# *Introduction to Statistical Machine Learning*

Cheng Soon Ong & Christian Walder

Machine Learning Research Group
Data61 | CSIRO
and
College of Engineering and Computer Science
The Australian National University

Canberra
February – June 2019

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

# Part II

## *Introduction*

# Flavour of this course

- Formalise intuitions about problems
- Use language of mathematics to express models
- Geometry, vectors, linear algebra for reasoning
- Probabilistic models to capture uncertainty
- Design and analysis of algorithms
- Numerical algorithms in python
- Understand the choices when designing machine learning methods

# Goals for today

- Use 1D regression as illustration
  (more regression next week)
- Model
- Performance
- Generalization
- Finding best parameters

# What is Machine Learning?




Ong & Walder & Webers
Data61 | CSIRO
The Australian National University



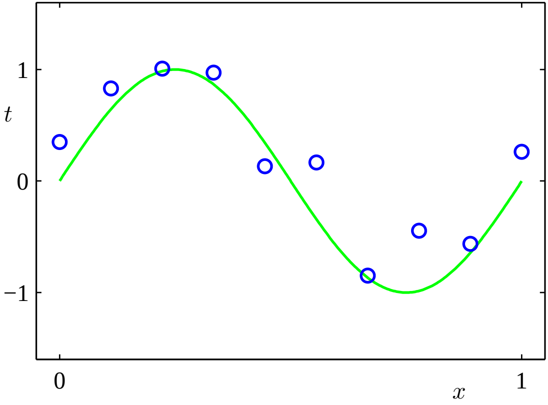
*Definition (Mitchell, 1998)*

A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$.

# Polynomial Curve Fitting

- some artificial data created from the function

$$\sin(2\pi x) + \text{random noise} \qquad x = 0, \dots, 1$$

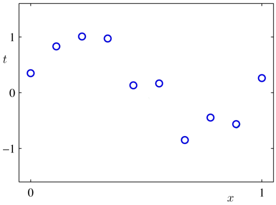


# Polynomial Curve Fitting - Input Specification

$$N = 10$$

$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$

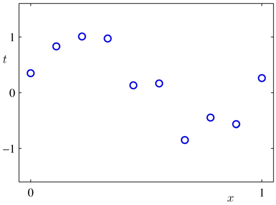
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$



# Polynomial Curve Fitting - Input Specification



$$N = 10$$
$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$
$$x_i \in \mathbb{R} \quad i = 1, \dots, N$$
$$t_i \in \mathbb{R} \quad i = 1, \dots, N$$

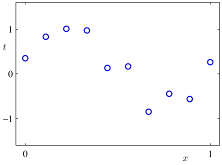


# Polynomial Curve Fitting - Model Specification





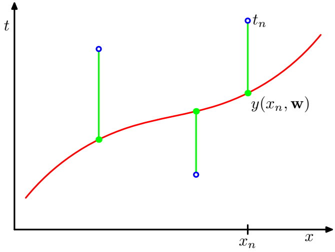
M : order of polynomial



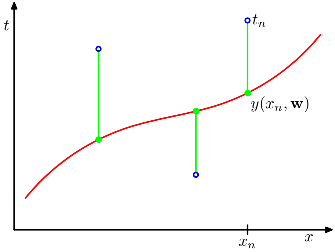

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{m=0}^{M} w_m x^m$$

- nonlinear function of $x$
- linear function of the unknown model parameter $\mathbf{w}$
- How can we find good parameters $\mathbf{w} = (w_1, \ldots, w_M)^T$?

# Learning is Improving Performance



# Learning is Improving Performance



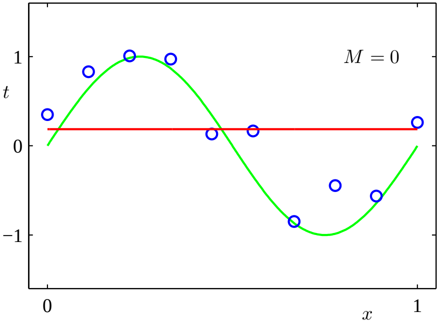

- Performance measure : Error between target and prediction of the model for the training data

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

- unique minimum of $E(\mathbf{w})$ for argument $\mathbf{w}^\star$

# Model Comparison or Model Selection

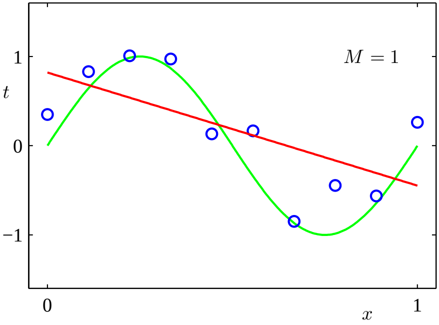
$$y(x, \mathbf{w}) = \sum_{m=0}^{M} w_m \, x^m \Bigg|_{M=0} = w_0$$



# Model Comparison or Model Selection



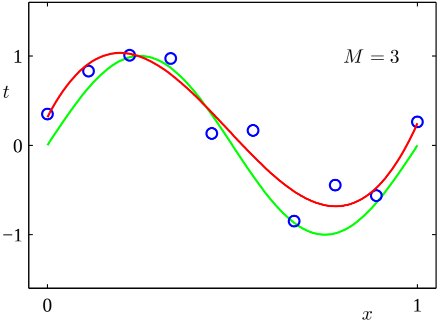
$$
\begin{aligned}
y(x, \mathbf{w}) &= \sum_{m=0}^{M} w_m \, x^m \quad \Bigg|_{M=1} \\
&= w_0 + w_1 \, x
\end{aligned}
$$



# Model Comparison or Model Selection



$$y(x, \mathbf{w}) = \sum_{m=0}^{M} w_m x^m \Bigg|_{M=3}$$
$$= w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

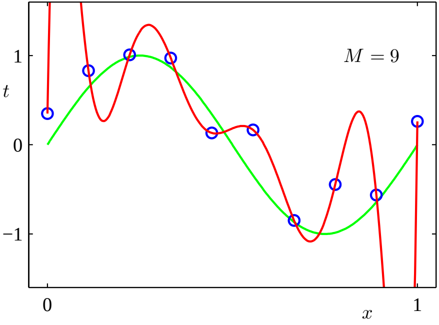


# Model Comparison or Model Selection



$$y(x, \mathbf{w}) = \sum_{m=0}^{M} w_m x^m \Bigg|_{M=9}$$
$$= w_0 + w_1 x + \cdots + w_8 x^8 + w_9 x^9$$

- overfitting

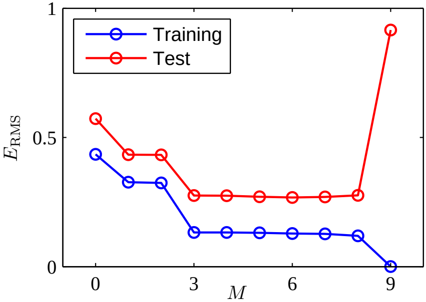


# Testing the Model



- Train the model and get $\mathbf{w}^\star$
- Get $100$ new data points
- Root-mean-square (RMS) error

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



# Testing the Model

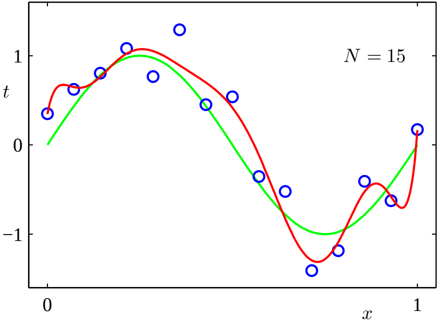
| | M = 0 | M = 1 | M = 3 | M = 9 |
|---|---|---|---|---|
| $w_0^\star$ | 0.19 | 0.82 | 0.31 | 0.35 |
| $w_1^\star$ | | -1.27 | 7.99 | 232.37 |
| $w_2^\star$ | | | -25.43 | -5321.83 |
| $w_3^\star$ | | | 17.37 | 48568.31 |
| $w_4^\star$ | | | | -231639.30 |
| $w_5^\star$ | | | | 640042.26 |
| $w_6^\star$ | | | | -1061800.52 |
| $w_7^\star$ | | | | 1042400.18 |
| $w_8^\star$ | | | | -557682.99 |
| $w_9^\star$ | | | | 125201.43 |

*Table:* Coefficients $\mathbf{w}^\star$ for polynomials of various order.

# More Data

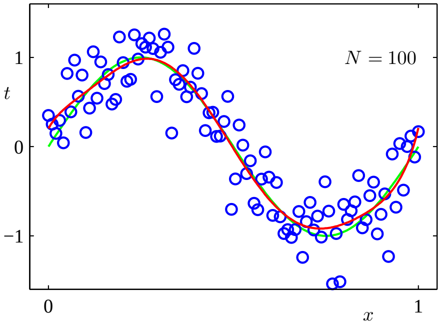


- $N = 15$



# More Data

- $N = 100$
- heuristics : have no less than $5$ to $10$ times as many data points than parameters
- but number of parameters is not necessarily the most appropriate measure of model complexity !
- later: Bayesian approach



# Regularisation







- How to constrain the growing of the coefficients $\mathbf{w}$ ?
- Add a regularisation term to the error function

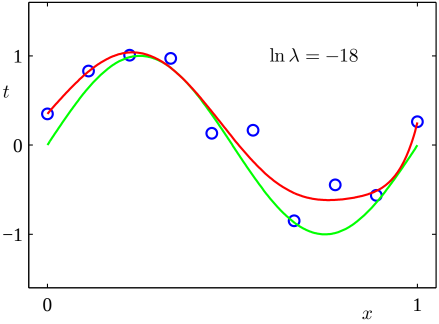
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left( y(x_n, \mathbf{w}) - t_n \right)^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- Squared norm of the parameter vector $\mathbf{w}$

$$\|\mathbf{w}\|^2 \equiv \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \cdots + w_M^2$$

# Regularisation

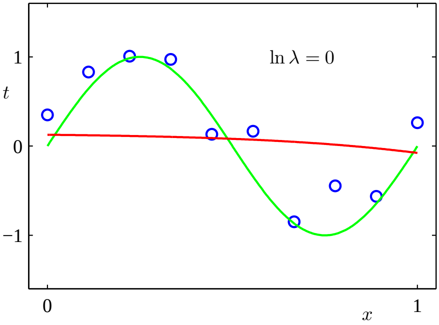
- $M = 9$



# Regularisation

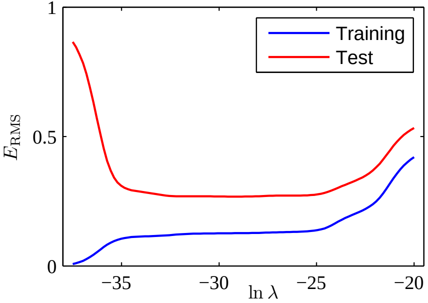


- $M = 9$



# Regularisation

- $M = 9$



# What is Machine Learning?







*Definition (Mitchell, 1998)*

A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$.

- Task: regression
- Experience: $\mathbf{x}$ input examples, $\mathbf{t}$ output labels
- Performance: squared error
- Model choice
- Regularisation
- **do not train on the test set!**

# Linear Curve Fitting - Input Specification



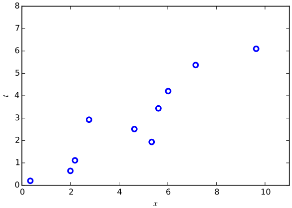
$$N = 10$$

$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$

$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$

$$x_i \in \mathbb{R} \quad i = 1, \dots, N$$

$$t_i \in \mathbb{R} \quad i = 1, \dots, N$$



# Strategy in this course







- Estimate best predictor = training = learning
  Given data $(x_1, y_1), \ldots, (x_n, y_n)$, find a predictor $f_{\mathbf{w}}(\cdot)$.
  1. Identify the type of input $x$ and output $y$ data
  2. Propose a (linear) mathematical model for $f_{\mathbf{w}}$
  3. Design an objective function or likelihood
  4. Calculate the optimal parameter ($\mathbf{w}$)
  5. ~~Model uncertainty using the Bayesian approach~~
  6. Implement and compute (the algorithm in python)
  7. ~~Interpret and diagnose results~~

# Linear Curve Fitting

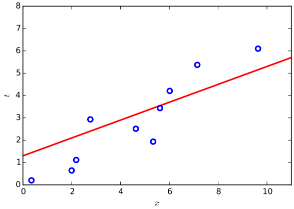
$$N = 10$$

$$\mathbf{x} \equiv (x_1, \ldots, x_N)^T$$

$$\mathbf{t} \equiv (t_1, \ldots, t_N)^T$$

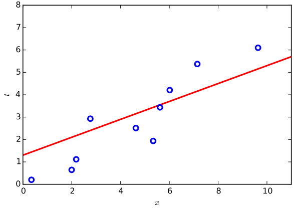
$$x_i \in \mathbb{R} \quad i = 1, \ldots, N$$

$$t_i \in \mathbb{R} \quad i = 1, \ldots, N$$



# Linear Curve Fitting - Choice of model

$$N = 10$$
$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$
$$x_i \in \mathbb{R} \quad i = 1, \dots, N$$
$$t_i \in \mathbb{R} \quad i = 1, \dots, N$$
$$y(x, \mathbf{w}) = w_1 x + w_0$$

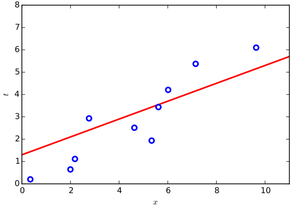


# Linear Curve Fitting - Augment for convenience





$$
\begin{aligned}
N &= 10 \\
\mathbf{x} &\equiv (x_1, \dots, x_N)^T \\
\mathbf{t} &\equiv (t_1, \dots, t_N)^T \\
x_i &\in \mathbb{R} \quad i = 1, \dots, N \\
t_i &\in \mathbb{R} \quad i = 1, \dots, N \\
y(x, \mathbf{w}) &= w_1 x + w_0 \\
X &\equiv [\mathbf{x} \quad 1]
\end{aligned}
$$



# Linear Algebra concepts



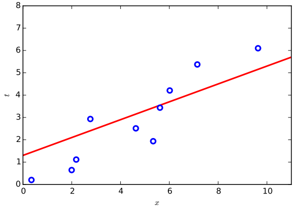




- matrix, vector, multiplication
- inner product
- projection
- rank
- inverse

# Linear Curve Fitting - Project onto plane

$$N = 10$$
$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$
$$x_i \in \mathbb{R} \quad i = 1,\dots, N$$
$$t_i \in \mathbb{R} \quad i = 1,\dots, N$$
$$y(x, \mathbf{w}) = w_1 x + w_0$$
$$X \equiv [\mathbf{x} \quad 1]$$




Find the best plane that will fit the data.

# *Orthogonal projection*

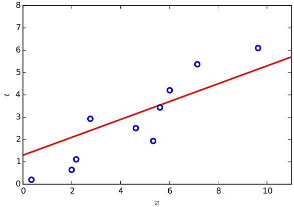
- Assume we have data points $(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)$
- Want to solve

$$Xw = \mathbf{t}$$

- If points don't fall perfectly on the line, cannot be solved
- Find a point $\hat{\mathbf{t}}$ that lies in the column space of $X$, and is closest to $y$.
- $\hat{\mathbf{t}}$ is found by the orthogonal projection of $\mathbf{t}$ onto the column space of $X$.

# Linear Curve Fitting - Least Squares

$$N = 10$$
$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$
$$x_i \in \mathbb{R} \quad i = 1, \dots, N$$
$$t_i \in \mathbb{R} \quad i = 1, \dots, N$$
$$y(x, \mathbf{w}) = w_1 x + w_0$$
$$X \equiv [\mathbf{x} \quad 1]$$
$$w^* = (X^T X)^{-1} X^T \mathbf{t}$$



# Vector calculus concepts

- Differentiation
- Partial differentiation
- Differentiation of vector valued functions $f : \mathbb{R}^n \to \mathbb{R}^m$
- Product rule, Quotient rule, Sum rule, Chain rule

# Solving with Least Squares loss



- Assume we have data points $(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)$
- Define a loss function

$$\|\mathbf{t} - \hat{\mathbf{t}}\|^2$$

  where $\hat{\mathbf{t}} = X\mathbf{w}$.
- Take the gradient

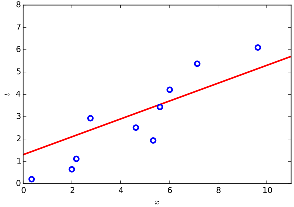
$$\frac{dl}{d\mathbf{w}} = 2(\mathbf{t} - X\mathbf{w})^\top X.$$

- Solve for stationary point

$$X^\top X\mathbf{w} = X^\top \mathbf{t}$$

# Linear Curve Fitting - Least Squares

$$N = 10$$
$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$
$$x_i \in \mathbb{R} \quad i = 1,\dots, N$$
$$t_i \in \mathbb{R} \quad i = 1,\dots, N$$
$$y(x, \mathbf{w}) = w_1 x + w_0$$
$$X \equiv [\mathbf{x} \quad 1]$$
$$w^* = (X^T X)^{-1} X^T \mathbf{t}$$




How do we choose a noise model?

# Sum Rule and Product Rule



- Sum Rule

$$p(X) = \sum_{Y} p(X, Y)$$

- Product Rule

$$p(X, Y) = p(X \mid Y)\, p(Y)$$

# Bayes Theorem

Use product rule

$$p(X, Y) = p(X \mid Y)\, p(Y) = p(Y \mid X)\, p(X)$$

Bayes Theorem

$$p(Y \mid X) = \frac{p(X \mid Y)\, p(Y)}{p(X)}$$

only defined for $p(X) > 0$

and

$$\begin{aligned} p(X) &= \sum_{Y} p(X, Y) && \text{(sum rule)} \\ &= \sum_{Y} p(X \mid Y)\, p(Y) && \text{(product rule)} \end{aligned}$$

# Expectations



- Weighted average of a function f(x) under the probability distribution $p(x)$

$$\mathbb{E}\left[f\right] = \sum_x p(x) f(x) \qquad \text{discrete distribution } p(x)$$

$$\mathbb{E}\left[f\right] = \int p(x) f(x) \, \mathrm{d}x \qquad \text{probability density } p(x)$$

# The Gaussian Distribution

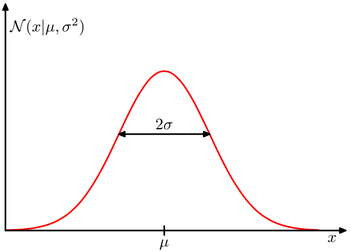






- $x \in \mathbb{R}$
- Gaussian Distribution with mean $\mu$ and variance $\sigma^2$

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\{-\frac{1}{2\sigma^2}(x-\mu)^2\}$$



# The Gaussian Distribution

- $\mathcal{N}(x \mid \mu, \sigma^2) > 0$
- $\int_{-\infty}^{\infty} \mathcal{N}(x \mid \mu, \sigma^2)\, \mathrm{d}x = 1$
- Expectation over $x$

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}(x \mid \mu, \sigma^2)\, x \; \mathrm{d}x = \mu$$

- Expectation over $x^2$

$$\mathbb{E}\left[x^2\right] = \int_{-\infty}^{\infty} \mathcal{N}(x \mid \mu, \sigma^2)\, x^2 \; \mathrm{d}x = \mu^2 + \sigma^2$$

- Variance of x

$$\mathrm{var}[x] = \mathbb{E}\left[x^2\right] - \mathbb{E}[x]^2 = \sigma^2$$

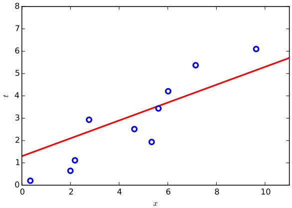
# Linear Curve Fitting - Least Squares







$$N = 10$$
$$\mathbf{x} \equiv (x_1, \dots, x_N)^T$$
$$\mathbf{t} \equiv (t_1, \dots, t_N)^T$$
$$x_i \in \mathbb{R} \quad i = 1, \dots, N$$
$$t_i \in \mathbb{R} \quad i = 1, \dots, N$$
$$y(x, \mathbf{w}) = w_1 x + w_0$$
$$X \equiv [\mathbf{x} \quad 1]$$
$$w^* = (X^T X)^{-1} X^T \mathbf{t}$$




We assume

$$t = \underbrace{y(\mathbf{x}, \mathbf{w})}_{\text{deterministic}} + \underbrace{\epsilon}_{\text{Gaussian noise}}$$

# Strategy in this course



- Estimate best predictor = training = learning
  Given data $(x_1, y_1), \ldots, (x_n, y_n)$, find a predictor $f_{\mathbf{w}}(\cdot)$.
  1. Identify the type of input $x$ and output $y$ data
  2. Propose a (linear) mathematical model for $f_{\mathbf{w}}$
  3. Design an objective function or likelihood
  4. Calculate the optimal parameter ($\mathbf{w}$)
  5. Model uncertainty using the Bayesian approach
  6. Implement and compute (the algorithm in python)
  7. Interpret and diagnose results

# Topics to Review

- Linear Algebra
- Analytic Geometry
- Matrix Decomposition
- Vector Calculus
- Probability and Statistics
- Continuous Optimization

https://mml-book.com