Linear Models for Regression and Classification

Sumantra Dutta Roy
Department of Electrical Engineering
Indian Institute of Technology Delhi
Hauz Khas, New Delhi - 110 016, INDIA.
http://www.cse.iitd.ac.in/∼sumantra
sumantra@ee.iitd.ac.in

Introduction

- Unsupervised Learning: Given points $\mathbf{x}_1, \ldots \mathbf{x}_N$: N points in $D-$ dimensional space.
  Aim: To cluster/group them/put them into clumps/classes, given no other information
- Supervised Learning: Given points $\mathbf{x}_1, \ldots \mathbf{x}_N$: N points in $D-$ dimensional space, and target values/labels $t_1, \ldots t_N$
  Aim of Regression: Prediction $t_i = y(\mathbf{x}_i, \mathbf{w}) + \varepsilon$,
  $\mathbf{w}$: parameters, ε: noise (modelled/unmodelled)
  Aim of Classification: Labels $t_i \in \{0, 1\}$ or $\{-1, 1\}$
  or multi-class: $\mathbf{t}_i = [0 \ldots 010 \ldots] \equiv \mathscr{C}_j$

Regression

- Regression: prediction. line fitting $y = mx + c$
- General: t: 1-D target variable, what is observed

$$t = y(\mathbf{x}) + \varepsilon$$

  $\mathbf{x}$ is the $M-1$-dimensional input data
  $y(\cdot)$: 1-D function of $(M-1)-$dim input: the model
  ε: noise ($\sim$ can't model, sometimes modelled)
- Reconciliation: may not be able to model all well
- Simple 2-D case: y: an implicit function of $\mathbf{x}$.
  e.g., $f(x,y) = ax+by+c = 0$, or $w_2x_2 + w_1x_1 + w_0 = 0$
- $y(\mathbf{x}) = y(\mathbf{x},\mathbf{w}) = w_2x_2 + w_1x_1 + w_0x_0$. $x_0 = 1$, w_0: bias
- Written equivalently in two ways:
  $y(\mathbf{x},\mathbf{w}) = \mathbf{w}^T\mathbf{x}$, for $(D+1) = M-$ dim data, $x_0 = 1$, or
  $y(\mathbf{x},\mathbf{w}) = \mathbf{w}^T\mathbf{x} + w_0$, for $D = (M-1)-$ dim data

- $y(\mathbf{x},\mathbf{w}) = w_0 x_0 + \ldots w_{M-1} x_{M-1} = \sum_{j=0}^{M-1} w_j x_j$
- Generalising to scalar basis functions $\phi_j(\mathbf{x})$:
- $y(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = w_o \phi_0(\mathbf{x}) + \ldots w_{M-1} \phi_{M-1}(\mathbf{x})$
- Model: linear combo of fixed basis fns (lin/non-lin)

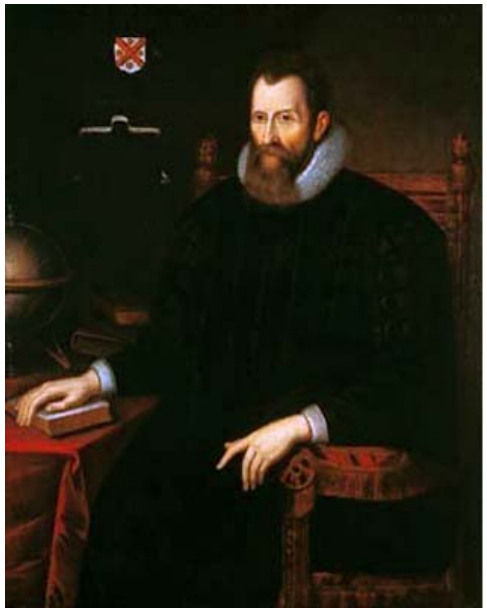
$$y(\mathbf{x},\mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

- Not most general, but practically imp! Examples:
  - Polynomial basis fns: x^j: global, unlike splines
  - Gaussian basis fns
  - Sigmoidal basis fns
  - Fourier basis fns
  - Wavelet basis fns localised in space & freq

Maximum Likelihood, Least Sq

- $t = y(\mathbf{x}) + \varepsilon$; $t = y(\mathbf{x}, \mathbf{w}) + \varepsilon$
- t: target variable $y(\cdot)$: deterministic fn (model)
- $\mathbf{x}$: input, $\mathbf{w}$: parameters, ε: noise
- ε: take as the unmodelled part: the residue, or
- . . . model ε as well. Common: $\varepsilon = \mathcal{N}(0, \sigma^2)$
- $t = \mathcal{N}(y(\cdot), \sigma^2)$, Mean: $y(\mathbf{x}, \mathbf{w})$, variance: σ^2
- If no $y(\cdot)$, t usually 0, or small +/-: weighing m/c noise: zero error, offset: $y(\cdot)$
- $p(t|\mathbf{x}, \mathbf{w}, \sigma^2) = p(t|\mathbf{w}, \sigma^2) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \sigma^2)$

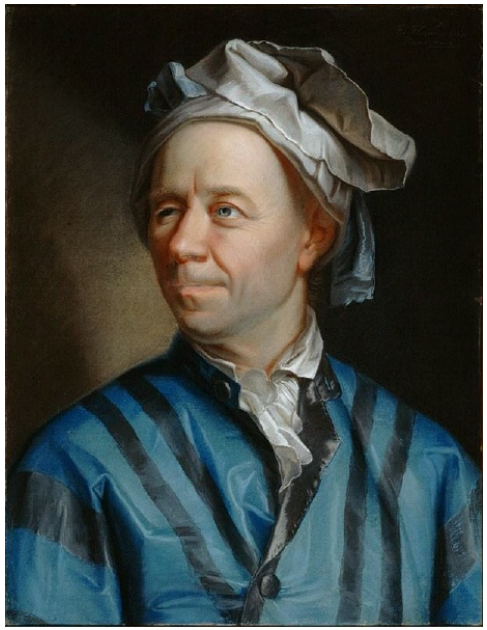
- Given N obs $\mathbf{t} = \{t_1 \ldots t_N\} \equiv$ input $\mathbf{X} = \{\mathbf{x}_1 \ldots \mathbf{x}_N\}$
- Assume i.i.d.: independence: probs multiplied, identically distr: all from same model $\mathcal{N}(t|y(\cdot), \sigma^2)$
- $p(\mathbf{t}|\mathbf{w}, \sigma^2) = \prod_{i=1}^{N} \mathcal{N}(t_i|y(\mathbf{x}_i, \mathbf{w}), \sigma^2)$ $y(\cdot) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)$
- LHS: Likelihood: prob of getting the data, given model params: Reasonable to maximise it!
- Maximise log-likelihood: inc fn, mults $\rightarrow$ additions
- $\log p(\mathbf{t}|\mathbf{w}, \sigma^2) = \sum_{i=1}^{N} \log \mathcal{N}(t_i|y(\mathbf{x}_i, \mathbf{w}), \sigma^2)$
- log-likelihood $= \sum_{i=1}^{N} \log(\frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2}{2\sigma^2})$
- $= -\frac{N}{2}\log 2\pi - \frac{N}{2}\log \sigma^2 - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{N}(t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2$
- Params σ^2, $\mathbf{w}$: $\frac{\partial \text{ log-likelihood}}{\partial \text{parameter}} = 0$



Logarithms

John Napier
[1550-1617]

Leonhard Euler
[1707-1783]



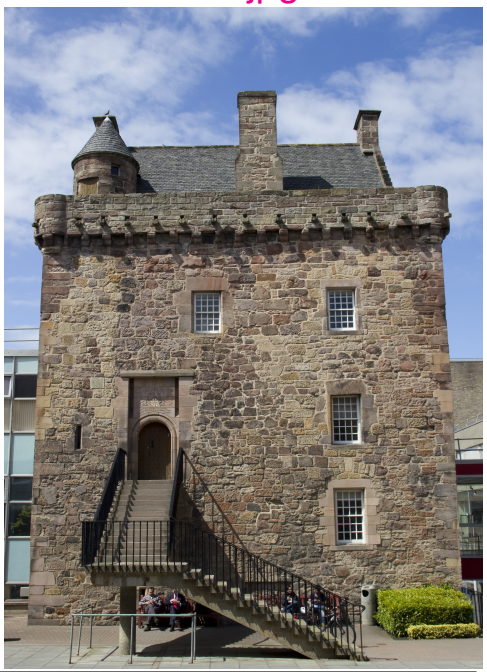
https://upload.wikimedia.org/wikipedia/commons/e/e3/John_Napier.jpg

https://upload.wikimedia.org/wikipedia/commons/d/d7/Leonhard_Euler.jpg

Edinburgh Napier UNIVERSITY

https://upload.wikimedia.org/wikipedia/en/e/e4/Edinburgh_Napier_University_logo.png

https://upload.wikimedia.org/wikipedia/commons/1/1f/Merchiston_Castle



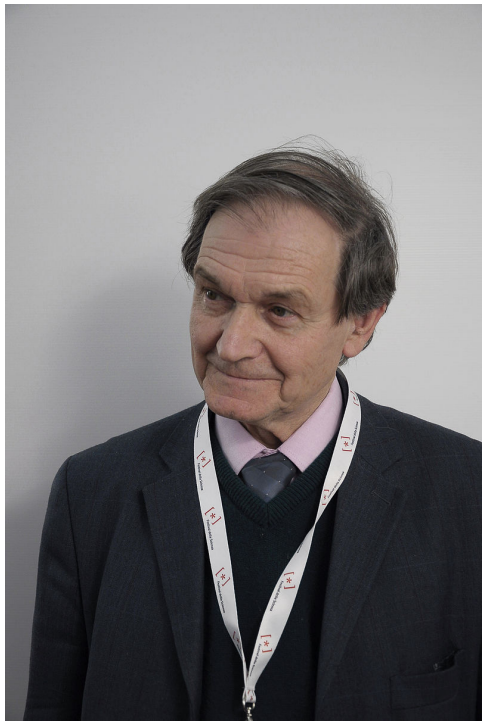
- param = σ^2:
  $\frac{\partial \text{ log-lh}}{\partial \sigma^2} = -\frac{N}{2}\frac{1}{\sigma^2} - \frac{-1}{\sigma^4}\frac{1}{2}\sum_{i=1}^{N}(t_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i))^2 = 0$
- $\sigma^2_{ML} = \frac{1}{N}\sum_{i=1}^{N}(t_i - \mathbf{w}_{ML}^T\boldsymbol{\phi}(\mathbf{x}_i))^2$; $y(\mathbf{x}_i, \mathbf{w}) = \mathbf{w}_{ML}^T\boldsymbol{\phi}(\mathbf{x}_i)$
- Phy Sig: The ML variance is the sample variance of the target values around the regression fn
- param = $\mathbf{w}$: log-lh = $(\sum a_i^2 = \mathbf{a}^T\mathbf{a}; \frac{\partial \mathbf{a}^T\mathbf{a}}{\partial \mathbf{a}} = 2\mathbf{a}^T)$
  $-\frac{N}{2}\log 2\pi - \frac{N}{2}\log\sigma^2 - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{N}(t_i - \boldsymbol{\phi}^T(\mathbf{x}_i)\mathbf{w})^2$
- $\frac{\partial \text{log-lh}}{\partial \mathbf{w}^T} = -\frac{1}{\sigma^2}\frac{1}{2}2(-1)\sum_{i=1}^{N}(t_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i))\boldsymbol{\phi}^T(\mathbf{x}_i) = 0$:
- $\Longrightarrow \sum_{i=1}^{N} t_i\boldsymbol{\phi}^T(\mathbf{x}_i) = \mathbf{w}^T\sum_{i=1}^{N}\boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}^T(\mathbf{x}_i)$ ($\mathbf{w}^T$ out)
- Break the sums up as inner products

- $\sum_{i=1}^{N} t_i \boldsymbol{\phi}^T(\mathbf{x}_i) = \mathbf{w}^T \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}^T(\mathbf{x}_i) \implies$
- $[t_1 \dots t_N] \begin{bmatrix} \boldsymbol{\phi}^T(\mathbf{x}_1) \\ \cdots \\ \boldsymbol{\phi}^T(\mathbf{x}_N) \end{bmatrix} = \mathbf{w}^T [\boldsymbol{\phi}(\mathbf{x}_1) \dots \boldsymbol{\phi}(\mathbf{x}_N)] \begin{bmatrix} \boldsymbol{\phi}^T(\mathbf{x}_1) \\ \cdots \\ \boldsymbol{\phi}^T(\mathbf{x}_N) \end{bmatrix}$
- $\implies \mathbf{t}^T\boldsymbol{\Phi} = \mathbf{w}^T(\boldsymbol{\Phi}^T\boldsymbol{\Phi})$. 'Design Matrix': bases $\forall \mathbf{x}_i$
- $\boldsymbol{\Phi}_{N\times M} = \begin{bmatrix} \boldsymbol{\phi}^T(\mathbf{x}_1) \\ \cdots \\ \boldsymbol{\phi}^T(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) \dots \phi_{M-1}(\mathbf{x}_1) \\ \cdots \\ \phi_0(\mathbf{x}_N) \dots \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}_{N\times M}$
- Transpose both sides: $(\mathbf{t}^T\boldsymbol{\Phi})^T = (\mathbf{w}^T(\boldsymbol{\Phi}^T\boldsymbol{\Phi}))^T \implies$
- $\boldsymbol{\Phi}^T\mathbf{t} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{w} \implies \boxed{\mathbf{w}_{ML} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{t} = \boldsymbol{\Phi}^\dagger\mathbf{t}}$
- $\boldsymbol{\Phi}^\dagger$: Moore-Penrose Pseudo-Inverse

Moore-Penrose Pseudo-Inverse

E. H. Moore
(1920)
[1862-1932]

A. Bjerhammar
(1951)
[1917-2011]



R. Penrose
(1955)
[1931-]

https://upload.wikimedia.org/wikipedia/commons/9/96/Moore_Eliakim_2.jpeg

https://thumbnail.myheritageimages.com/021/731/40021731/500/500151_62403790fd28b4g41eg208_W_189x256.jpg

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d5/Roger_Penrose_at_Festival_della_Scienza_Oct_29_2011.jpg/

800px-Roger_Penrose_at_Festival_della_Scienza_Oct_29_2011.jpg