Machine Learning, Machine Learning (extended)

3 - Supervised Learning: Linear Modelling by Maximum Likelihood

Kashif Rajpoot

k.m.rajpoot@cs.bham.ac.uk

School of Computer Science

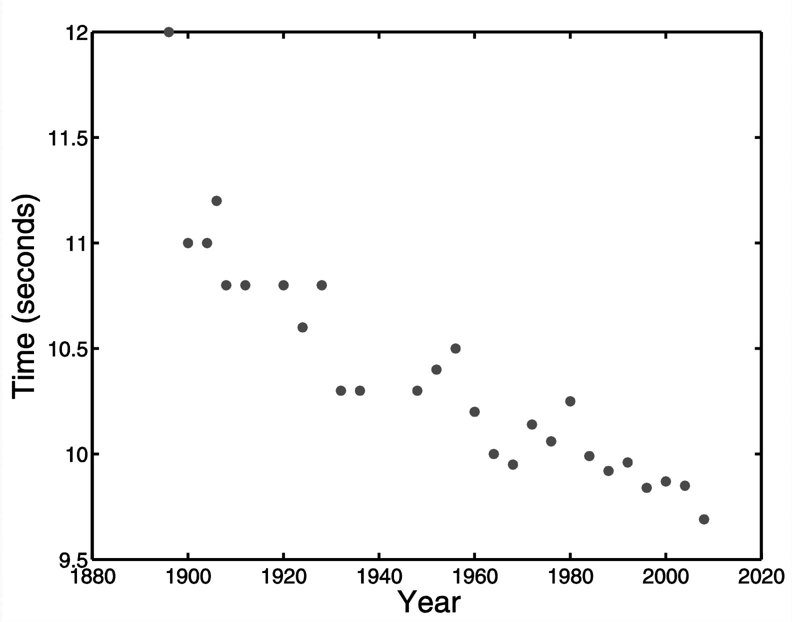
University of Birmingham

Outline

- Linear modelling
- Error = noise
- Thinking generatively
- Likelihood
- Maximum likelihood
- Model complexity
- Bias-variance tradeoff

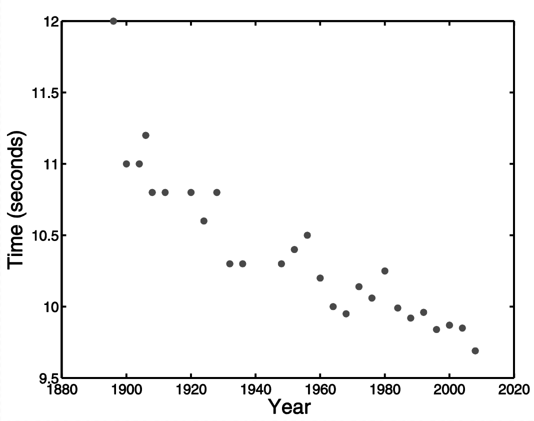
Linear modelling

- One of the most straightforward learning problems
  - Learn a linear function between attributes and responses
- Is there a functional dependence between Olympics year and 100m winning time?
  - Draw a line?
- Can we predict winning time for future games?



Linear modelling

- Learner model/function
  - Maps input attributes to output response
- Let's consider we can predict time $t = f(x)$
  - x?
  - t?
- Training samples
  - N attribute-response pairs $(x_1, t_1), (x_2, t_2), \ldots (x_N, t_N)$



Linear modelling

- Linear modelling by minimizing loss

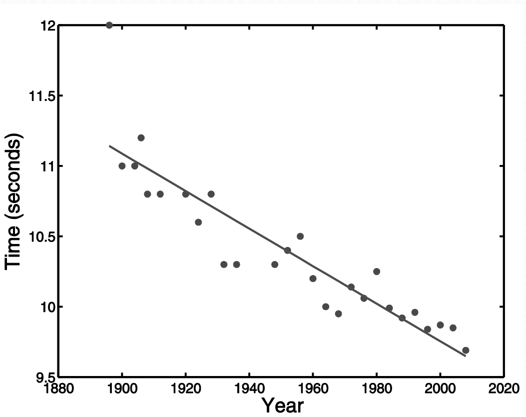
$$t_n = \widehat{w}_0 + \widehat{w}_1 x_n$$

where the model parameters are estimated from Olympics data

$$\widehat{w}_1 = \frac{\overline{xt} - \bar{x}\bar{t}}{\overline{x^2} - (\bar{x})^2}$$

$$\widehat{w}_0 = \bar{t} - \widehat{w}_1 \bar{x}$$

$$f(x; w_0, w_1) = 36.416 - 0.013x$$



- With the vector notation

$$t_n = \widehat{\boldsymbol{w}}^T \boldsymbol{x}_n$$

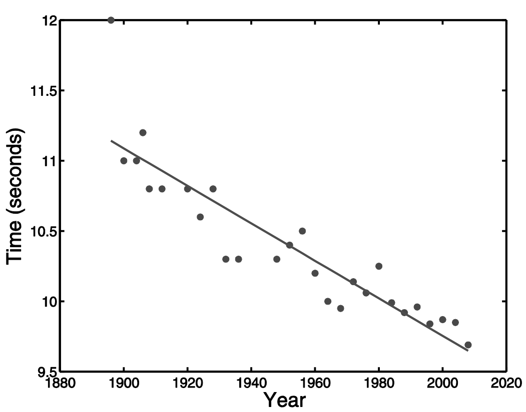
$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \text{ and } \boldsymbol{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix}$$

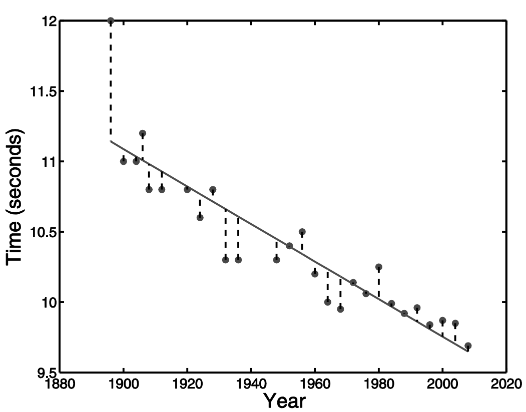
while model parameters are estimated as: $\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{t}$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}$$

Linear modelling

- Linear modelling by minimizing loss
  - Model shows to capture trend in data observations
  - Model fails to explain each data observation correctly (i.e. error)
- Let's recall our assumptions
  - There is a relationship between Olympics year and winning time
  - This relationship is linear
  - This relationship will hold in future
- Are these good assumptions?
- Still, ignoring the error is not right
  - Let's consider error as noise and model it



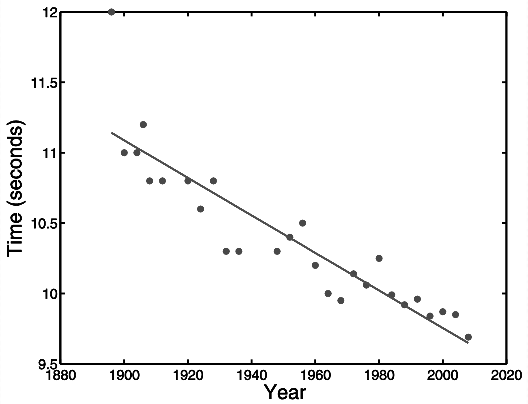


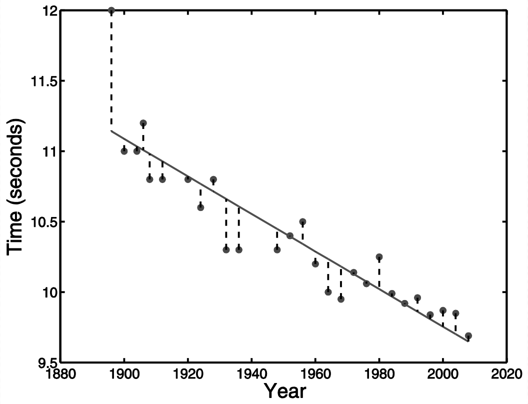
Linear modelling

- Approaches to linear modelling
  - Minimize the loss: minimize the squared error between model predictions and training observations
  - Maximize the likelihood: maximize the likelihood of match between model predictions and training observations
    - Build a model to generate data
    - Explicitly model the noise (i.e. the error between model and observations)

Thinking generatively

- Let's recall that our assumptions are weak
  - The process generating this data is very complex
- Still, can we try to build a model that can *generate* such data?
- *Generative modelling*: build a model to *generate* data that *looks like* data observations
  - $t_n = \boldsymbol{w}^T \boldsymbol{x}_n$
  - Error?
  - Modelling error





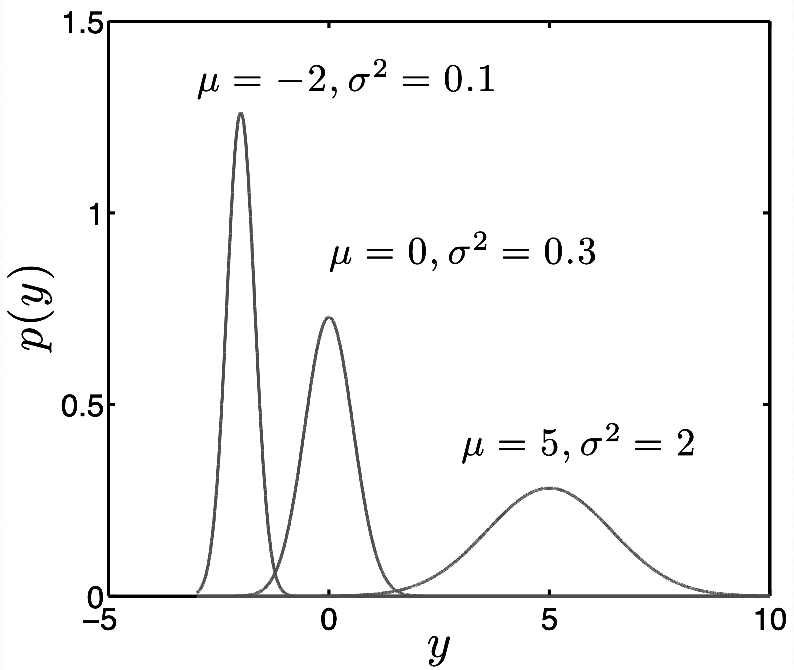
Refresher: Probability & Stats (DIY)

- Random variable (X, Y)
- Probability of tossing a coin (head=1, tail=0)
  - $P(Y = 0) = 0.5$
  - $P(Y = 1) = 0.5$
- Conditional probability
  - $P(Y = y|X = x)$
- Joint probability
  - $P(Y = y, X = x)$
- Probability distribution
- Probability density function (pdf)
- Expectation
  - $E(X) = \sum_x xP(x)$

Refresher: Gaussian pdf

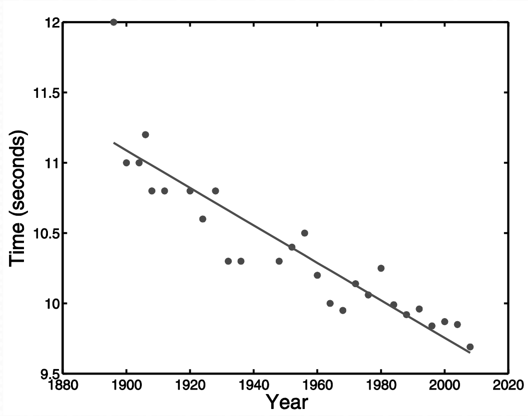
- Gaussian (or normal) pdf

$$p(y|\mu,\sigma^2) = \mathcal{N}(\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\}$$

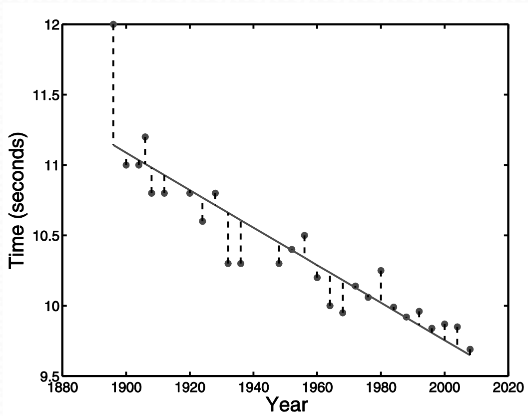


Thinking generatively

- *Generative modelling*: build a model to *generate* data that *looks like* data observations
  - $t_n = \boldsymbol{w}^T \boldsymbol{x}_n$
  - Error?



- Generative model with noise considerations:
  - $t_n = \boldsymbol{w}^T \boldsymbol{x}_n + \varepsilon_n$
  - Additive noise?



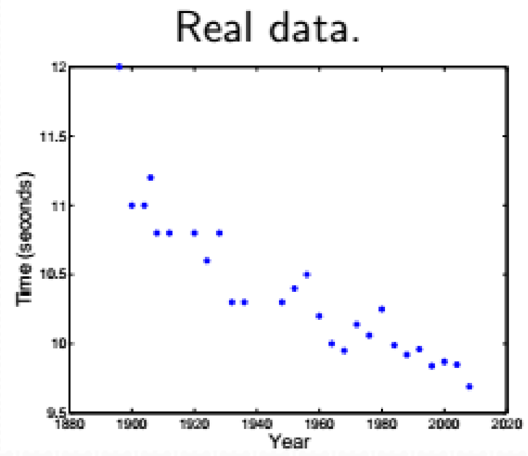
- Noise (ε_n)
  - Is it a random variable?
  - Is it discrete or continuous?
  - What pdf for ε_n?

Thinking generatively

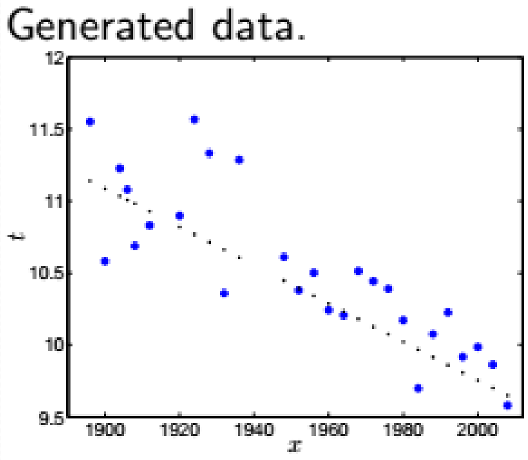
- Probability density of ε_n

  $$p(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N) = \prod_{n=1}^{N} p(\varepsilon_n)$$

  each ε_n is independent
- Let's use a Gaussian density for $p(\varepsilon_n) = \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(0{,}0.05)$



- Generative model with noise considerations:
  - $t_n = \boldsymbol{w}^T \boldsymbol{x}_n + \varepsilon_n$
- Deterministic component (i.e. trend)
- Random component (i.e. noise)

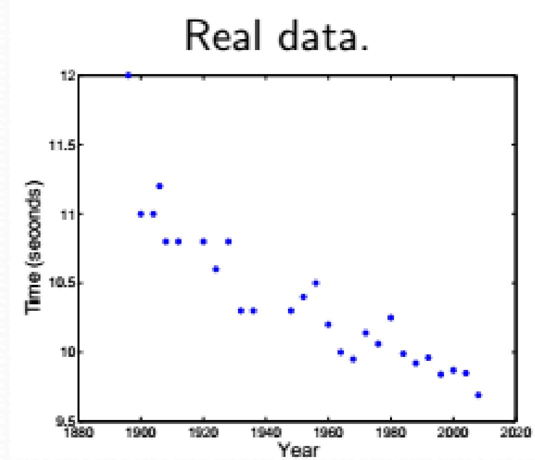


Thinking generatively

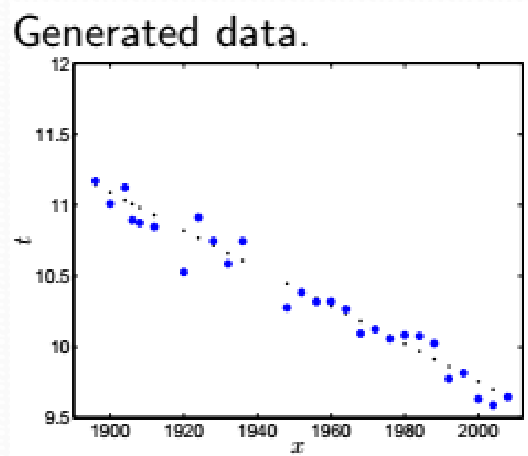
- Probability density of ε_n

  $$p(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N) = \prod_{n=1}^{N} p(\varepsilon_n)$$

  each ε_n is independent
- Let's use a Gaussian density for $p(\varepsilon_n) = \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(0{,}0.01)$

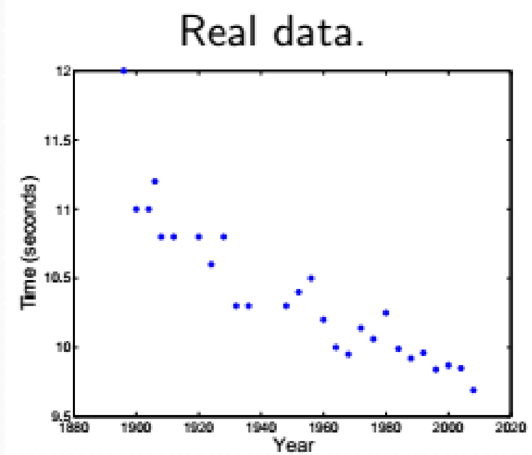


- Generative model with noise considerations:
  - $t_n = \boldsymbol{w}^T \boldsymbol{x}_n + \varepsilon_n$
- Deterministic component (i.e. trend)
- Random component (i.e. noise)

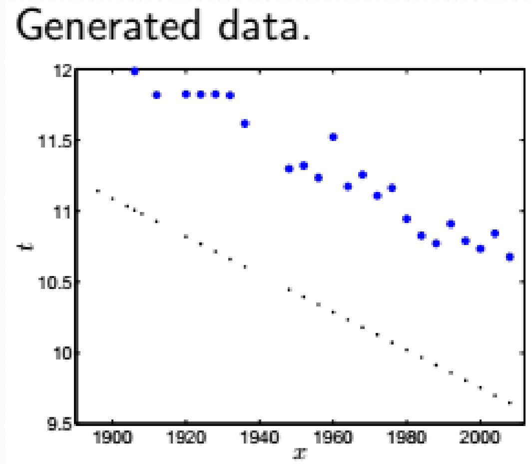


Thinking generatively

- Probability density of ε_n
  $$p(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N) = \prod_{n=1}^{N} p(\varepsilon_n)$$
  each ε_n is independent
- Let's use a Gaussian density for $p(\varepsilon_n) = \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(1, 0.01)$



- Generative model with noise considerations:
  - $t_n = \boldsymbol{w}^T \boldsymbol{x}_n + \varepsilon_n$
- Deterministic component (i.e. trend)
- Random component (i.e. noise)



Thinking generatively

- Generative model with noise considerations:
  - $t_n = \boldsymbol{w}^T \boldsymbol{x}_n + \varepsilon_n$
  - $t_n = f(\boldsymbol{x}_n; \boldsymbol{w}) + \varepsilon_n$, where $p(\varepsilon_n) = \mathcal{N}(0, \sigma^2)$
  - Model is now determined not only by $\boldsymbol{w}$ but also σ^2

- t_n can now be considered a random variable itself, due to addition of ε_n
  - i.e. for a given $\boldsymbol{x}_n$, t_n is not a single fixed value but rather is drawn out from a pdf
  - Thus finding $\boldsymbol{w}$ and σ^2 by minimizing loss (with least squares approach) is not possible

Thinking generatively

- The probability density of t_n is:
$$p(t_n|\boldsymbol{x}_n, \boldsymbol{w}, \sigma^2) = \mathcal{N}(\boldsymbol{w}^T\boldsymbol{x}_n, \sigma^2)$$

- Let's recall $p(\varepsilon_n) = \mathcal{N}(0, \sigma^2)$

$$y = a + z$$
$$p(z) = \mathcal{N}(m, s)$$
$$p(y) = \mathcal{N}(m + a, s)$$

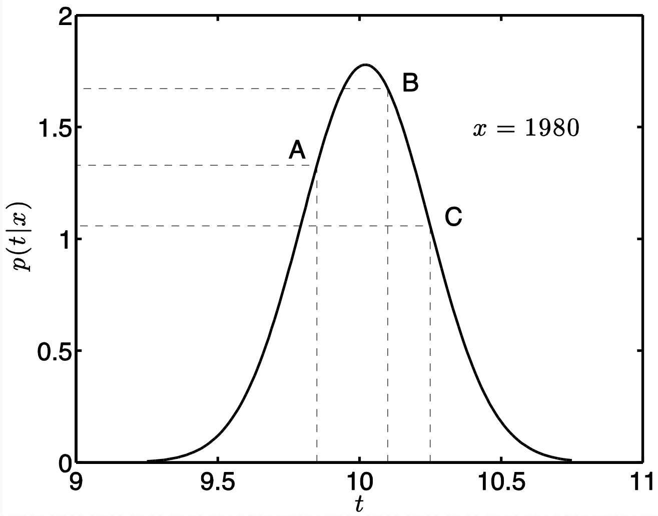
- Note that $\boldsymbol{w}^T\boldsymbol{x}_n$ determines the mean (i.e. trend) and σ^2 determines variance (i.e. noise)

Thinking generatively

- Let's use this generative model to look at pdf for year 1980

$$p(t_n|\boldsymbol{x}_n = [1{,}1980]^T, \boldsymbol{w} = [36.416, -0.0133]^T, \sigma^2 = 0.05)$$

This generates a Gaussian pdf
$\mathcal{N}(\boldsymbol{w}^T\boldsymbol{x}_n = 10.02, \sigma^2 = 0.05)$



- Height of the curve corresponds to how likely it is to observe a particular t
- A, B, or C – most likely?
- *Likelihood* at $t_n = 10.25$?
- Can we determine $\boldsymbol{w}$ and σ^2 that maximize *likelihood* of observed data t_n with generated data?

Likelihood

- For each input-response pair $(\boldsymbol{x}_n, t_n)$, we have a Gaussian likelihood: $p(t_n|\boldsymbol{x}_n, \boldsymbol{w}, \sigma^2)$
- Maximize the likelihood of matching all observed responses $(t_1, t_2, \dots t_N)$ conditioned on the observed data $(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \boldsymbol{x}_N)$ and model parameters $(\boldsymbol{w}, \sigma^2)$

$$p(t_1, t_2, \dots t_N|\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \boldsymbol{x}_N, \boldsymbol{w}, \sigma^2) = p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2)$$

- Let's recall that $p(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N) = \prod_{n=1}^{N} p(\varepsilon_n)$

i.e. each ε_n is independent

- Thus, likelihood can be estimated as:

$$L = p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2)$$

Maximum likelihood

- Likelihood estimate

$$L = p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2)$$

- How to find $\boldsymbol{w}$ and σ^2?
- Find "best" $\boldsymbol{w}$ and σ^2 that maximize likelihood L

$$\underset{\boldsymbol{w}, \sigma^2}{argmax}\, L$$

- It's mathematically convenient if we, instead, maximize the log of likelihood L

$$\underset{\boldsymbol{w}, \sigma^2}{argmax}\, \log(L)$$

- Model parameters $(\boldsymbol{w}, \sigma^2)$ that maximize log likelihood $(\log(L))$ also maximize likelihood (L)

Maximum likelihood

- Likelihood estimate

$$L = p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2)$$

- Log likelihood estimate

$$\log(L) = \log\left[\prod_{n=1}^{N} \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2)\right] = \sum_{n=1}^{N} \log[\mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2)]$$

- Let's recall

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\}$$

so

$$\mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(t_n - \boldsymbol{w}^T \boldsymbol{x}_n)^2\right\}$$

Maximum likelihood

$$\log(L) = \sum_{n=1}^{N} \log[\mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2)]$$

- Considering that

$$\mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(t_n - \boldsymbol{w}^T \boldsymbol{x}_n)^2\right\}$$

we get

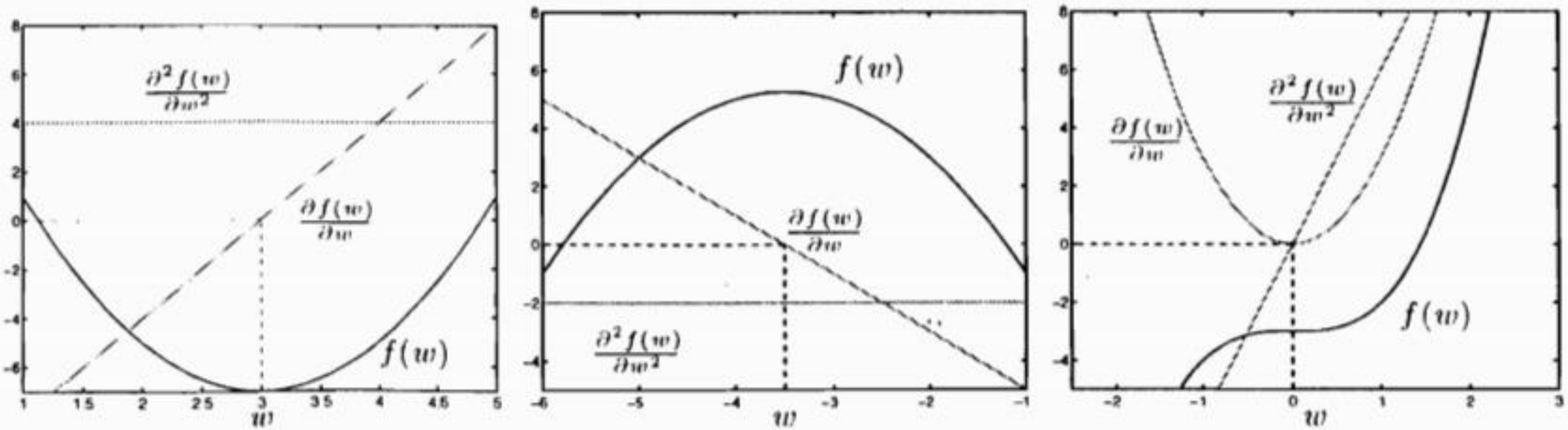
$$\log(L) = \sum_{n=1}^{N} \log\left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(t_n - \boldsymbol{w}^T \boldsymbol{x}_n)^2\right\}\right]$$

which can be simplified to:

$$\log(L) = -\frac{N}{2}\log(2\pi) - N\log(\sigma) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - \boldsymbol{w}^T \boldsymbol{x}_n)^2$$

Finding function's maximum

- A function's minimum or maximum can be determined where the 1st derivative is zero
  - Local maxima
  - Local minima



Maximum likelihood

- Finding function's maximum

$$\frac{\partial \log(L)}{\partial \boldsymbol{w}} = \frac{\partial}{\partial \boldsymbol{w}}\left[-\frac{N}{2}\log(2\pi) - N\log(\sigma) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2\right]$$

$$\frac{\partial \log(L)}{\partial \boldsymbol{w}} = \frac{1}{\sigma^2}\sum_{n=1}^{N}\boldsymbol{x}_n(t_n - \boldsymbol{x}_n^T\boldsymbol{w})$$

$$\boxed{\boldsymbol{x}_n^T\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{x}_n}$$

$$\frac{\partial \log(L)}{\partial \boldsymbol{w}} = \frac{1}{\sigma^2}\sum_{n=1}^{N}(\boldsymbol{x}_n t_n - \boldsymbol{x}_n\boldsymbol{x}_n^T\boldsymbol{w}) = \boldsymbol{0}$$

$$\frac{\partial \log(L)}{\partial \boldsymbol{w}} = \frac{1}{\sigma^2}(\boldsymbol{X}^T\boldsymbol{t} - \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}) = \boldsymbol{0}$$

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{t}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}$$

- It's exactly the same solution as from least squares
  - Minimizing squared loss = maximizing likelihood

Maximum likelihood

- Finding function's maximum

$$\frac{\partial \log(L)}{\partial\sigma} = \frac{\partial}{\partial\sigma}\left[-\frac{N}{2}\log(2\pi) - N\log(\sigma) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2\right]$$

$$\frac{\partial \log(L)}{\partial\sigma} = -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{n=1}^{N}(t_n - \boldsymbol{x}_n^T\widehat{\boldsymbol{w}})^2 = 0$$

$$\boxed{\boldsymbol{x}_n^T\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{x}_n}$$

$$\widehat{\sigma^2} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \boldsymbol{x}_n^T\widehat{\boldsymbol{w}})^2$$

$$\widehat{\sigma^2} = \frac{1}{N}(\boldsymbol{t} - \boldsymbol{X}\widehat{\boldsymbol{w}})^T(\boldsymbol{t} - \boldsymbol{X}\widehat{\boldsymbol{w}})$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}$$

Model complexity

$$\log(L) = -\frac{N}{2}\log(2\pi) - N\log(\sigma) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2$$

- Consider that:

$$\widehat{\sigma^2} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \boldsymbol{x}_n^T\widehat{\boldsymbol{w}})^2$$

so the log likelihood estimate at maximum is:

$$\log(L) = -\frac{N}{2}\log(2\pi) - N\log(\sigma) - \frac{1}{2\widehat{\sigma^2}}N\widehat{\sigma^2}$$

$$\log(L) = -\frac{N}{2}(1 + \log(2\pi)) - \frac{N}{2}\log\left(\widehat{\sigma^2}\right)$$

Model complexity

- Log likelihood estimate at maximum:

$$\log(L) = -\frac{N}{2}(1 + \log(2\pi)) - \frac{N}{2}\log\left(\widehat{\sigma^2}\right)$$

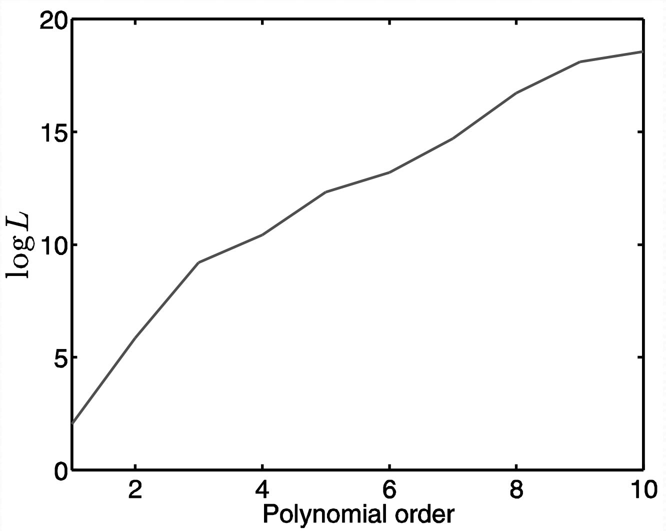
- Decrease in $\widehat{\sigma^2}$ will result in increase in log likelihood

- Note that $\boldsymbol{w}^T\boldsymbol{x}_n$ determines the mean (i.e. trend) and σ^2 determines variance (i.e. noise)

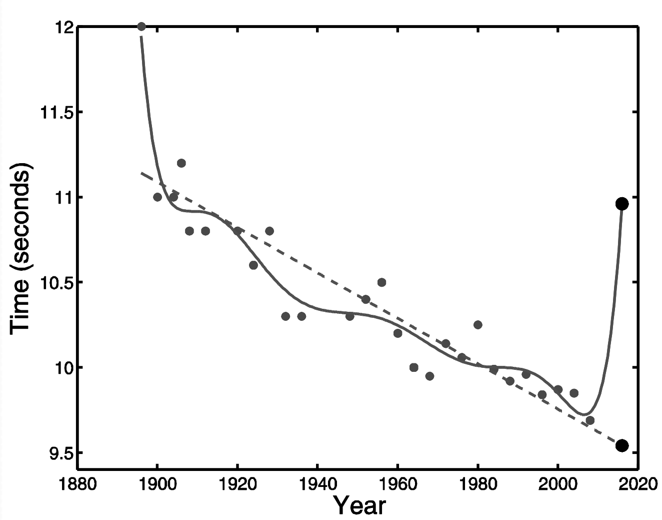
- Decrease in σ^2 will result in those values of model parameters $\boldsymbol{w}$ that capture observations closely
  - i.e. over-fitting and poor generalization

Model complexity

- Modelling Olympics data again..
- Higher model complexity results in maximum likelihood
  - Generalization and over-fitting tradeoff



(a) Increase in log likelihood as the polynomial order increases



(b) 1st and 8th order polynomial functions fitted to the Olympics men's 100 m data. Large dark circles correspond to predictions for the 2016 Olympics

Bias-variance tradeoff

- Generalization and over-fitting tradeoff

- Error between predicted values and observed values is due to bias and model variance:

$$\bar{\mathcal{M}} = \mathcal{B}^2 + \mathcal{V}$$

- $\mathcal{B}^2$ – bias: systematic mismatch between our model and the actual process that generated data
  - Decrease in bias $\mathcal{B}^2$ to control $\bar{\mathcal{M}}$

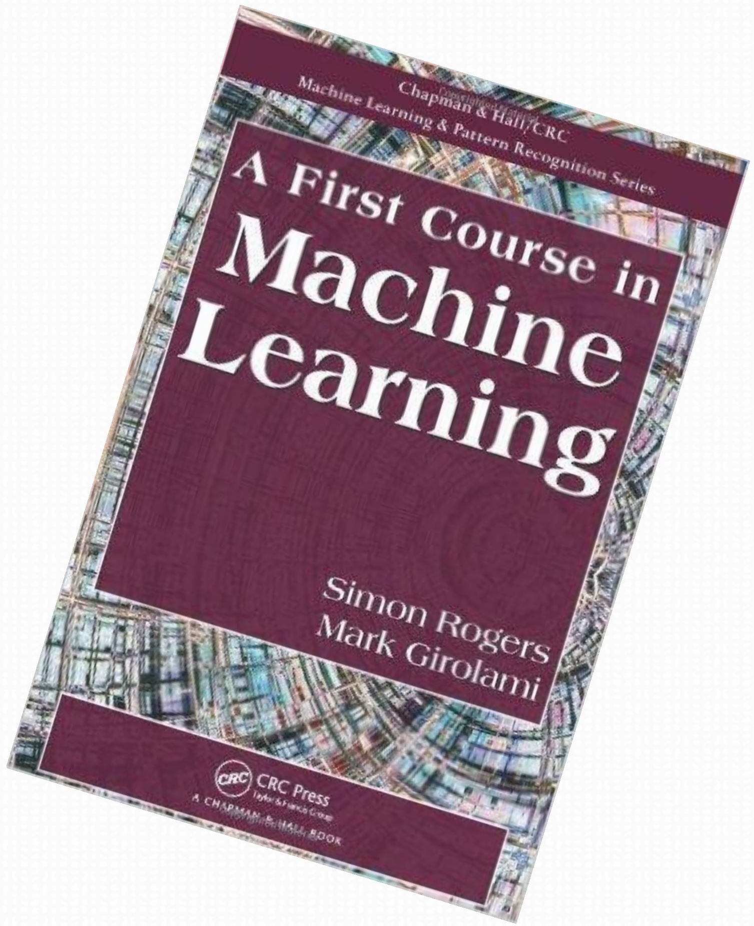
- $\mathcal{V}$ – variance: more complex model has higher variance

Summary

- Explicit modelling of error as noise
- Noise modelling as Gaussian random variable
- Likelihood of model predictions and observed data
- Maximizing the likelihood
- Generalization and over-fitting tradeoff

Exercise (ungraded)

- Book (FCML) – exercise 2.1
- Book (FCML) – exercise 2.8
- Book (FCML) – exercise 2.11
- Book (FCML) – exercise 2.12
- MATLAB code – olymplike.m
- MATLAB code – genolymp.m



Author’s material (Simon Rogers)

That's All

Thank You