CMP-6002B - Machine Learning

Gavin Cawley

Lecture 2 - Basic Principles

Introduction

- In the previous lecture:
  - Introduction to machine learning.
  - Types of machine learning.
  - Unit information.

- In the this lecture:
  - Example of statistical pattern recognition.
  - Parametric and non-parametric modelling.
  - Machine learning and optimisation.
  - Maximum likelihood.
  - Value of probabilistic modelling.
  - Generalisation and over-fitting.

Example Classification Problem

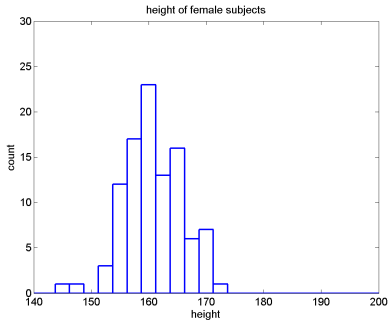
- Consider the problem of determining whether someone is male or female by measuring their height.
- Some things are intuitively obvious:
  - On average men are taller than women.
  - There is considerable variability in men's and women's heights.
  - There is an overlap in the heights of men and women.

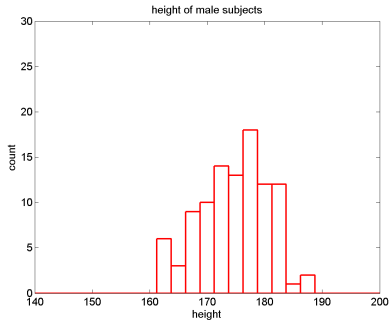
    "Now I can wear heels" [Nicole Kidman, commenting on her break-up with Tom Cruise - August 2001]

- Task: Identify a threshold height at which someone is more likely to be male than female.
- Resources: Empirical data describing population.
  - Attribute, x, records measured height of each subject.
  - Response, $y = 1 \implies$ subject is male, $y = 0$ implies female.

Simple Histogram Based Approach

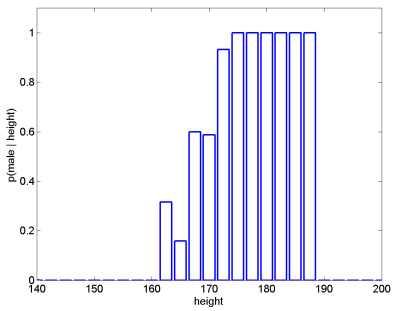
- Compute a histogram of the heights of 50 men and 50 women.





- Compute the fraction of male subjects in each bin.

$$p(y = 1|x) \approx \frac{N_m(x)}{N_m(x) + N_f(x)}.$$



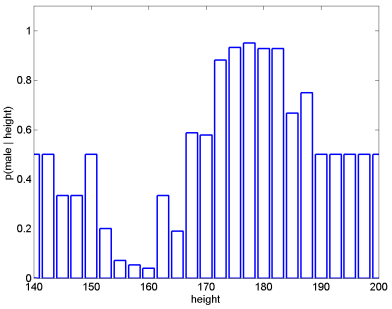
- Threshold is at about 169cm.

Advantages and Disadvantages

- Advantages - fast and simple to implement.
- Disadvantage - the model is heavily quantized
  - Each bin represents $\approx 1in$.
- Disadvantage - conditional probability $p(y = 1|\boldsymbol{x})$ non-monotonic
  - Not in accordance with our intuition.
  - Does not take advantage of data in adjacent bins.
- Smaller bins $\rightarrow$ fewer patterns per bin $\rightarrow$ more noise.
- Use the Laplace correction

$$p(y = 1|x) \approx \frac{N_m(x) + 1}{N_m(x) + N_f(x) + 2}$$

- In the absence of the data, the model claims ignorance!



Bayes Rule

- Thomas Bayes (1702 – 1761) - mathematician and clergyman.
- THE most important equation in machine learning, Bayes' Rule!

$$\begin{aligned} p(A, B) &= P(A|B)P(B) \\ = p(B, A) &= P(B|A)P(A) \\ \implies p(A|B) &= \frac{P(B|A)P(A)}{P(B)} \end{aligned}$$

- Probability as degree of belief in an uncertain proposition.
- Bayes rule provides a means to update our beliefs given data

$$posterior = \frac{likelihood \times prior}{normalising\ constant}$$

- Bayes' rule will crop up again later in the unit...

Classical Parametric Classification

- Given data $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$, where inputs $\boldsymbol{x}_n$ are continuous or discrete, and targets $\boldsymbol{y}_n$ are binary categorical variables.
- Devise parametric form of the *likelihood* function, $p(\boldsymbol{x}|\mathcal{C}_k)$.
  - A Gaussian distribution is appropriate.
- Determine the parameters of these distributions.
  - Mean μ_k and variance, σ_k^2 for each class.
- Compute the *a-priori* probability for each class, $p(\mathcal{C}_k)$.
  - Proportion of patterns belonging to class $\mathcal{C}_k$.
- Use Bayes' rule to find the *a-posteriori* probabilities

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})}.$$

- Assign to the class with the highest *a-posteriori* probability.

Fitting the model

- Adopt a Gaussian likelihood for each class,

$$p(x|\mathcal{C}_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left\{\frac{-(x-\mu_k)^2}{2\sigma_k^2}\right\}.$$

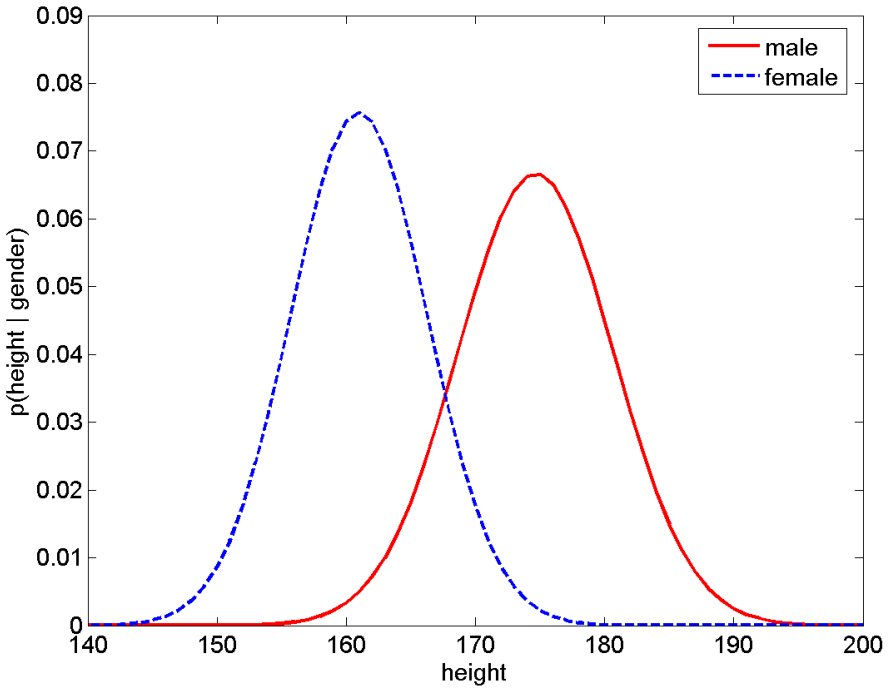
- Distributions characterised by a mean μ_k and a variance, σ_k^2.

$$\begin{aligned} \mu_k &= \frac{1}{N_k}\sum_{i\in\mathcal{C}_k} x_i, \\ \sigma_k^2 &= \frac{1}{N_k - 1}\sum_{i\in\mathcal{C}_k}(x_i - \mu_k)^2. \end{aligned} \qquad \begin{aligned} \mu_m &= 174.7 \\ \mu_f &= 161.0 \\ \sigma_m &= 6.0 \\ \sigma_f &= 5.3 \end{aligned}$$

- The prior probability, $p(\mathcal{C}_k) = N_k/N$.

$$p(\mathcal{C}_m) = \frac{50}{100} = 0.5 \qquad p(\mathcal{C}_f) = \frac{50}{100} = 0.5$$

Fitted *Likelihoods*



Applying Bayes Rule

- Use Bayes' rule gives that

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})}.$$

- But $p(\boldsymbol{x})$ is the same for both classes, so we can make decision using only

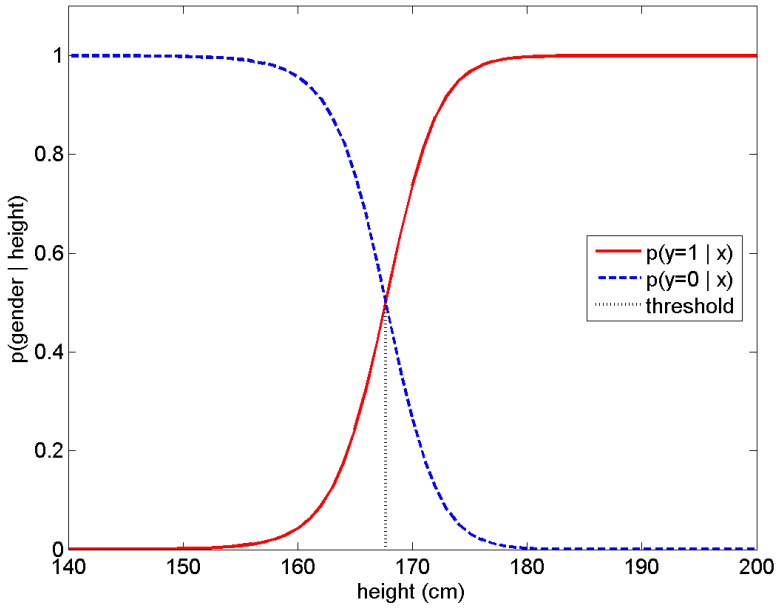
$$p(\mathcal{C}_k|\boldsymbol{x}) \propto p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k).$$

- $p(\boldsymbol{x})$ is a normalisation term to ensure that

$$\sum_k p(\mathcal{C}_k|\boldsymbol{x}) = 1$$

  i.e. in this case the subject is either a man or woman.

Inferred *A-Posteriori* Probability



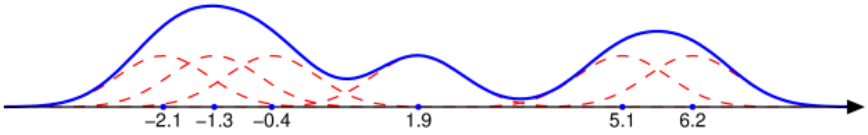
This model gives the threshold at 167.6 cm.

Non-Parametric Pattern Recognition

- Do not assume a particular form for the *likelihood*.
- Parzen kernel density estimator

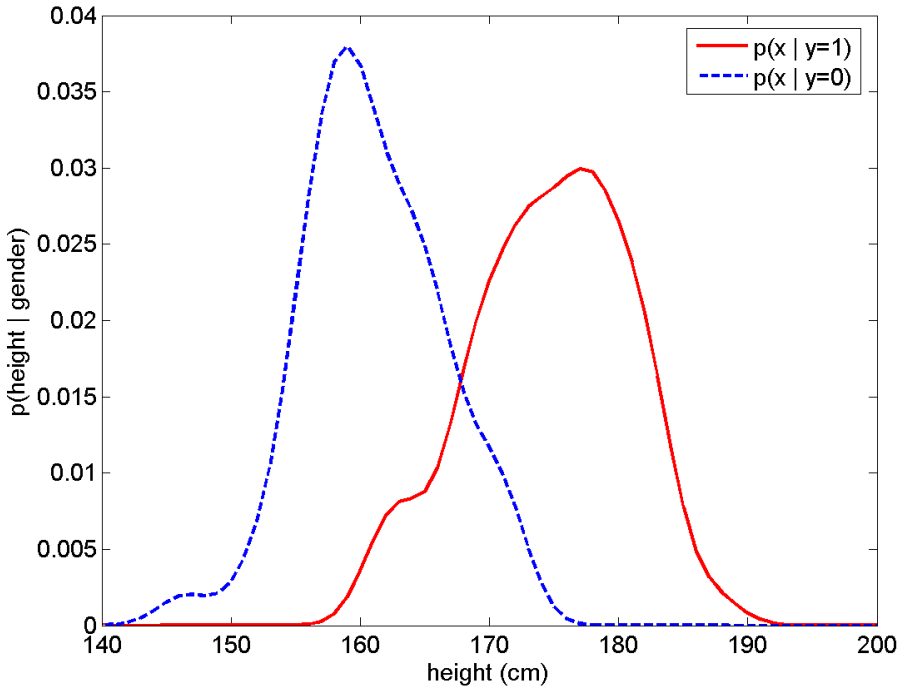
$$p(x) = \frac{1}{Nh} \sum_{i=1}^{N} \mathcal{K}\left\{\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right\}, \qquad \mathcal{K}(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\}.$$

- $\mathcal{K}(\cdot)$ is a kernel function, h is a smoothing parameter.
- The kernel enforces the idea that the likelihood should be similar for similar patterns.

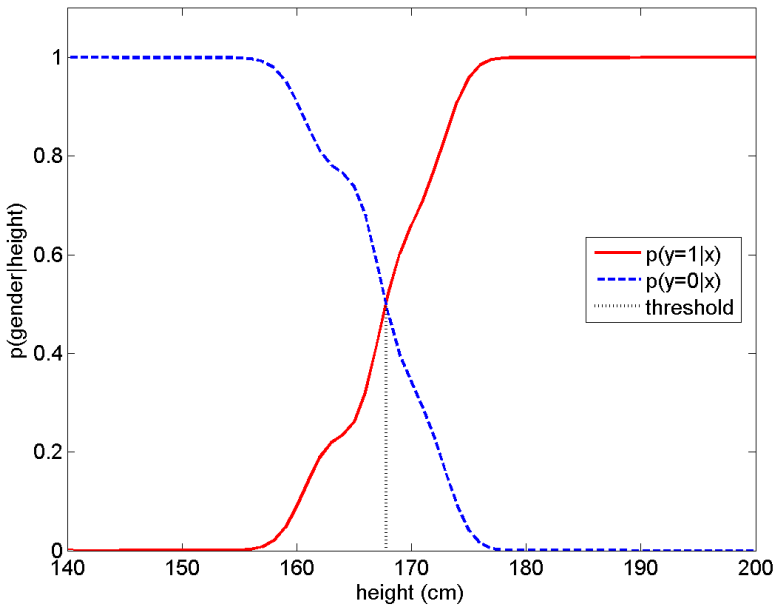


- We require $\int_{-\infty}^{+\infty} \mathcal{K}(x)dx = h$ for a proper density estimate.

Non-Parametric Class-Conditional Density Estimates



Non-Parametric *A-Posteriori* Probabilities



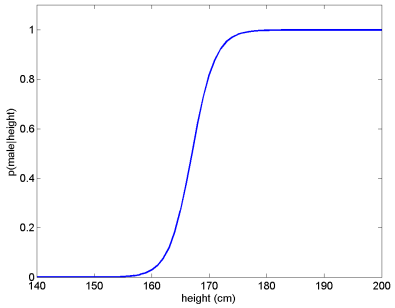
Threshold is at 167.8 cm.

Machine Learning via Optimisation

- Estimate the *a-posteriori* probability directly.
- Use the sigmoidal logistic function

$$p(y = 1|x) \approx \hat{y} = \frac{1}{1 + \exp\{-\alpha(x - \beta)\}}$$

- How should we select the model parameters α & β?
- α governs the slope.
- β governs the position.
- The model is suitably montonic.
- The output is constrained to lie in the range $(0,\ 1)$.
- We will use the maximum likelihood principle.



Maximum Likelihood

- If the model output $\hat{y}_n$ estimates $p(y_n = 1|\boldsymbol{x_n})$, the probability of observing the response for the n^{th} training pattern is,

$$p(y_n|\boldsymbol{x_n}) = (\hat{y}_n)^{y_n}(1 - \hat{y}_n)^{1-y_n}.$$

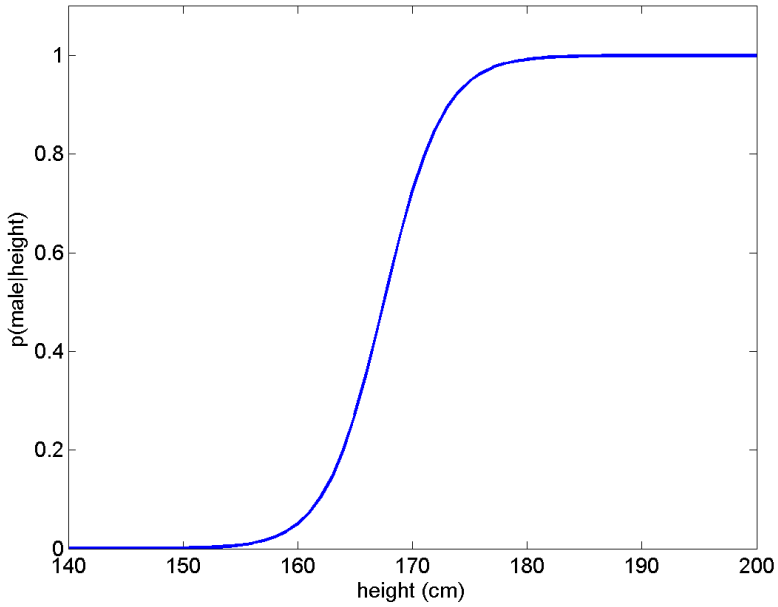
- The *likelihood* of observing the entire data set is then

$$\mathcal{L} = \prod_n (\hat{y}_n)^{y_n}(1 - \hat{y}_n)^{1-y_n}$$

- Train the network so that the outputs maximise $\mathcal{L}$.
- The best classifier the one that is most likely to generate the data observed in the training set.
- Find α & β that minimise the negative log-likelihood,

$$-\log \mathcal{L} = -\sum_{n=1}^{N} \left[y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right].$$

Model Based *A-Posteriori* Probabilities



Threshold is at 167.5 cm.

Comparison of Approaches

- Parametric approach:
  - Strong assumptions about the distribution of data - inflexible.
  - Bad choices may lead to poor performance.
  - Few parameters - easy to estimate accurately.
  - Can build expert knowledge into the model.
- Non-parametric approach:
  - No assumptions about the distribution of data - flexible.
  - Number of parameters grows with number of training patterns.
  - Parameter estimation more difficult.
- Model based approach:
  - Estimate *a-posteriori* probability directly.
  - Can be either parametric or non-parametric.
- Machine learning is *mostly* concerned with model based non-parametric inference.

The Value of Probabilistic Classifiers

- Discrete classifiers give yes/no decisions.
  - Solving a less complex problem, should be easier.
  - Nearest neighbour methods*, decision trees*, SVM*, etc.
- Probabilistic classifiers estimate the *a-posteriori* probability.
  - Not only gives a decision, but indicates confidence.
  - Logistic regression, artificial neural networks, etc.
- Advantages of probabilistic classification:
  - Dealing with disparate training set and operational priors.
  - Minimum risk classification.
  - Make a reject option possible.
  - Combining models to get better predictions.
- In choosing a classifier, we need to decide if any of these factors are important.

* An estimates of *a-posteriori* probability can be obtained, but it isn't a fundamental element of the method.

Setting Rejection Thresholds

- The margin between between the highest and next highest *a-posteriori* probabilities gives a measure of the confidence of the classification.
- If the margin is small we might choose not to classify the pattern at all, but instead reject the pattern.
- In a medical screening test if a confident classification cannot be made it might be better to refer the decision to a human expert rather than to make an unsafe automatic classification.
- Examples:

$p(\mathcal{C}_1)$	$p(\mathcal{C}_2)$	$p(\mathcal{C}_3)$	Result
0.15	0.8	0.05	clearly $\mathcal{C}_2$
0.29	0.35	0.36	reject

Compensating For Prior Probabilities

- Sometimes the prior probabilities in the training set are different from those used in operation.
  - Operational priors may vary.
  - Practical difficulties in collecting data.
  - Computational convenience.
- This is a *very* common problem in practical applications.
- Compensate using Bayes' rule:

$$p_o(\mathcal{C}_k|\boldsymbol{x}) \propto p(\boldsymbol{x}|\mathcal{C}_k)p_t(\mathcal{C}_k) \times \frac{p_o(\mathcal{C}_k)}{p_t(\mathcal{C}_k)}$$

- Can re-normalise so that probabilities sum to one if requires.
- No need to retrain the model!

Minimum Risk Classification

- Some types of misclassification are worse than others.
- Medical screening tests - false negatives are much more serious than false positives.
- Define matrix, $\boldsymbol{L}$, of costs associated with each outcome

True	Predicted	
	cancer	normal
cancer	0	1000
normal	1	0

- Build costs into decision rule, choose class, $\mathcal{C}_j$, minimising

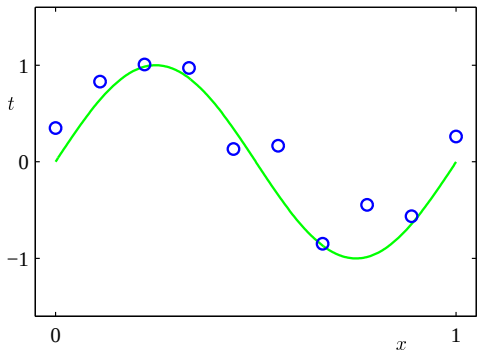
$$\sum_k L_{kj} p(\mathcal{C}_k|\boldsymbol{x})$$

- This is a called a *minimum risk* classifier.

Generalisation and Over-fitting

- We don't just want a model that works well on the training data, but one that gives accurate predictions on operational data. We call this *Generalisation*.
- Consider the problem of fitting a polynomial to a set of noisy data from a sinusiodal function.

$$\begin{aligned} y_i &= sin(2\pi x_i) + \epsilon_i, \\ \epsilon_i &\sim \mathcal{N}(0, \sigma^2) \end{aligned}$$



- Aim: predict the sinusoid.
- But ignore the noise!
- Use a model of the form

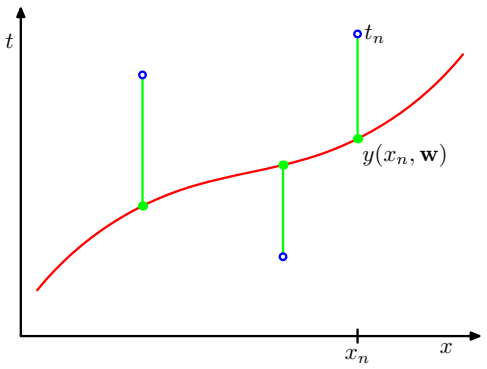
$$\hat{y}(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j.$$

Fitting the Model

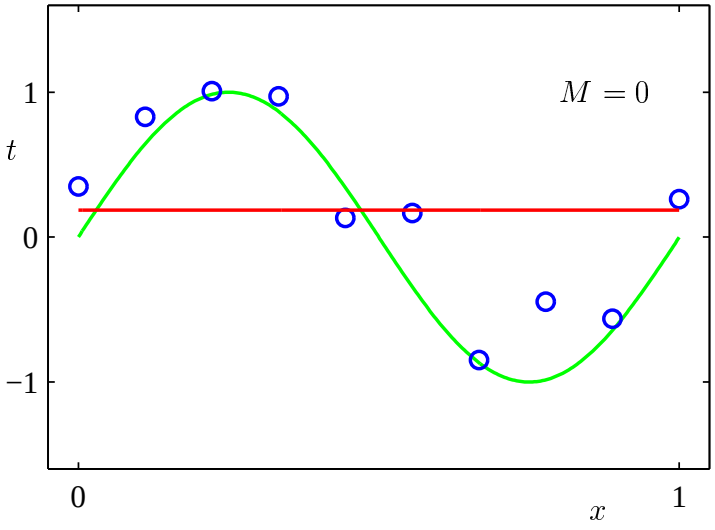
- Set parameters, $\boldsymbol{w} = (w_0, w_1, \ldots, w_M)^T$, so as to minimise the sum of squared errors (SSE)

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} [\hat{y}(x_n; \boldsymbol{w}) - y_n]^2$$

- Places a greater penalty on the function the further it is from the data.
- The solution can be found in closed form.
- Maximum likelihood interpretation.
- How should we choose the order of the polynomial?

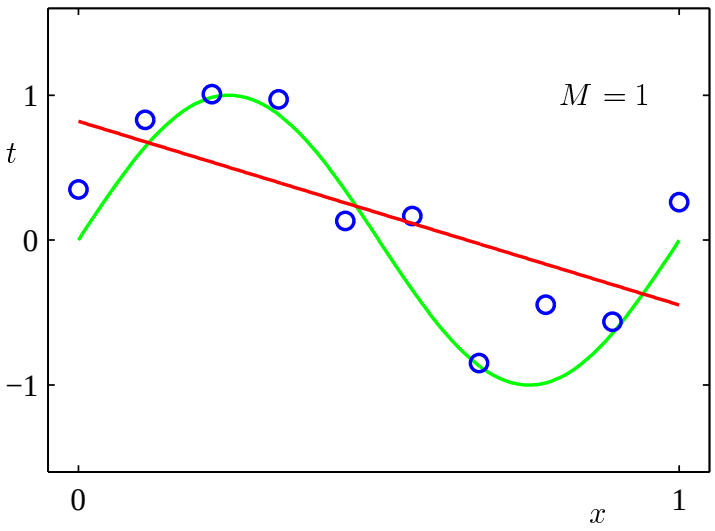


Polynomial of Order M=0



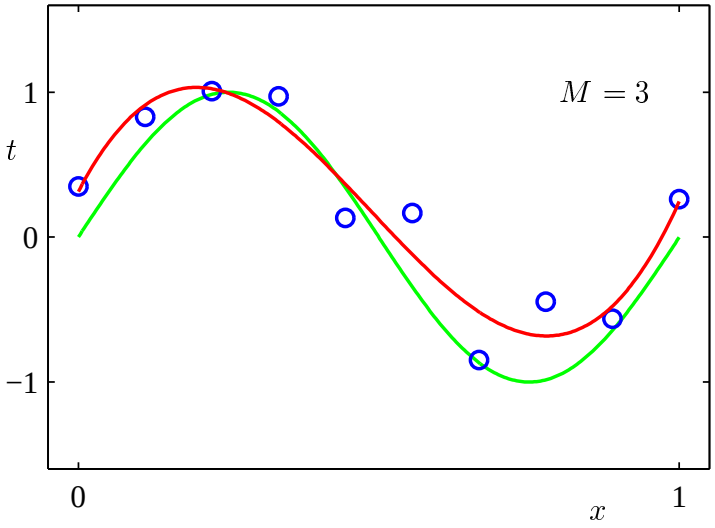
- Model severely under-fits the data.
- The model is far too simple.
- Error high on both training data and in operation.

Polynomial of Order M=1



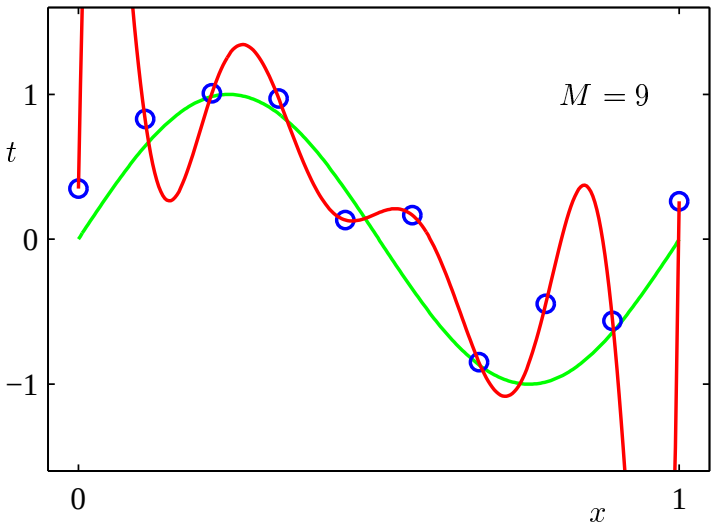
- Model still under-fits the data.
- The model is still a bit too simple.
- Significant error on both training data and in operation.

Polynomial of Order M=3



- Model provides a good fit to the data.
- The complexity of the model is well matched to the data.
- Low error on both training data and in operation.

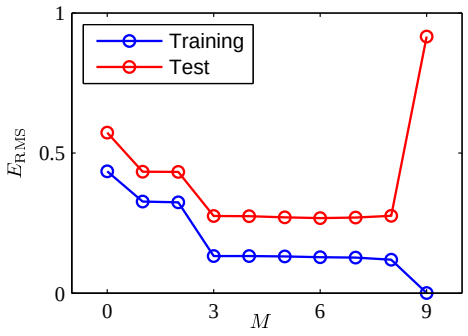
Polynomial of Order M=9



- Model badly over-fits the data.
- The model is too complex, it can memorise the noise component.
- Zero error on training data but high error in operation.

Model Complexity

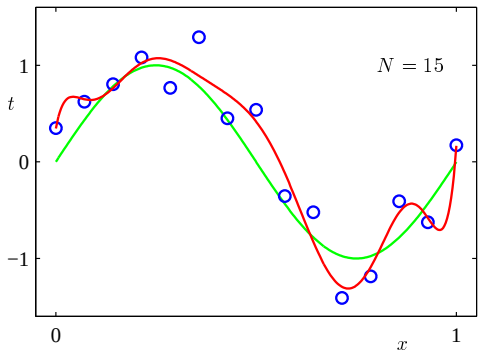
- Need to match the complexity of the model to match the difficulty of the learning task.
- Plot training set and test set (operational) SSE for different values of M.
- For small M the model is too simple to explain the deterministic component.
- For large M the model explains the deterministic component but also memorises the noise.
- In between we get the best generalisation.

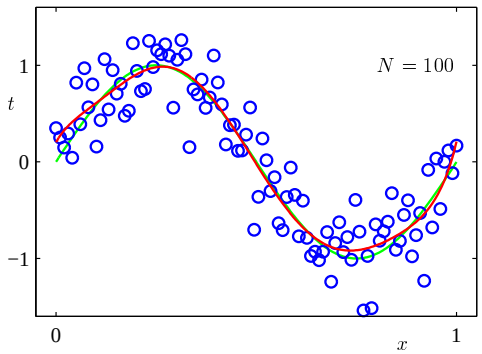


- **Low training error does not imply a good model!**

Over-Fitting in Small and Large Datasets

- In general the more training data the less likely we are to see over-fitting, even with complex models.





- Large and Small datasets present different problems:
  - Large - computational expense, efficient algorithms needed.
  - Small - over-fitting likely, need to control model complexity.
- This is the down-side of non-parametric modelling - beware!

Summary

- In the this lecture:
  - Example of statistical pattern recognition.
  - Parametric and non-parametric modelling.
  - Machine learning and optimisation.
  - Maximum likelihood.
  - Value of probabilistic modelling.
  - Generalisation and over-fitting.

- In the next lecture:
  - Multivariate machine learning.
  - Nearest neighbour methods

Selected figures taken from C. M. Bishop, "Pattern Recognition and Machine Learning", Springer 2006.