Machine Learning

Lecture 1.2: Supervised Learning

Sami S. Brandt

Department of Computer Science
IT University of Copenhagen

29 August 2019

IT UNIVERSITY OF COPENHAGEN

Intended Learning Outcome of this Lecture

- **Recognise** a supervised learning problem and **select** a principle for its solution
- **Explain** the fundamentals of classification problem from training examples
- **Explain** the concept of VC dimension
- **Solve** simple least squares regression problems
- **Outline** the principles applied in model selection
- **Apply** validation set for selecting model complexity
- **Summarise** the steps in supervised learning algorithm

Outline of lecture

Classification

Regression

Model Complexity

Fundamentals of a Supervised Machine Learning Algorithm

Learning a Class from Examples

- Class C of a "family car"
  - **Prediction:** is car x a family car?
  - **Knowledge extraction:** What do people expect from a family car?

- Training data:
  Positive (+) and negative (-) **examples**

- Features:
  x_1: price, x_2: engine power.

Learning a Class from Examples

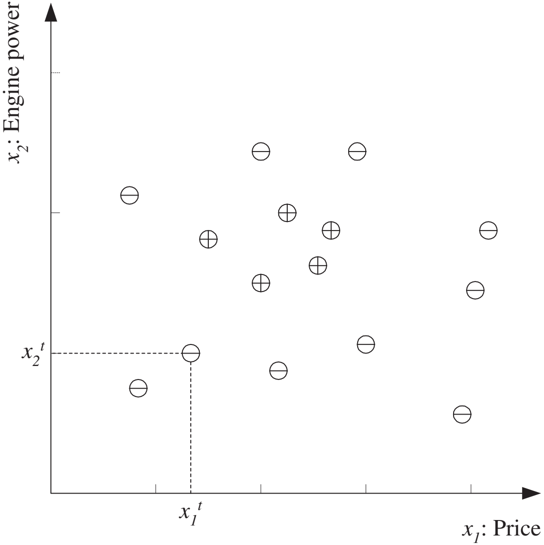
- Class C of a "family car"
  - **Prediction:** is car x a family car?
  - **Knowledge extraction:** What do people expect from a family car?

- Training data:
  Positive (+) and negative (-) **examples**

- Features:
  x_1: price, x_2: engine power.

- The trained classifier takes the features as the **input** and returns the class label as the **output**.

Training Set



$$\mathcal{X} = \{\mathbf{x}^n, r^n\}_{t=1}^{N} \text{ where}$$

$$r = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is a positive example} \\ 0 \text{ if } \mathbf{x} \text{ is a negative example.} \end{cases}$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Hypothesis Class

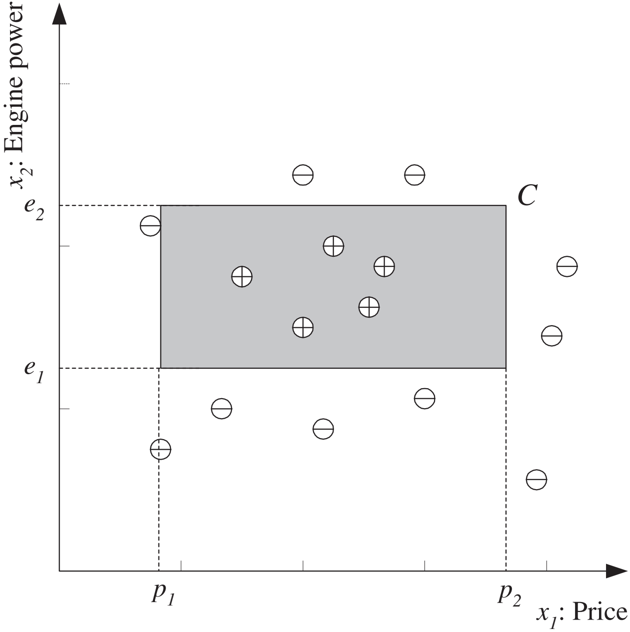
- Let us assume that there is a reason to believe the family car is defined by certain (unknown) closed intervals in the feature space

$$p_1 \leq \text{Price} \leq p_2 \quad \wedge \quad e_1 \leq \text{Engine Power} \leq e_2$$

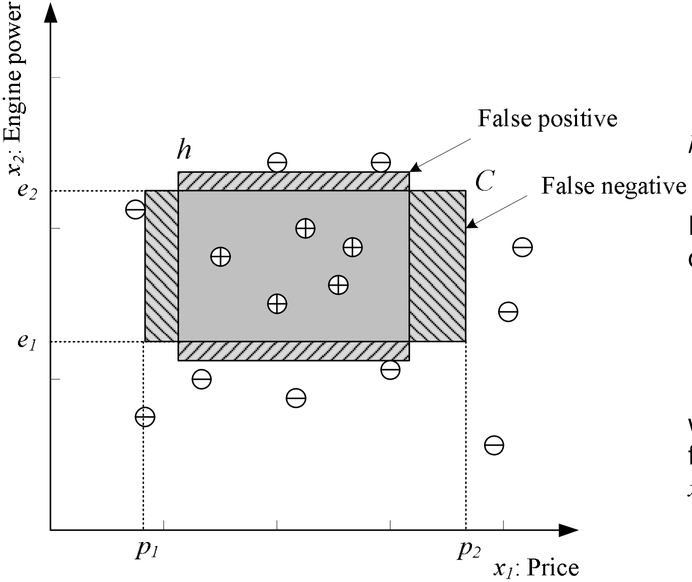
- Let $\mathcal{H}$ be the set of all hypotheses, here, the set of all rectangles in the feature space.

- $\mathcal{H}$ is referred to as the **hypothesis class**

True Class C



A Hypothesis vs. True Class



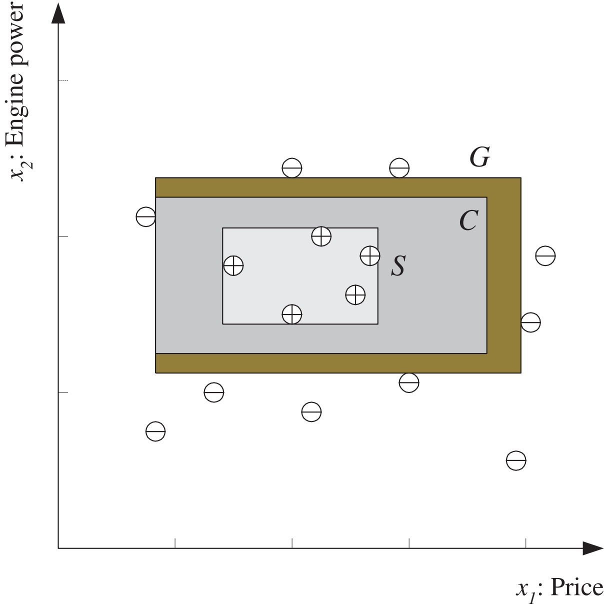
$$h(\mathbf{x}) = \begin{cases} 1 \text{ if } h \text{ suggests } \mathbf{x} \text{ is positive} \\ 0 \text{ if } h \text{ suggests } \mathbf{x} \text{ is negative.} \end{cases}$$

Error of h on the hypothesis class $\mathcal{H}$ yields

$$E(h|\mathcal{X}) = \sum_{n=1}^{N} I(h(\mathbf{x}^n) - r^n)$$

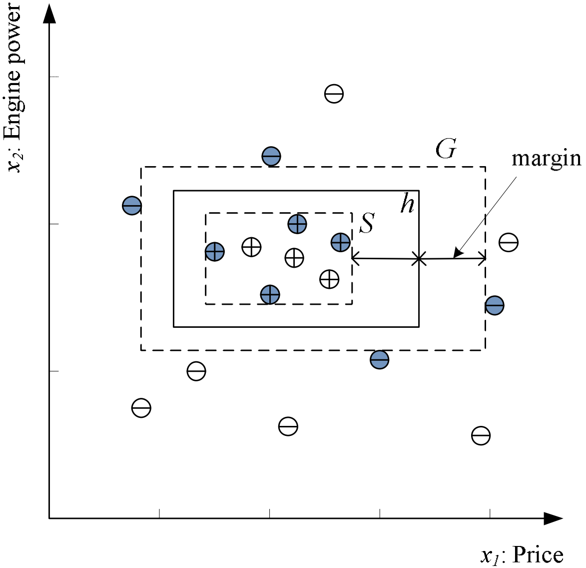
where I is the indicator function, for which $I(0) = 0$ and $I(x) = 1$, $x \neq 0$.

Most Specific (S) vs. Most General (G) Hypothesis

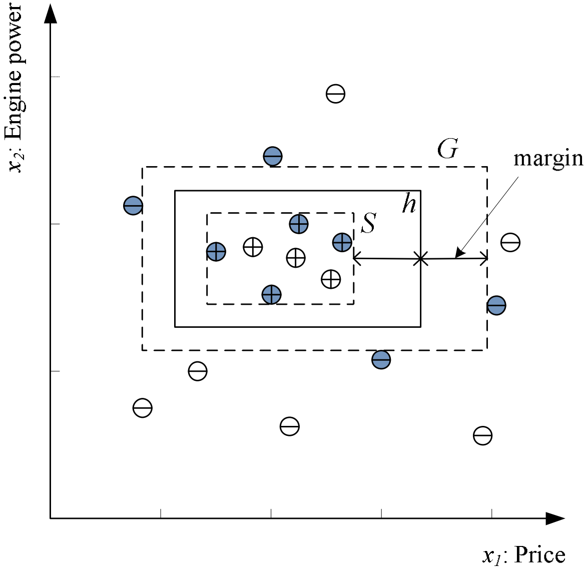


- Any hypothesis between S and G yields zero error, hence, is **consistent** with the training set
- They form the **version space**.

Hypothesis with the Largest Margin



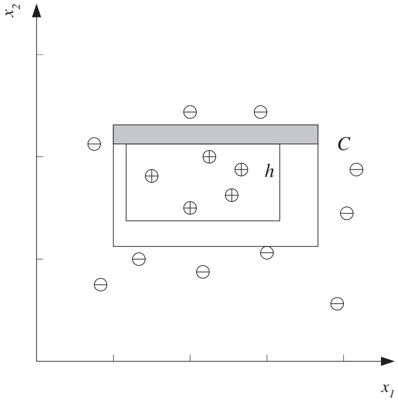
Hypothesis with the Largest Margin



We intend to minimise the error *and* maximise the margin

PAC Learning

*For how many training examples N should we have so that, **with probability at least** $1 - \delta$, the hypothesis h has the **error at most** ϵ? (Blumer et al., 1989)*



What is noise?

What is noise?

- Imprecision in recording, shifting the values

What is noise?

- Imprecision in recording, shifting the values
- Errors in labelling

What is noise?

- Imprecision in recording, shifting the values
- Errors in labelling
- Imperfections in the modelling, e.g., missing attributes

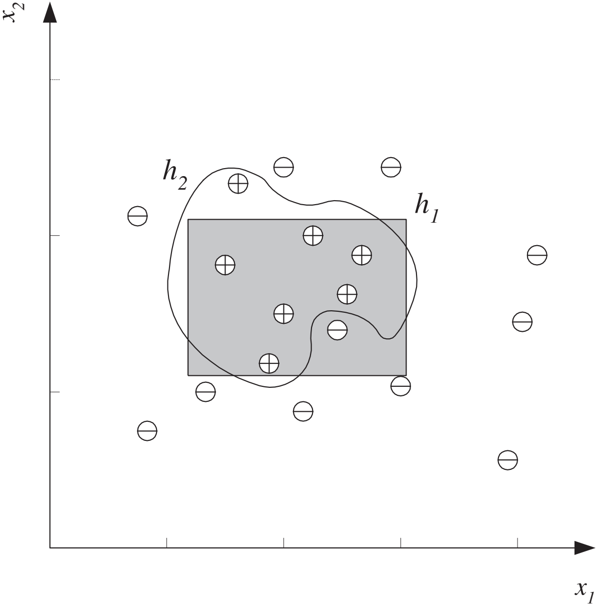
What is noise?

- Imprec
- Errors
- Imperf

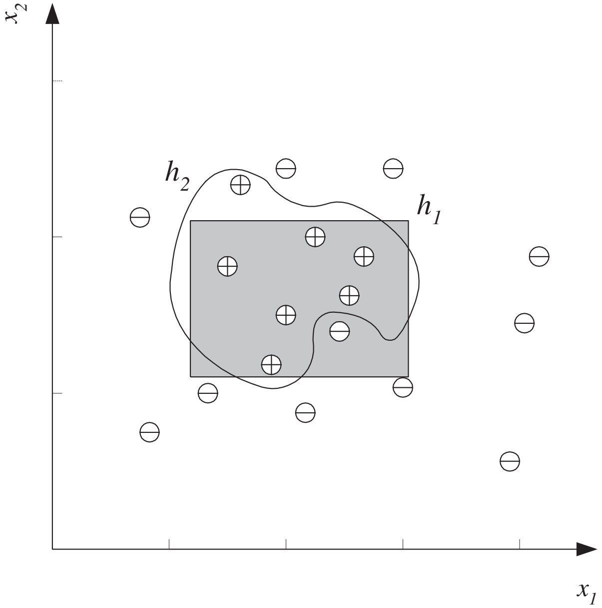
Noise removal is an ill-posed problem!

- In general, it is impossible to, without an uncertainty, to separate noise from data.
- It is an interesting philosophical question, what the noise is
  - An information theoretic definition for noise is that it the part of the signal that does is **not compressible**, i.e., it is maximally **random**

Noise: The Realisations are Distorted From the True Class

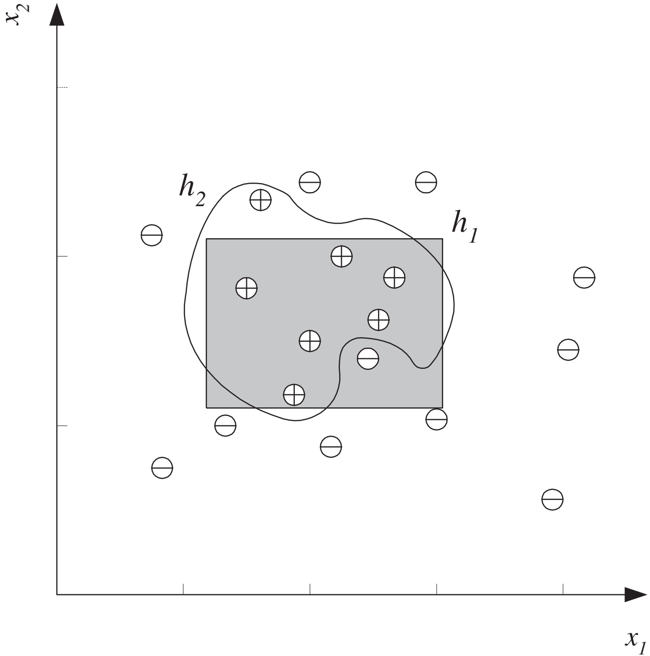


Simple Hypothesis is Favourable (why)?



Simple Hypothesis is Favourable

- Lower Computational Complexity
- Easier to Train
- Easier to Explain
- Better **generalisation** – **Occam's razor**.

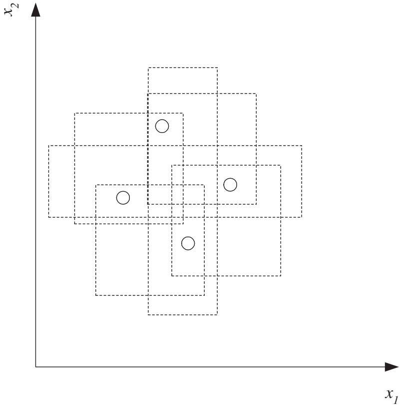


How to Characterise the Complexity of the Hypothesis Class: Vapnik–Chervonenkis Dimension

- N points can be labelled in 2^N ways to two classes (+/-)

Shattering of Points

We say, $\mathcal{H}$ **shatters** the N points iff for any fixed classification of the points, there is a hypothesis h in $\mathcal{H}$ that separates the positive examples from the negative.



- This property is used to define

VC Dimension

The maximum number of points that can be shattered by $\mathcal{H}$ is called the **Vapnik–Chervonenkis dimension** or **VC dimension** of $\mathcal{H}$.

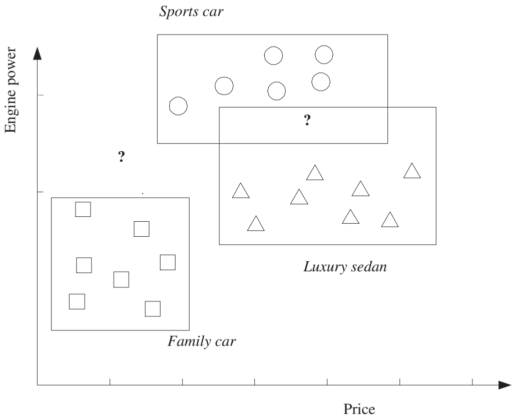
- Finding the Hypothesis class (VC dimension matched with the data), is the **model selection** problem

VC Dimension

Example

What is the VC Dimension a half plane in two-dimensions, i.e., when the class boundary is determined by a line? How about the VC dimension of a half space in N-dimensions, i.e, where the class boundary is a hyperplane?

Multiple Classes, C_k, $k = 1, 2, \ldots, K$



- The hypothesis set

$$\mathcal{X} = \{\mathbf{x}^n, r^n\}_{n=1}^N$$

- Ground truth, extended to the multiclass setup

$$r_k^n = \begin{cases} 1 \text{ if } \mathbf{x}^n \text{ belongs to the class } C_k, \\ 0 \text{ otherwise.} \end{cases}$$

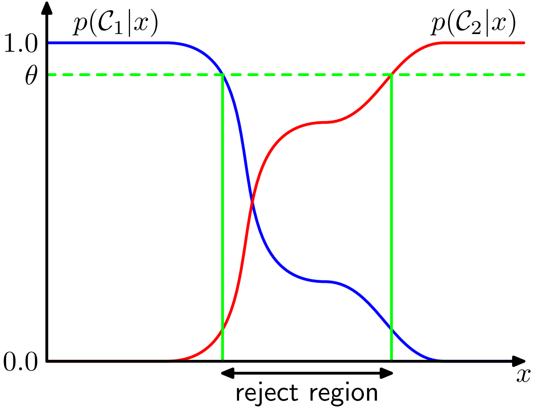
- The hypotheses $h_k(\mathbf{x})$ should be learnt so that they match with r_k^n, $k = 1, \ldots, K$.
- Achieved by minimising the **total empirical error**

$$E_{\text{emp}} = \sum_{n,k} I(h_k(\mathbf{x}^n) - r_k^n)$$

where I is the **indicator function** for which $I(0) = 0$ and $I(x) = 1$, $x \neq 0$.

The reject option

The reject option: In some applications, it may be appropriate to **avoid making a decision**, if we are **too uncertain** or have **doubt**.



Outline of lecture

Classification

Regression

Model Complexity

Fundamentals of a Supervised Machine Learning Algorithm

Regression

Assume we have the training dataset $\mathcal{X} = \{\mathbf{x}_n, r_n\}_{n=1}^{N}$.

Regression Problem

Find the function $r = g(\mathbf{x}, \mathbf{w})$ where g is our model depending on the parameters $\mathbf{w}$.
Assume that the data is noisy, i.e., $r_n = g(\mathbf{x}_n, \mathbf{w}) + \epsilon_n$, where ϵ_n represents noise.

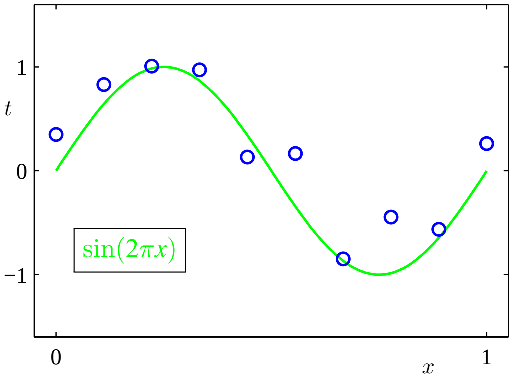
Solution: Let us form the error functional

$$E(\mathbf{w}) = \frac{1}{N} \sum (r_n - g(\mathbf{x}_n, \mathbf{w}))^2$$

and minimise it over the parameters $\mathbf{w}$.

Example: Polynomial Regression

Assume we are given a dataset $\mathcal{D} = \{(x_1, t_1), \ldots, (x_N, t_N)\}$ where $x_n, t_n \in \mathbb{R}$ and $N = 10$.



We want to fit the data using a polynomial function

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

Figure(s) from Bishop.

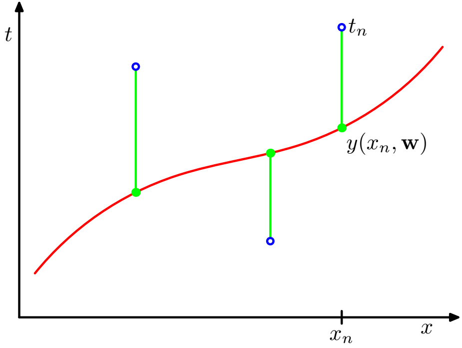
Error function

Example: Polynomial Curve Fitting

We find $\mathbf{w}$ by minimising an **error function** that **measures the misfit** between $y(x, \mathbf{w})$ and $\mathcal{D}$.

The **sum-of-squares** error function is given by

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



The **optimal solution** $\mathbf{w}^* = \arg\min_{\mathbf{w}} E(\mathbf{w})$ can be found in closed form.

Figure(s) from Bishop.

Outline of lecture

Classification

Regression

Model Complexity

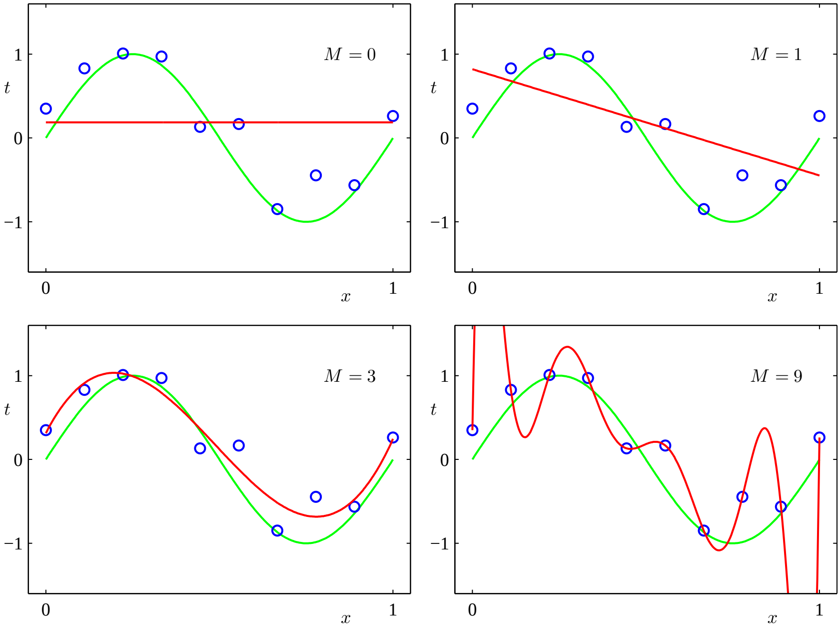
Fundamentals of a Supervised Machine Learning Algorithm

Model Selection and Generalisation

- Learning is an **ill-posed problem**; data is not sufficient to find a unique solution
- To make learning possible, one needs to make further assumptions about H, we say, there is an **inductive bias**
- **Generalisation**: How well a model performs on new data
- **Overfitting**: $\mathcal{H}$ too flexible
- **Underfitting**: $\mathcal{H}$ too rigid

Model selection: how to choose the order M?

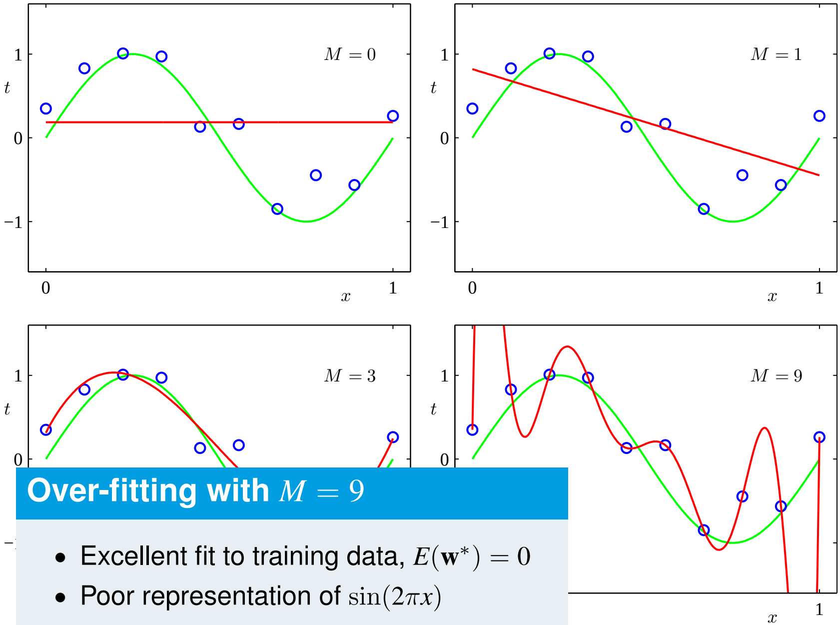
Example: Polynomial Curve Fitting



Figure(s) from Bishop.

Model selection: how to choose the order M?

Example: Polynomial Curve Fitting



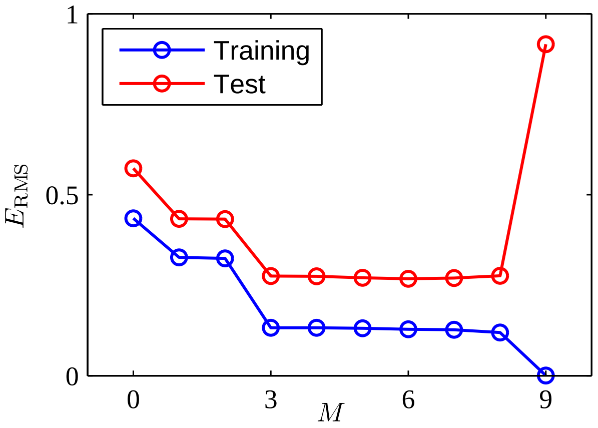
Over-fitting with $M = 9$

- Excellent fit to training data, $E(\mathbf{w}^*) = 0$
- Poor representation of $\sin(2\pi x)$

Figure(s) from Bishop.

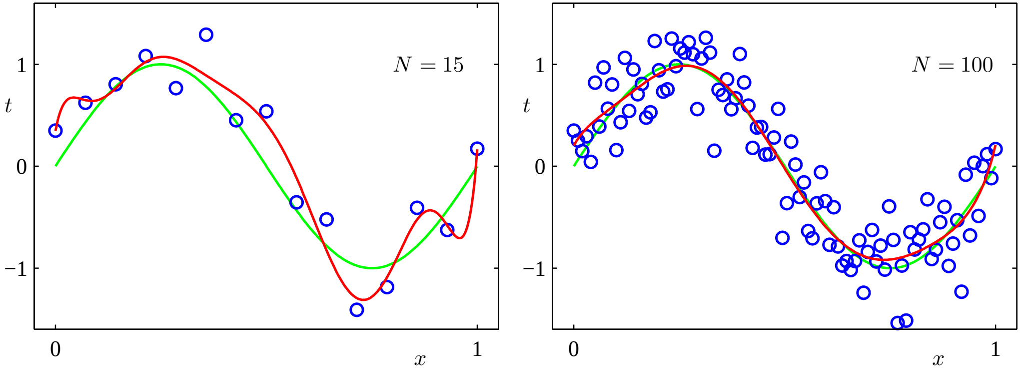
Over-fitting

Example: Polynomial Curve Fitting



Root-mean-square (RMS) error is on the same scale as t: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$

How does it behave when the size of the dataset is varied?



Polynomial of order $M = 9$

Triple Trade-Off

- There is a trade-off between three factors (Dietterich, 2003)
    1. **Complexity of $\mathcal{H}$**
    2. **Training set size**
    3. **Generalisation error E**

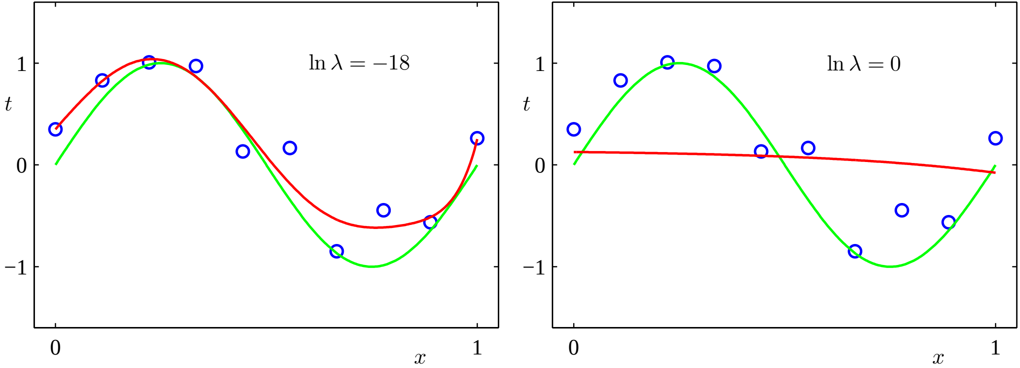
- When training set size grows error goes down.

- When complexity of $\mathcal{H}$ grows, the error first goes down, then up.

A Way to Control the Complexity: Regularisation

Over-fitting can be controlled using **regularisation**: add a term to the error function that penalises large values of the weights:

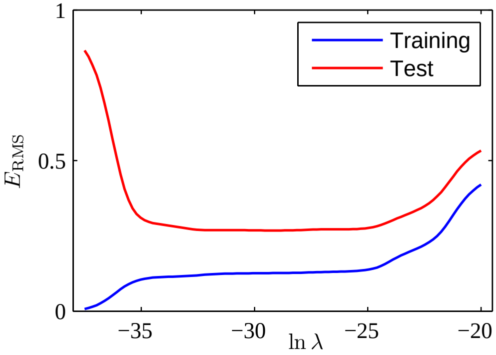
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n,\mathbf{w}) - t_n)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2.$$



Polynomial of order $M = 9$

Figure(s) from Bishop.

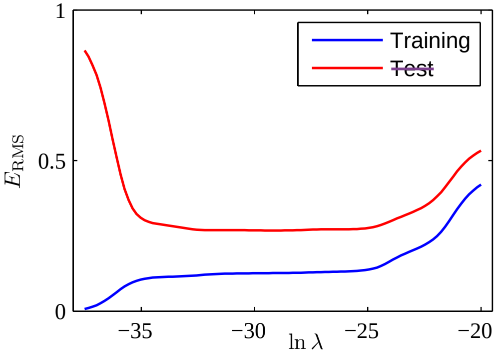
Regularisation and model complexity



A palatial approach to selection: partition the data into

- **a training set** used to learn the coefficients $\mathbf{w}$
- a separate **validation set** used to optimise the model complexity (M or λ)

Regularisation and model complexity



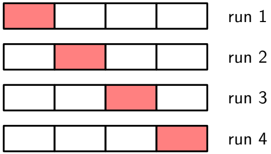
Validation

A palatial approach to selection: partition the data into

- **a training set** used to learn the coefficients $\mathbf{w}$
- a separate **validation set** used to optimise the model complexity (M or λ)

Cross-Validation

- How do we do modelling if the data set is small?
  - **Performance on the training data** is not a good indicator due **over-fitting**.
  - But we can **compare models a validation set**.
  - This suggest to use **cross-validation**: To estimate generalisation error, we additionally need data unseen during training.

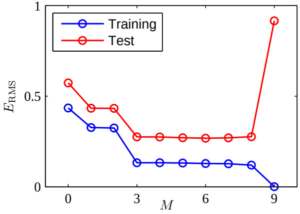


- We could split the data as
  - **Training set** (50%)
  - **Validation set** (25%)
  - **Test set** (25%)

- We can also use **folding** and **resampling** when there is limited data—more in Chapter 19.

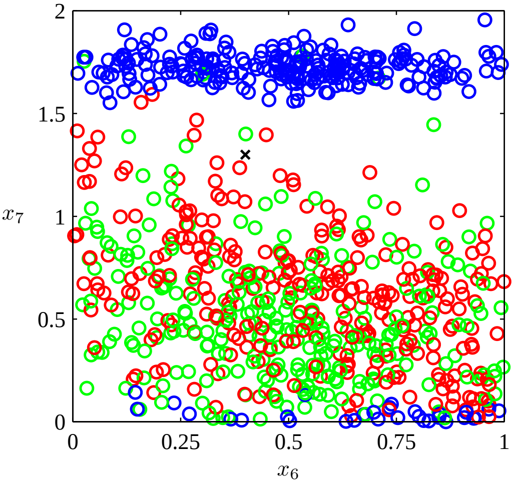
Other Approaches for Model Selection

- However, cross-validation can be **computationally expensive** if the model has **multiple parameters**.
- Various **information criteria** have been proposed based on the energy functional defined by the model likelihood.
- In principle, the energy functional is supplemented with a model complexity term that punishes more complex models
- Examples: **Akaike information criterion**, **Bayesian information criterion**, **Structural Risk Minimisation**, **MDL principle**...

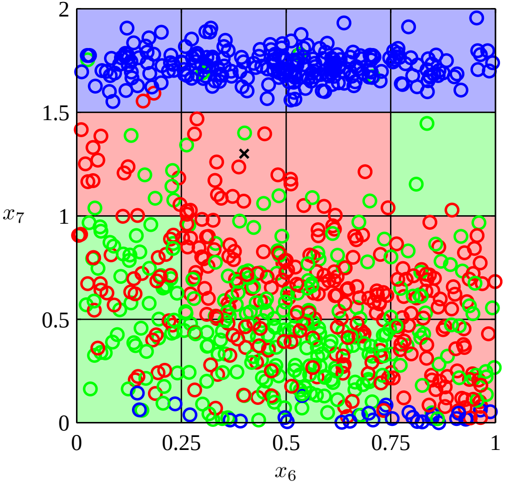


The curse of dimensionality

Consider the following **classification problem**, where we want to predict the class of 'x':



A simple solution: divided the input space into cells and assign the class of the majority to the cell:

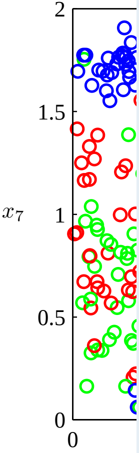


Figure(s) from Bishop.

The curse of dimensionality

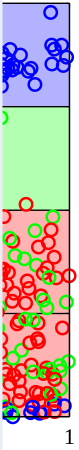
Consider the following **classification problem**, wh

of 'x':



A simple solution: divided the input space

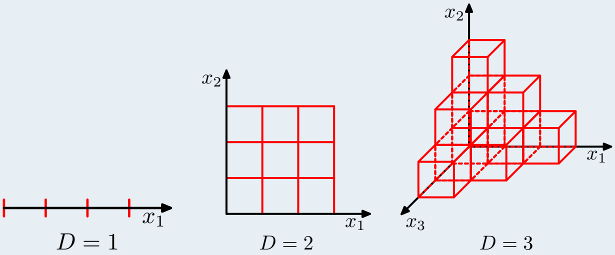
majority



What happens in higher dimensions?

If we divide each dimension into k boxes, then number of boxes are k^d.

As the number of boxes grows **exponentially** we would need we would need an **exponentially large quantity of training data** to ensure no cells are empty.



These types of problems that can arise in spaces of many dimensions is often called the *curse of dimensionality*

Outline of lecture

Classification

Regression

Model Complexity

Fundamentals of a Supervised Machine Learning Algorithm

Supervised Machine Learning Algorithm

Fundamental Parts of a Supervised Machine Learning Algorithm

1. **Model** $g(x, \theta)$, where x is the input and θ are the parameters.
2. **Loss function** $L(r^n, g(x^n, \theta))$, quantifying the difference between the desired and modelled class label, and the (penalised) **approximation error functional** $E(\theta) = \sum_n L(r^n, g(x^n, \theta)) \quad (+\text{possible penalty on complexity})$
3. **Optimisation procedure**

$$\theta^* = \arg\min_{\theta} E(\theta)$$

Next lecture

- Bayesian Decision Theory

References I

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.