Machine Learning, Machine Learning (extended)

2 – Supervised Learning: Linear Modelling by Least Squares

Kashif Rajpoot

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

School of Computer Science

University of Birmingham

Outline

- Linear modelling
  - Least squares
  - Loss function
- Finding function's minimum
- Making predictions from model
- Linear modelling with vectors
- Non-linear response from a linear model
- Generalization and over-fitting
- Cross-validation
- Regularized least squares

Function modelling

- Determine a learner function/model
  - Learn relationship between attributes (i.e. features) and responses or targets (i.e. labels)

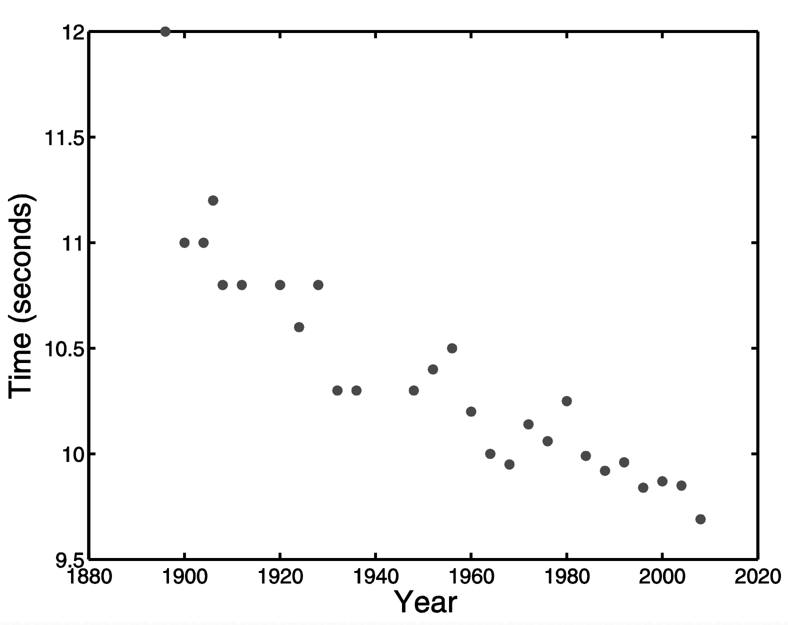
- Examples
  - Disease diagnosis
  - Image classification
  - Face recognition
  - Recommendation system
  - Speaker identification

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?

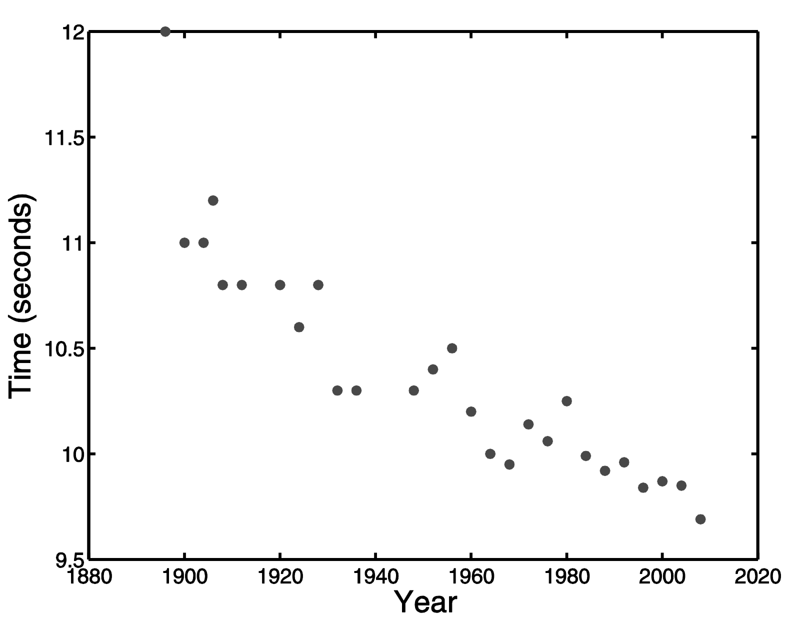
- Can we predict winning time for future games?



Linear modelling

- BIG 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?

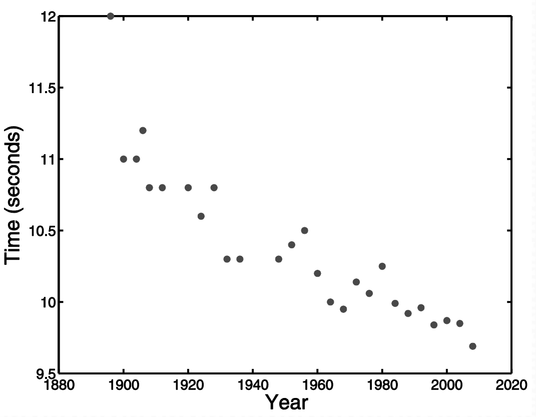


Linear modelling

- Learner model/function
  - Maps input attributes to output response

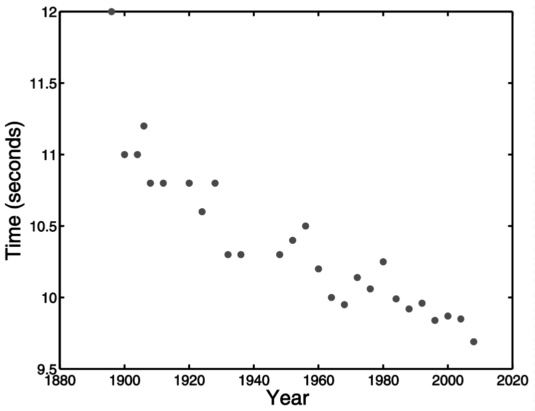
- Let's consider we can predict time $t = f(x)$
  - x?
  - t?

- Training samples
  - N attribute-response pairs $(x_1, t_1), (x_2, t_2), \ldots (x_N, t_N)$



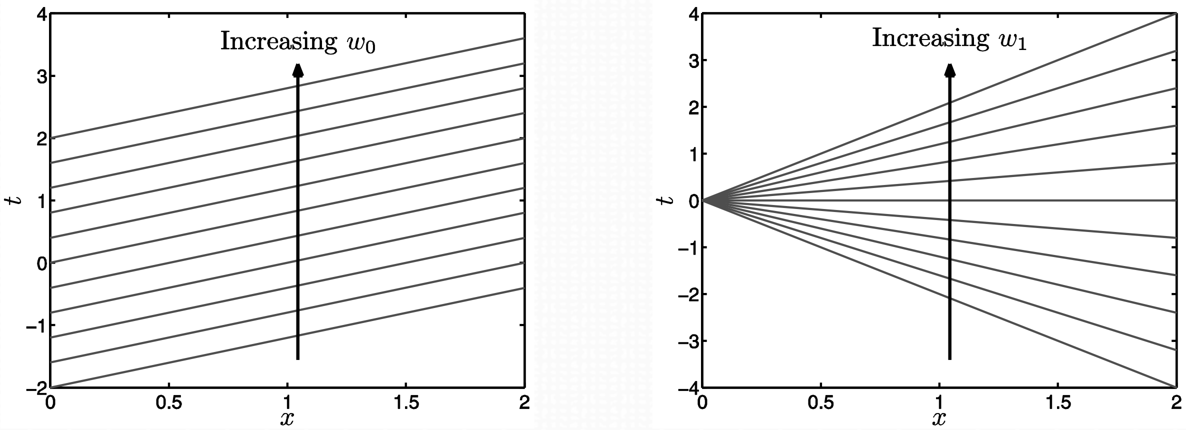
Linear modelling

- Let's consider we can predict time $t = f(x)$
  - Linear function: $f(x) = x$ and $f(x) = mx + c$
  - Non-linear function: $f(x) = sin(x)$
- Can we model Olympic years and winning time with a linear function?
  - i.e. winning time drops by same amount every M years?
  - How about $t = f(x) = x$?
  - How about $t = f(x; w) = wx$?
  - How about $t = f(x; w_0, w_1) = w_0 + w_1 x$?



Linear modelling

$$t = f(x; w_0, w_1) = w_0 + w_1 x$$



(a) Increasing w_0 changes the point at which the line crosses the t axis

(b) Increasing w_1 changes the gradient of the line

- Learner function/model
  - Learn parameters w_0 and w_1 from past data $(x_1, t_1), (x_2, t_2), \ldots (x_N, t_N)$ to predict future winning time

What's a good model?

- Generate a line that passes as close as possible to the past example points
- Loss function: loss between 'ground truth' t_n and model prediction $f(x_n; w_0, w_1)$

$$\mathcal{L}_n(t_n, f(x_n; w_0, w_1)) = (t_n - f(x_n; w_0, w_1))^2$$

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(t_n, f(x_n; w_0, w_1))$$

- Can we use trial and error to find "best" w_0 and w_1?

What's a good model?

- Generate a line that passes as close as possible to the past example points
- Loss function: loss between 'ground truth' t_n and model prediction $f(x_n; w_0, w_1)$

$$\mathcal{L}_n(t_n, f(x_n; w_0, w_1)) = (t_n - f(x_n; w_0, w_1))^2$$

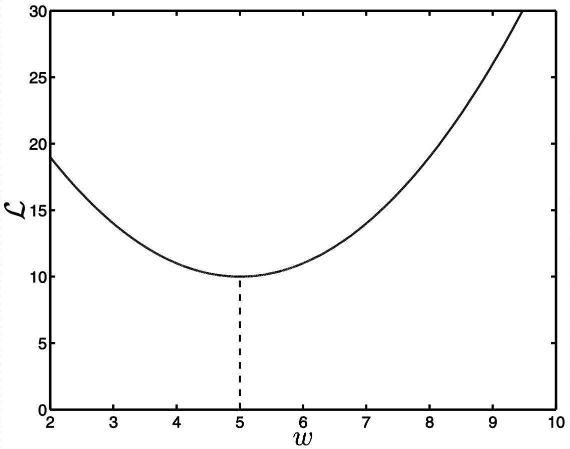
$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_n(t_n, f(x_n; w_0, w_1))$$

- Find "best" w_0 and w_1 that reduces loss $\mathcal{L}$

$$\underset{w_0,w_1}{argmin}\,\mathcal{L}$$

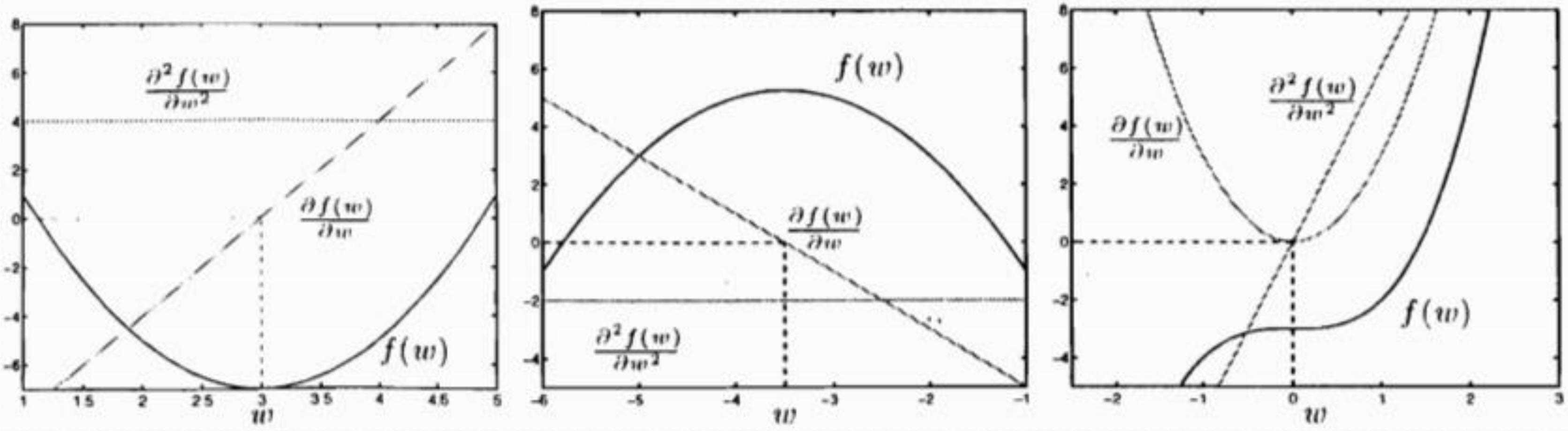
- Minimize least squares error

$$\underset{w_0,w_1}{argmin}\,\tfrac{1}{N}\textstyle\sum_{n=1}^{N}(t_n - f(x_n; w_0, w_1))^2$$



Finding function’s minimum

- A function’s minimum or maximum can be determined where the 1st derivative is zero
  - Local minima
  - Local maxima
- To ensure we have found a function’s minimum, 2nd derivative should be verified to be positive



Linear modelling: least squares solution

- We are aiming to find a functional dependence relationship between Olympics year and winning time

$$f(x; w_0, w_1) = w_0 + w_1 x$$

- We can determine "best" w_0 and w_1 parameters (or weights) that minimize loss function $\mathcal{L}$

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(t_n - (w_0 + w_1 x_n))^2$$

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(w_1^2 x_n^2 + 2w_1 x_n(w_0 - t_n) + w_0^2 - 2w_0 t_n + t_n^2)$$

Finding function's minimum

- Partial derivatives, with respect to w_0 and w_1, at minimum of $\mathcal{L}$ must be 0

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial}{\partial w_0}\left[\frac{1}{N}\sum_{n=1}^{N}(w_1^2 x_n^2 + 2w_1 x_n(w_0 - t_n) + w_0^2 - 2w_0 t_n + t_n^2)\right]$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial}{\partial w_0}\left[w_0^2 + 2w_0 w_1 \frac{1}{N}\sum_{n=1}^{N} x_n - 2w_0 \frac{1}{N}\sum_{n=1}^{N} t_n\right]$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) = 0$$

$$\widehat{w}_0 = \bar{t} - w_1 \bar{x}$$

- w_1?

Finding function's minimum

- Partial derivatives, with respect to w_0 and w_1, at minimum of $\mathcal{L}$ must be 0

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial}{\partial w_1}\left[\frac{1}{N}\sum_{n=1}^{N}(w_1^2 x_n^2 + 2w_1 x_n(w_0 - t_n) + w_0^2 - 2w_0 t_n + t_n^2)\right]$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial}{\partial w_1}\left[w_1^2 \frac{1}{N}\sum_{n=1}^{N} x_n^2 + 2w_1 \frac{1}{N}\sum_{n=1}^{N} x_n(w_0 - t_n)\right]$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right) = 0$$

- w_0?

Finding function's minimum

- Partial derivatives, with respect to w_0 and w_1, at minimum of $\mathcal{L}$ must be 0

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right) = 0$$

$$\boxed{\hat{w}_0 = \bar{t} - w_1\bar{x}}$$

$$2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(\hat{w}_0 - t_n)\right) = 0$$

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

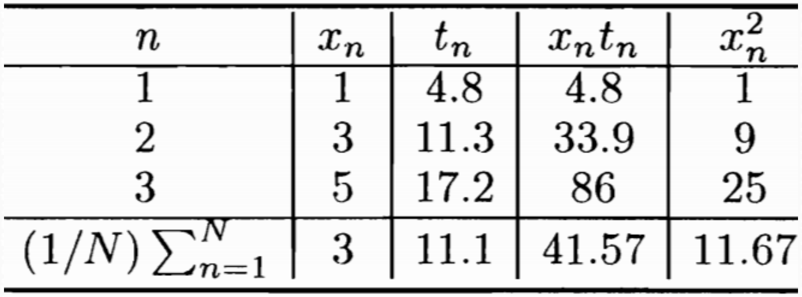
$$\boxed{\hat{w}_0 = \bar{t} - \hat{w}_1\bar{x}}$$

Linear modelling

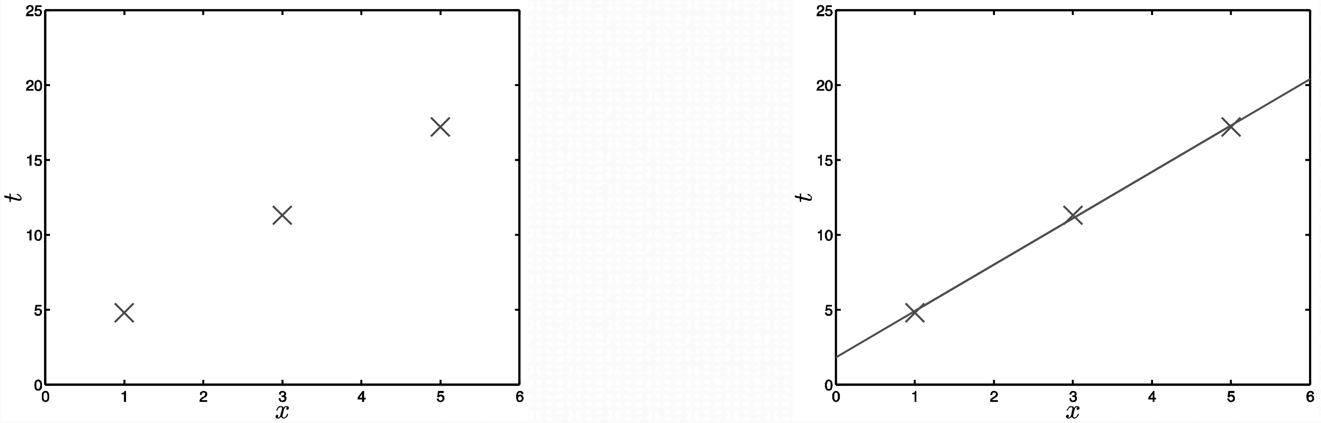
$$\widehat{w}_1 = \frac{\overline{xt} - \bar{x}\bar{t}}{\overline{x^2} - (\bar{x})^2}$$

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

- Compute model parameters from data



n	x_n	t_n	$x_n t_n$	x_n^2
1	1	4.8	4.8	1
2	3	11.3	33.9	9
3	5	17.2	86	25
$(1/N)\sum_{n=1}^{N}$	3	11.1	41.57	11.67



(a) The three synthetic data points described in Table 1.1

(b) The least squares fit defined by $f(x; w_0, w_1) = 1.8 + 3.1x$

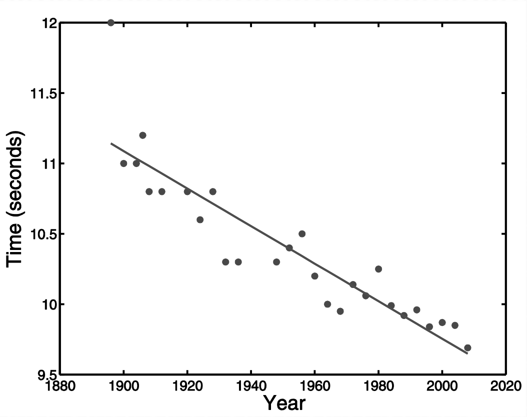
Linear modelling

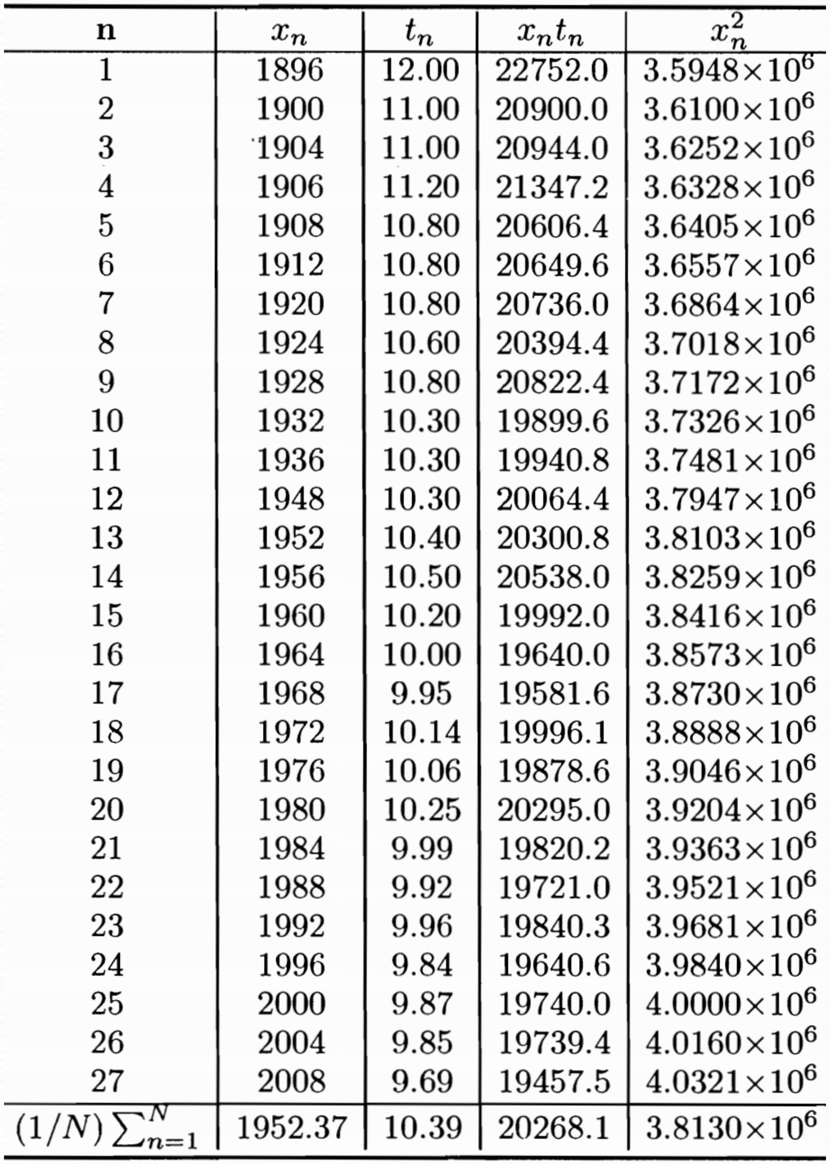
$$\hat{w}_1 = \frac{\overline{xt} - \bar{x}\bar{t}}{\overline{x^2} - (\bar{x})^2}$$

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

- Compute model parameters from Olympics data

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





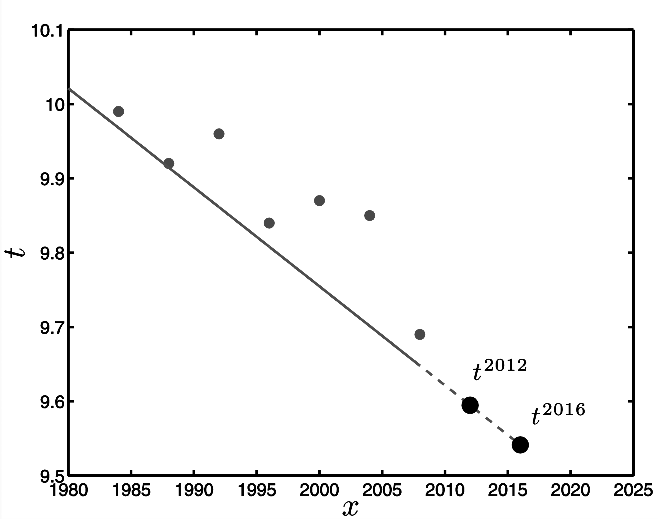
n	x_n	t_n	$x_n t_n$	x_n^2
1	1896	12.00	22752.0	3.5948×10^6
2	1900	11.00	20900.0	3.6100×10^6
3	1904	11.00	20944.0	3.6252×10^6
4	1906	11.20	21347.2	3.6328×10^6
5	1908	10.80	20606.4	3.6405×10^6
6	1912	10.80	20649.6	3.6557×10^6
7	1920	10.80	20736.0	3.6864×10^6
8	1924	10.60	20394.4	3.7018×10^6
9	1928	10.80	20822.4	3.7172×10^6
10	1932	10.30	19899.6	3.7326×10^6
11	1936	10.30	19940.8	3.7481×10^6
12	1948	10.30	20064.4	3.7947×10^6
13	1952	10.40	20300.8	3.8103×10^6
14	1956	10.50	20538.0	3.8259×10^6
15	1960	10.20	19992.0	3.8416×10^6
16	1964	10.00	19640.0	3.8573×10^6
17	1968	9.95	19581.6	3.8730×10^6
18	1972	10.14	19996.1	3.8888×10^6
19	1976	10.06	19878.6	3.9046×10^6
20	1980	10.25	20295.0	3.9204×10^6
21	1984	9.99	19820.2	3.9363×10^6
22	1988	9.92	19721.0	3.9521×10^6
23	1992	9.96	19840.3	3.9681×10^6
24	1996	9.84	19640.6	3.9840×10^6
25	2000	9.87	19740.0	4.0000×10^6
26	2004	9.85	19739.4	4.0160×10^6
27	2008	9.69	19457.5	4.0321×10^6
$(1/N)\sum_{n=1}^{N}$	1952.37	10.39	20268.1	3.8130×10^6

Making predictions from model

- Predict winning time for years 2012 and 2016

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

- Accurate & precise prediction?
  - Predicting past seen examples?
  - Predicting future unseen examples?



- Basis of prediction?

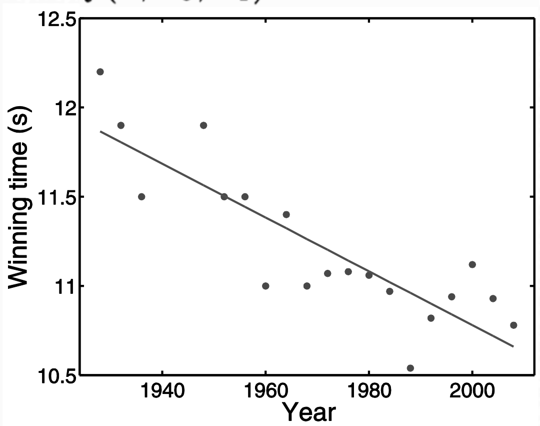
Making predictions from model

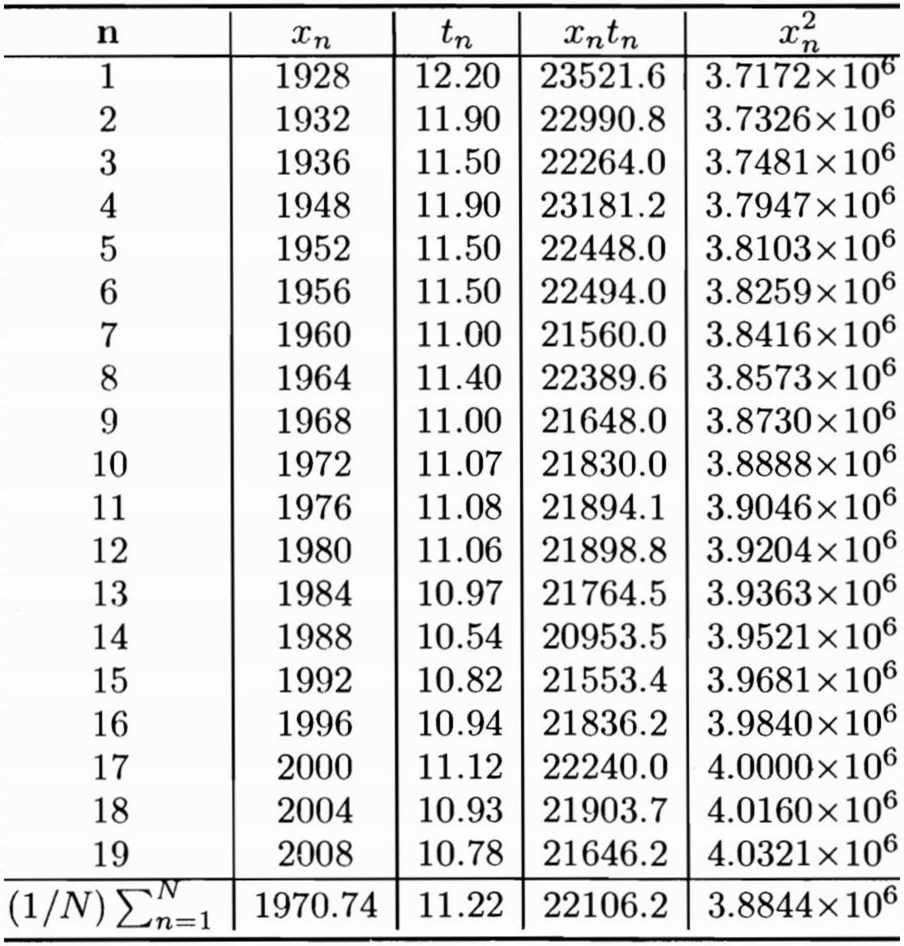
$$\hat{w}_1 = \frac{\overline{xt} - \bar{x}\bar{t}}{\overline{x^2} - (\bar{x})^2}$$

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

- Compute model parameters from Olympics women's 100m data

$f(x; w_0, w_1) = 40.92 - 0.015x$





n	x_n	t_n	$x_n t_n$	x_n^2
1	1928	12.20	23521.6	3.7172×10^6
2	1932	11.90	22990.8	3.7326×10^6
3	1936	11.50	22264.0	3.7481×10^6
4	1948	11.90	23181.2	3.7947×10^6
5	1952	11.50	22448.0	3.8103×10^6
6	1956	11.50	22494.0	3.8259×10^6
7	1960	11.00	21560.0	3.8416×10^6
8	1964	11.40	22389.6	3.8573×10^6
9	1968	11.00	21648.0	3.8730×10^6
10	1972	11.07	21830.0	3.8888×10^6
11	1976	11.08	21894.1	3.9046×10^6
12	1980	11.06	21898.8	3.9204×10^6
13	1984	10.97	21764.5	3.9363×10^6
14	1988	10.54	20953.5	3.9521×10^6
15	1992	10.82	21553.4	3.9681×10^6
16	1996	10.94	21836.2	3.9840×10^6
17	2000	11.12	22240.0	4.0000×10^6
18	2004	10.93	21903.7	4.0160×10^6
19	2008	10.78	21646.2	4.0321×10^6
$(1/N)\sum_{n=1}^{N}$	1970.74	11.22	22106.2	3.8844×10^6

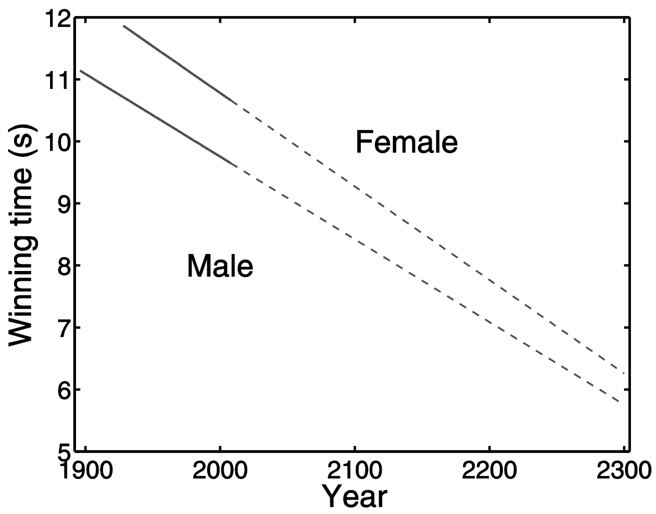
Making predictions from model

- Linear model from men’s 100m data

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

- Linear model from women’s 100m data

$$f(x; w_0, w_1) = 40.92 - 0.015x$$



- Accurate & precise prediction?
  - Intersection of lines?
  - Distance between predicted and observed points?
  - Approaching zero?

Linear modelling

- What if we want to learn linear model from data having more than one attribute?
  - Olympics year (x)
  - Each athlete's personal best in lanes 1 to 8 (s_n)
- Linear model can then be represented as:

$$t = f(x, s_1, \ldots, s_8; w_0, \ldots, w_9)$$

$$\begin{aligned} t &= w_0 + w_1 x + w_2 s_1 + w_3 s_2 + w_4 s_3 \\ &\quad + w_5 s_4 + w_6 s_5 + w_7 s_6 + w_8 s_7 + w_9 s_8 \end{aligned}$$

- How to find the parameters w_n?
  - Derive loss function $\mathcal{L}$?
  - Take partial derivative with respect to each w_n?

Linear modelling with vectors

- Let's consider our model

$$t = w_0 + w_1 x$$

- We can write parameters and attributes in vector form:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad \mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix}$$

- The model in equivalent vector notation:

$$t = f(x_n; w_0, w_1) = \mathbf{w}^{\mathsf{T}} \mathbf{x}_n = w_0 + w_1 x_n$$

- Thus

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} (t_n - (w_0 + w_1 x_n))^2$$

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

becomes

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} (t_n - \mathbf{w}^{\mathsf{T}} \mathbf{x}_n)^2 = \frac{1}{N} (\mathbf{t} - \mathbf{X}\mathbf{w})^{\mathsf{T}} (\mathbf{t} - \mathbf{X}\mathbf{w})$$

$$= \frac{1}{N} (\mathbf{X}\mathbf{w} - \mathbf{t})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{t})$$

Linear modelling with vectors

- Finding loss function's minimum

$$\mathcal{L} = \frac{1}{N}(\mathbf{X}\mathbf{w} - \mathbf{t})^{\mathsf{T}}(\mathbf{X}\mathbf{w} - \mathbf{t})$$

$$\mathcal{L} = \frac{1}{N}\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathsf{T}}\mathbf{t}$$

- Obtain partial derivatives

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_0} \\ \frac{\partial \mathcal{L}}{\partial w_1} \end{bmatrix}$$

Linear modelling with vectors

- Obtain partial derivatives of loss function $\mathcal{L}$

$$\mathcal{L} = \frac{1}{N}(\mathbf{w}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{w} - 2\mathbf{w}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{t} + \mathbf{t}^\mathsf{T}\mathbf{t})$$

Exercise 1.3

$$\frac{1}{N}\mathbf{w}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{w} = w_0^2\frac{1}{N}\left(\sum_{n=1}^{N} X_{n0}^2\right) + 2w_0w_1\frac{1}{N}\left(\sum_{n=1}^{N} X_{n0}X_{n1}\right) + w_1^2\frac{1}{N}\left(\sum_{n=1}^{N} X_{n1}^2\right)$$

$$2\mathbf{w}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{t} = 2w_0\frac{1}{N}\left(\sum_{n=1}^{N} X_{n0}t_n\right) + 2w_1\frac{1}{N}\left(\sum_{n=1}^{N} X_{n1}t_n\right)$$

- Considering $X_{n0} = 1$ and $X_{n0} = x_n$, we get:

$$\mathcal{L} = w_0^2 + 2w_0w_1\frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) + w_1^2\frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) - 2w_0\frac{1}{N}\left(\sum_{n=1}^{N} t_n\right) - 2w_1\frac{1}{N}\left(\sum_{n=1}^{N} x_nt_n\right)$$

Linear modelling with vectors

- Obtain partial derivatives of loss function $\mathcal{L}$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1\bar{x} - 2\bar{t}$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_0\bar{x} + 2w_1\overline{x^2} - 2\overline{xt}$$

which is equivalent to our earlier derivation:

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1\frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1\frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)$$

Linear modelling with vectors

- Obtain partial derivatives of loss function $\mathcal{L}$, with vector notation:

$$\mathcal{L} = \frac{1}{N}\mathbf{w}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{t} + \frac{1}{N}\mathbf{t}^\mathsf{T}\mathbf{t}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{2}{N}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{X}^\mathsf{T}\mathbf{t} = 0$$

$$\widehat{\mathbf{w}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{t}$$

$f(\mathbf{w})$	$\frac{\partial f}{\partial \mathbf{w}}$
$\mathbf{w}^\mathsf{T}\mathbf{x}$	$\mathbf{x}$
$\mathbf{x}^\mathsf{T}\mathbf{w}$	$\mathbf{x}$
$\mathbf{w}^\mathsf{T}\mathbf{w}$	$2\mathbf{w}$
$\mathbf{w}^\mathsf{T}\mathbf{C}\mathbf{w}$	$2\mathbf{C}\mathbf{w}$

- To predict the winning time for a new vector of attributes:

$$t_{\text{new}} = \widehat{\mathbf{w}}^\mathsf{T}\mathbf{x}_{\text{new}}$$

Nonlinear response from a linear model

- Linear model is far too simplistic – it predicts winning time of -3.5 seconds in year 3000
- Linear model $f(x; \boldsymbol{w}) = w_0 + w_1 x$ is linear in both parameters (w) and data (x)
    - Linearity in parameters is useful to obtain analytical solution we have seen earlier
- Nonlinearity in data

$$\mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \\ x_n^2 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix}$$

and determining parameters: $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$

Nonlinear response from a linear model

- The model now becomes:

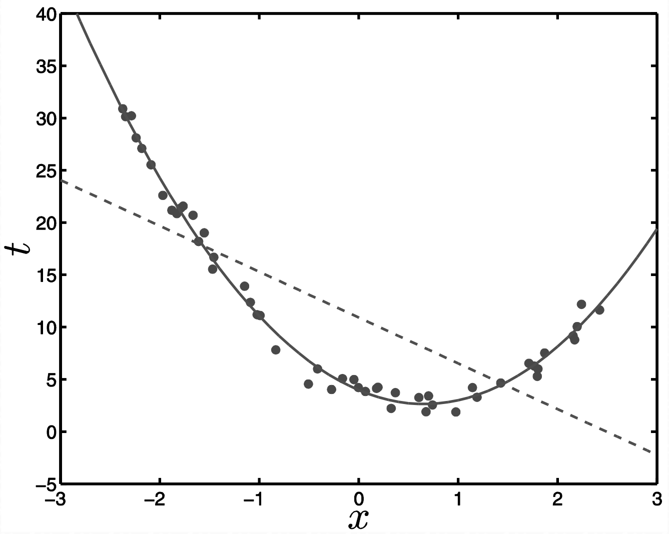
$$f(x;\mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x} = w_0 + w_1 x + w_2 x^2$$

which is linear in parameters ($\boldsymbol{w}$)

- The parameters $\hat{\boldsymbol{w}}$ can be determined like before:

$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{t}$$

- Quadratic data
  - Quadratic fit
  - Linear fit



Nonlinear response from a linear model

- In general, data can be represented as a “polynomial” function of any order:

$$\mathbf{X} = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^K \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^K \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_N^0 & x_N^1 & x_N^2 & \cdots & x_N^K \end{bmatrix}$$

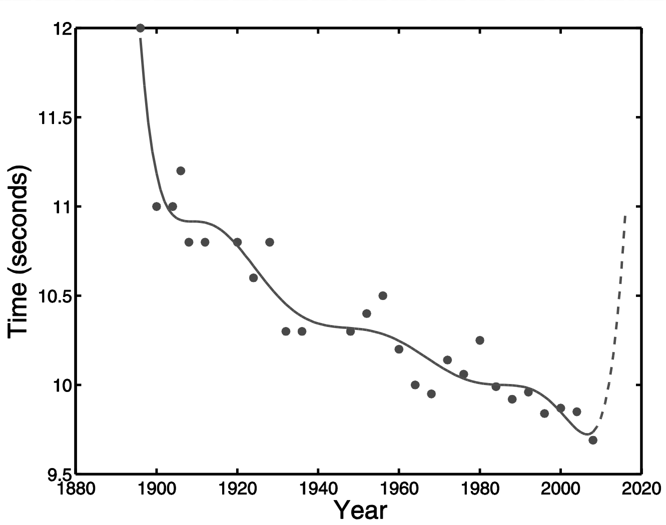
such that the model is: $f(x;\mathbf{w}) = \sum_{k=0}^{K} w_k x^k$

while the parameters can be computed as before:

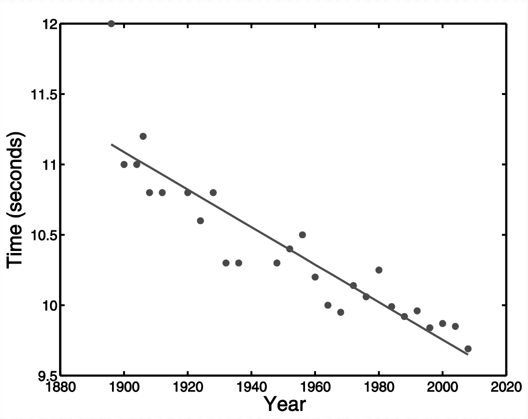
$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{t}$$

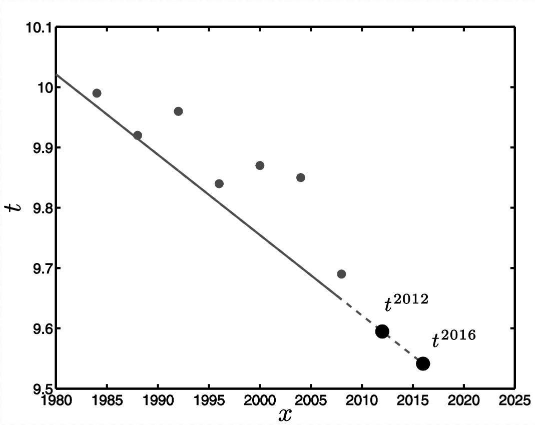
Nonlinear response from a linear model

- 8th order polynomial fit on Olympic men's 100m data
- Model selection: which model is better?
    - What is "better"?
    - $\mathcal{L}^8 = 0.459$, $\mathcal{L}^1 = 1.358$
- Fitting vs prediction



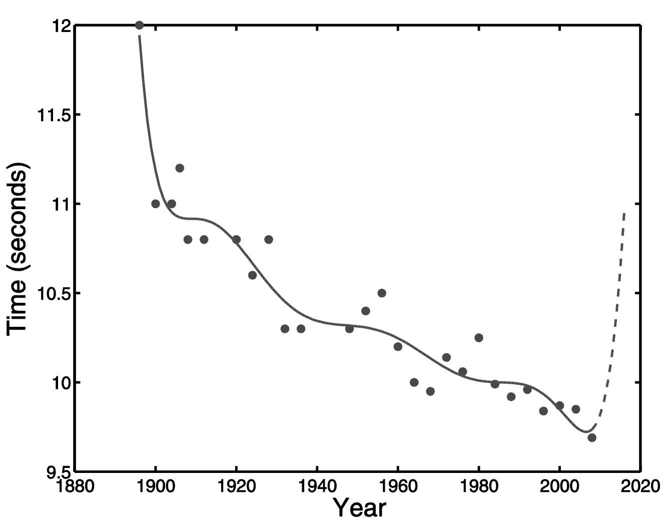
$$f(x; w_0, w_1) = 36.416 - 0.013x$$





Generalization and over-fitting

- Learning aims to build model from past examples in order to predict future examples
  - What is a "good" model?
  - A good model should *generalize* beyond *training* examples – i.e. minimize loss on unseen data
  - Do we have unseen data available?
- Over-fitting
  - Model very closely fits to the training data (observed data)
  - Model does not generalize well to unseen data
- Model complexity
  - More complex models typically have poor generalization



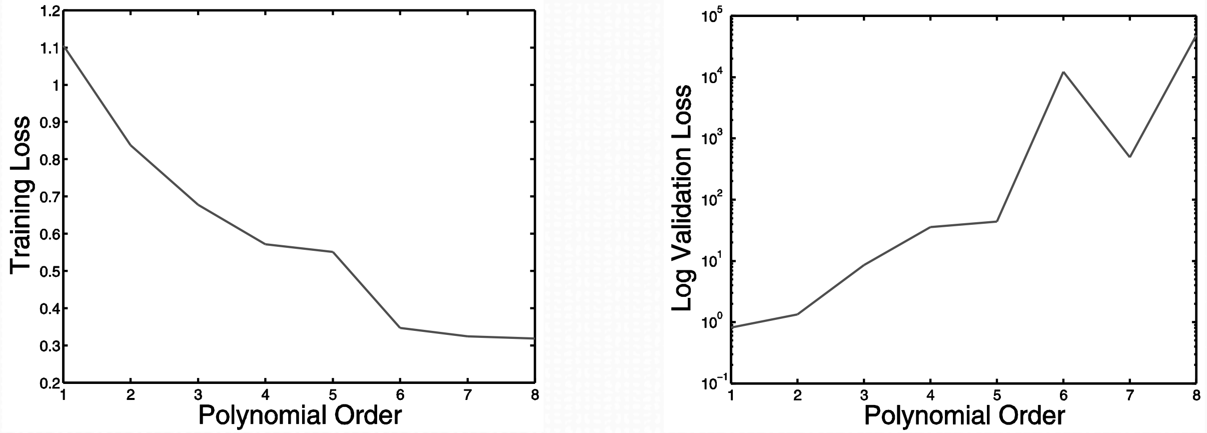
Validation data

- Do we have unseen data available?
  - Validation data can be used to validate the predictive performance of a model

- Where to get validation data from?
  - Retain a “proportion” from the available data
  - For example: (i) train the model on 1980 and before Olympics data, (ii) ask the model to predict post-1980 winning time, and (iii) compute the validation loss on this “unseen” data.
  - Lowest validation loss could help select “optimal” model

Generalization and over-fitting

- Monotonic decrease in training loss
- Validation loss
- Model complexity
- What is a "good" model?

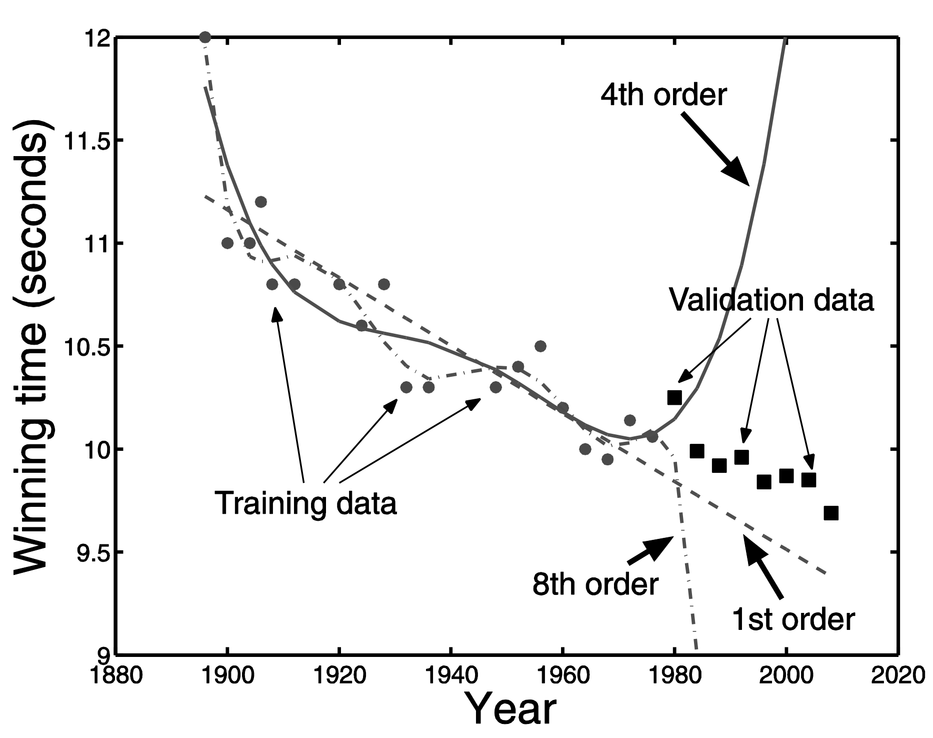


(a) Training loss for the Olympics men's 100 m data

(b) Log validation loss for the Olympics men's 100 m data. When using the squared loss, this is also known as the squared predictive error and measures how close the predicted values are to the true values. Note that the log loss is plotted as the value increases so rapidly

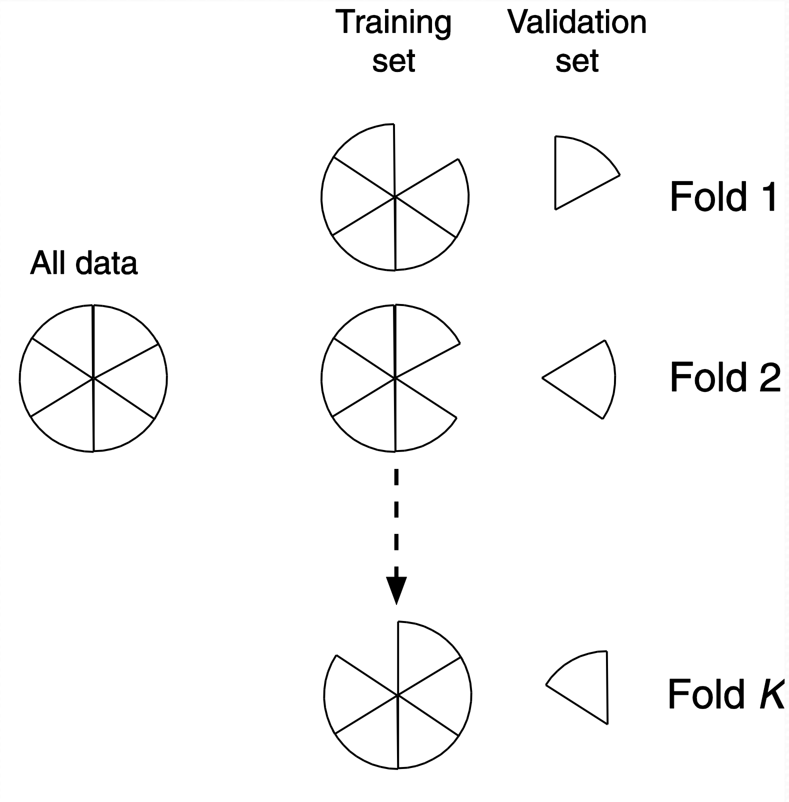
Generalization and over-fitting

- Model complexity
- What is a "good" model?



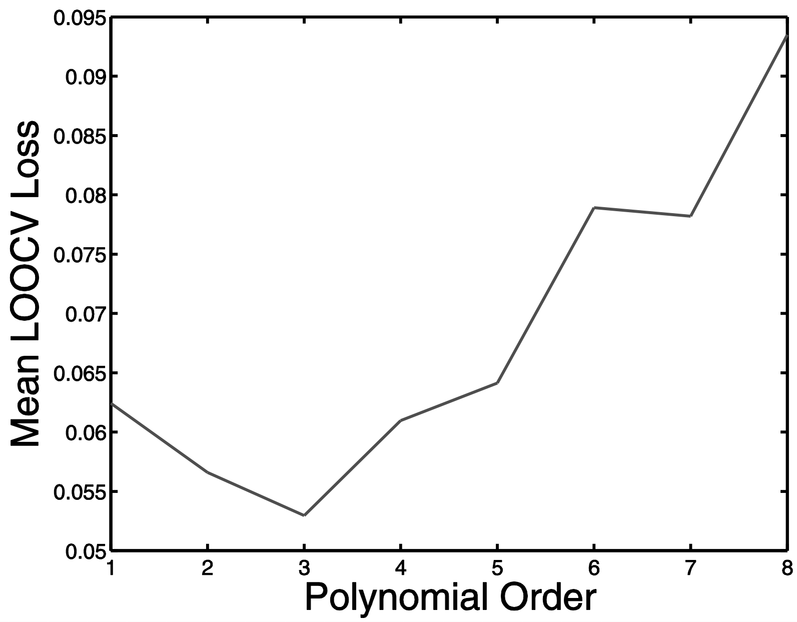
Validation data

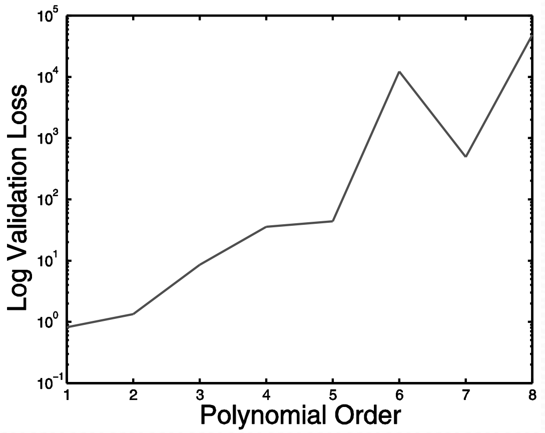
- Validation is biased towards choice of data in validation set, particularly if data is small
- K-fold cross-validation



Cross-validation

- Average K-fold cross validation loss
- Leave one out (LOO) cross validation (LOOCV)
  - Extreme case of K fold cross validation where K = N
  - LOOCV loss $\mathcal{L}^{LOOCV} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \hat{\boldsymbol{w}}_{-n}^{T}\boldsymbol{x}_n)^2$
- Cross-validation loss vs validation loss
- Model selection is difficult





Regularized least squares

- In model learning, the objective is to ensure good generalization and prevent over-fitting (i.e. avoid model complexity)
  - Regularization is a way of achieving this objective
- Let's take a very simple model:

$$f(x;\boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + w_5 x^5$$

by considering that $\boldsymbol{w} = [0,0,\dots,0]^T$

- Now assign non-zero parameter values one-by-one to each component of $\boldsymbol{w}$, this gradually increases model complexity
- The model complexity (thus possibly over-fitting) increases with an increase in $\boldsymbol{w}$ magnitude

Regularized least squares

- The increase in model complexity can be "controlled" by "controlling" $\boldsymbol{w}$

$$\sum_i w_i^2$$

or

$$\boldsymbol{w}^T\boldsymbol{w}$$

- Thus, a model should aim to minimize loss while simultaneously penalizing over-complexity

$$\mathcal{L}' = \mathcal{L} + \lambda\boldsymbol{w}^T\boldsymbol{w}$$

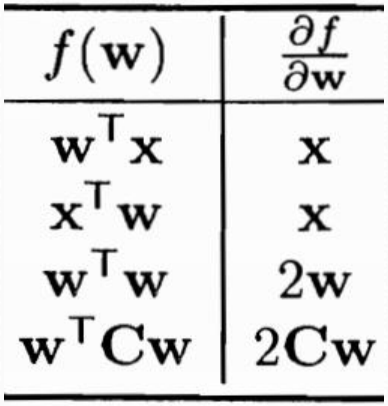
$$\mathcal{L}' = \frac{1}{N}\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathsf{T}}\mathbf{t} + \lambda\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

Regularized least squares

- Finding loss function's minimum

$$\mathcal{L}' = \frac{1}{N}\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathsf{T}}\mathbf{t} + \lambda\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

$$\frac{\partial\mathcal{L}'}{\partial\mathbf{w}} = \frac{2}{N}\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{X}^{\mathsf{T}}\mathbf{t} + 2\lambda\mathbf{w} = 0$$



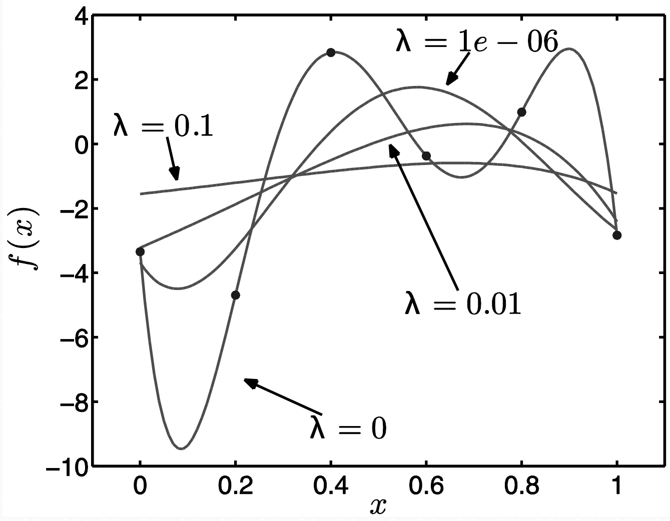
$f(\mathbf{w})$	$\frac{\partial f}{\partial \mathbf{w}}$
$\mathbf{w}^{\mathsf{T}}\mathbf{x}$	$\mathbf{x}$
$\mathbf{x}^{\mathsf{T}}\mathbf{w}$	$\mathbf{x}$
$\mathbf{w}^{\mathsf{T}}\mathbf{w}$	$2\mathbf{w}$
$\mathbf{w}^{\mathsf{T}}\mathbf{C}\mathbf{w}$	$2\mathbf{C}\mathbf{w}$

- The parameters for a regularized model of the data can be obtained as:

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X} + N\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T t$$

- How to determine λ?

- Over-fitting/generalization trade-off

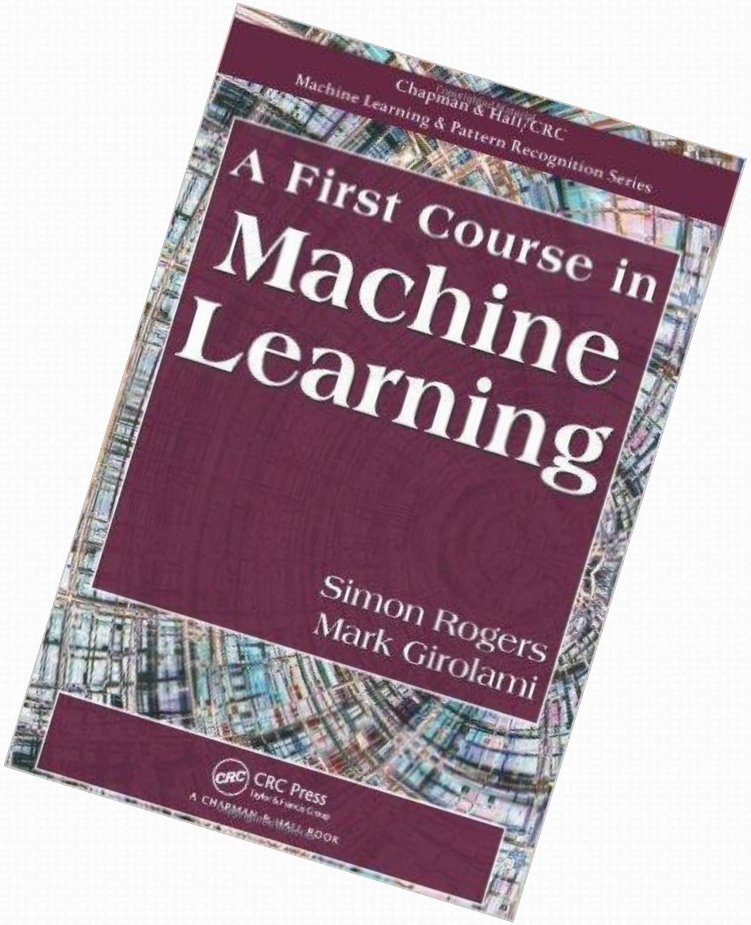


Summary

- Making predictions with linear model
- Loss estimated from training samples isn't reliable
- Generalization
- Model complexity
- Cross-validation
- Non-linear predictions from linear model

Exercise (ungraded)

- Book (FCML) – exercise 1.1
- Book (FCML) – exercise 1.3
- Book (FCML) – exercise 1.4
- Book (FCML) – exercise 1.6 (use vector notation and MATLAB)
- Book (FCML) – exercise 1.7
- Book (FCML) – exercise 1.8
- Book (FCML) – exercise 1.10


Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series

A First Course in Machine Learning

Simon Rogers
Mark Girolami

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Author’s material
(Simon Rogers)

That's All

Thank You