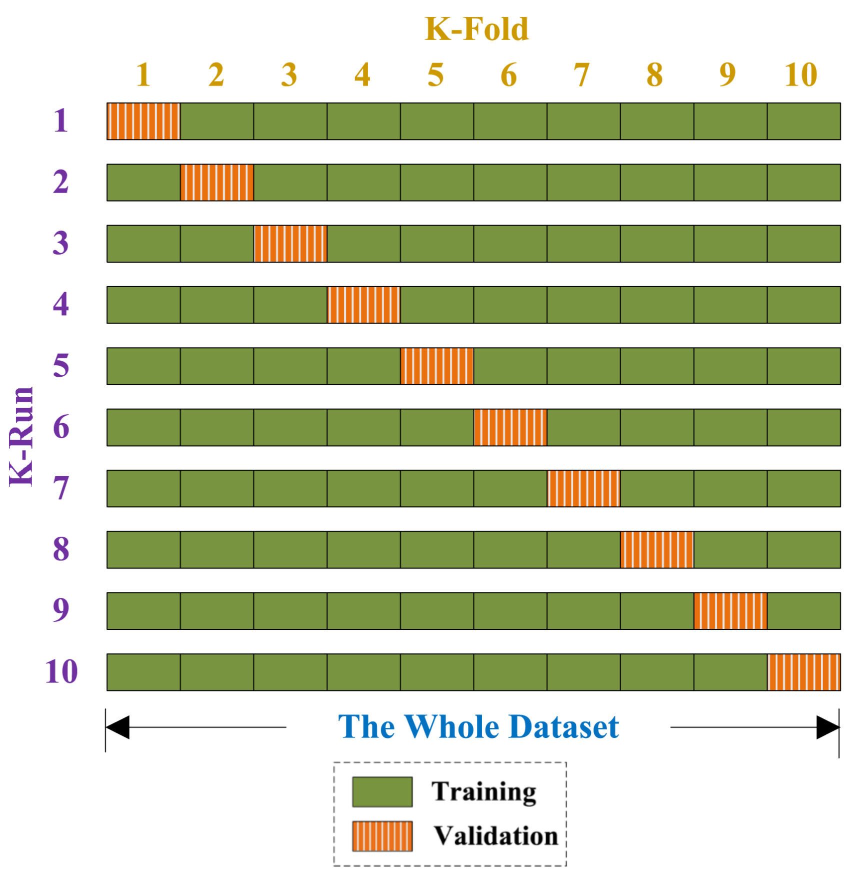
Hold-out validation technique?

Available data
•Training set (e.g. 90%)
•Test set (or called validation set)
➔ Too simple. Too wasteful of the training set

K-Fold Cross-Validation (more complicated approach)

Model Selection

❑ K-Fold Cross-Validation



- For each parameter setting:
  - Divide the available dataset into K equal-size distinct subsets
  - Each time use one of these subsets (1/K sample) as test set and the other (K−1) subsets as the training set.
  - This procedure is repeated K times to ensure all samples are used for both training (K−1 times) and validation (only once).
  - The average of the obtained validation errors is used as an estimation of the testing error.

Model Selection

❑ Leave-One-Out Cross-Validation

- A special case of K-Fold cross-validation where K (i.e., the number of folds/subsets) is equal to the size of the training dataset.
- In each iteration, one training data point is left out as the validation set.
- All the others are used to train the model.
- This procedure is repeated K times. This is to make sure that all data points are selected exactly once as in the validation phase.

Module 1: Elements of Machine Learning

- Module Objectives
  - o Provide an introduction to machine learning theory
  - o **Part A (Week 1):**
    - Understand the machine learning process
    - Understand key concepts of machine learning
    - Understand how to select a good learning model

Wrap up the lecture

- What is the machine learning process?
  - Learn a model of functional dependence between input attributes and target values from the training set, and use it to predict target values of unknown data
- What are key concepts of machine learning?
  - Need to determine the parameters of a model, if a model is parametric.
  - Need to well fit the model to the training: by minimizing an error function between predicted- and correct- target values
  - Also need to prevent overfitting
    - Many parameters to consider: the order, the regularization parameter
- How to choose a good statistical model?
  - Using a cross-validation
  - Measuring both training error and testing error

Tutorial (Week 1)

- Learn how **k-Nearest Neighbors (NN) classifier** works.
- Using k-NN, practice some of the basic concepts of machine learning in **R programming environment**.

What will we learn in Week 2

- **Part B (Week 2):**
  - Understand probabilistic machine learning
  - Understand prediction uncertainty and develop tools (**bootstrapping**) to measure it (**Tutorial**)


Thank You