Machine Learning
CS342

Lecture 2: Linear Regression with Ordinary Least Squares (OLS)

Dr. Theo Damoulas
T.Damoulas@warwick.ac.uk

Office hours: Mon & Fri 10-11am @ CS 307

Department of COMPUTER SCIENCE

Recap: ML subfields

Machine Learning

Supervised Learning
- Regression ← **We are here**
- Classification
- Ranking
- Structured Prediction

$\mathcal{D} = \{\mathbf{X}, \mathbf{t}\}, \quad e.g. \quad \mathbf{X} \in \Re^{N \times D}, \mathbf{t} \in \Re^{N}$

Unsupervised Learning
- Clustering
- Dimensionality Reduction $\quad \mathcal{D} = \{\mathbf{X}\}, \quad \mathbf{X} \in \Re^{N \times D}$
- Manifold Learning

Reinforcement Learning
- Markov Decision Process (e.g. Robotics)
- Multi-Agent Systems $\quad \mathcal{D} = \{\mathbf{S}, \mathbf{A}, \mathbf{R}, \pi^*, \mathbf{V}\}$
- TD/Q-Learning (State-Action-Reward)

Recap: Definition of "learning from data"

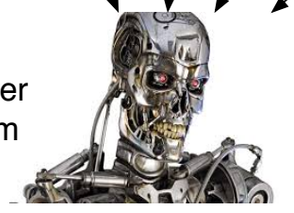
- Study of systems and algorithms that "**learn from data**"

- *A computer program is said to learn from experience **E* (data)** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.* [Tom Mitchell, 1998]

E: Data on Sarah

Learning Process

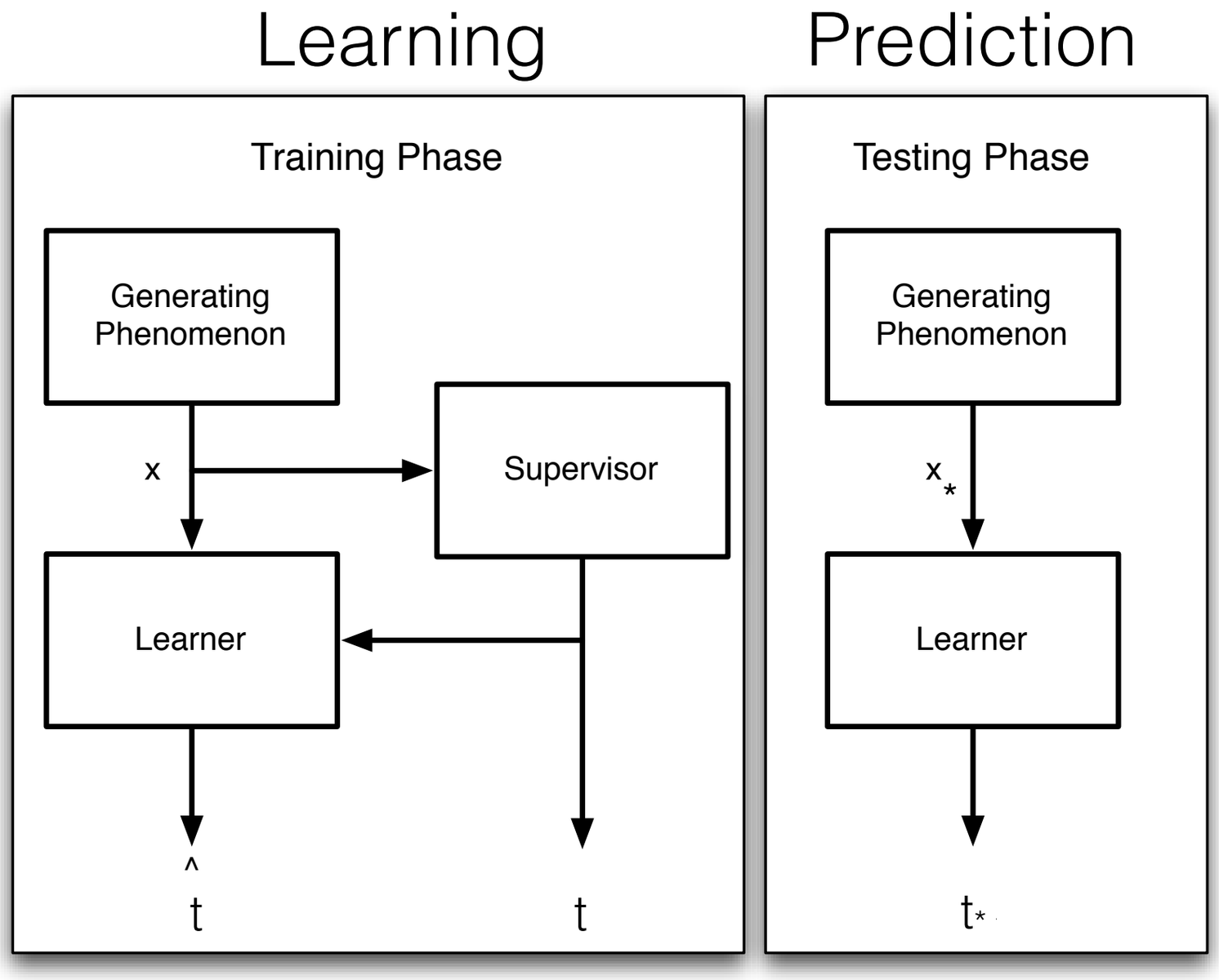
Computer Program



T: Find Sarah

P: % Correct recognitions

Recap: Schematic of Learning (Training) and Prediction phase



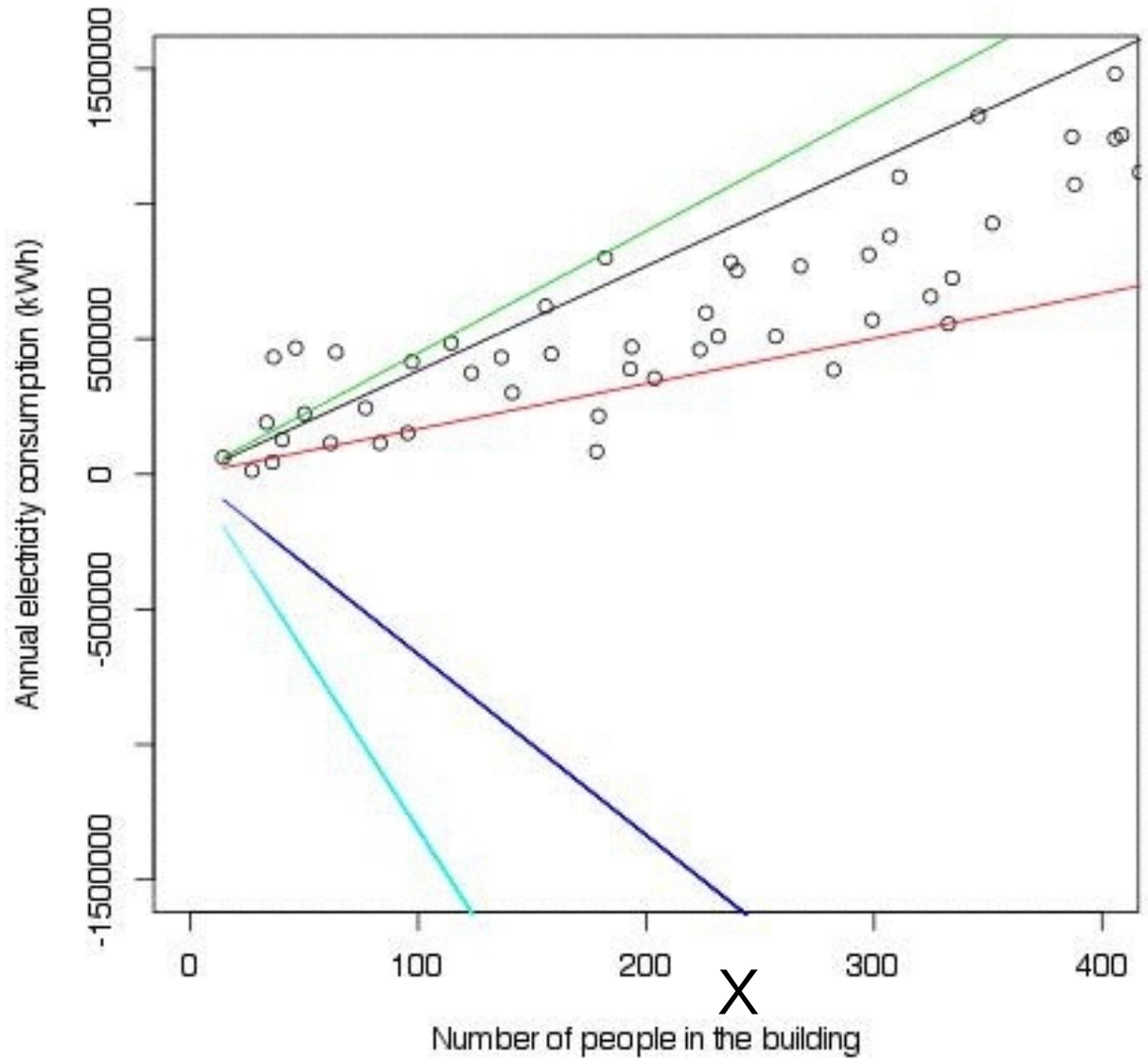
$$D_{\text{training}} = \{\mathbf{X}, \mathbf{t}\} = \{\mathbf{x}_n, t_n\}_{n=1}^{N} \qquad D_{\text{testing}} = \{\mathbf{X}_*\}$$

Recap: Linear Regression

Regression:

t is continuous

t



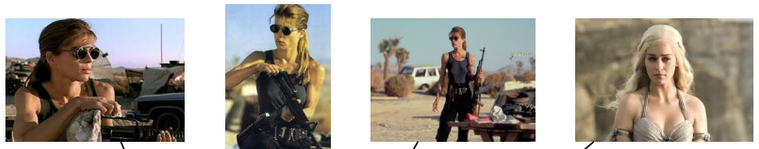
Hypothesis: The relationship between x and t is **linear**

$$\hat{t} = f(x; w_0, w_1) = w_0 + w_1 x$$

Department of COMPUTER SCIENCE

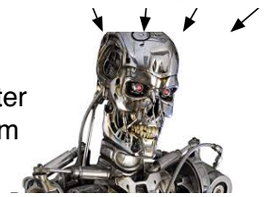
Recap: Supervised Learning components

E: Data on Sarah



Learning Process

Computer Program



T: Find Sarah

P: % Correct recognitions

Supervised Learning Process

Data
(inputs **X** / outputs **t**)

Hypothesis Space
(e.g. all lines in 2D)
Functions that map inputs to outputs

$$f(\mathbf{x};\mathbf{w}) : \mathbf{x} \rightarrow t$$

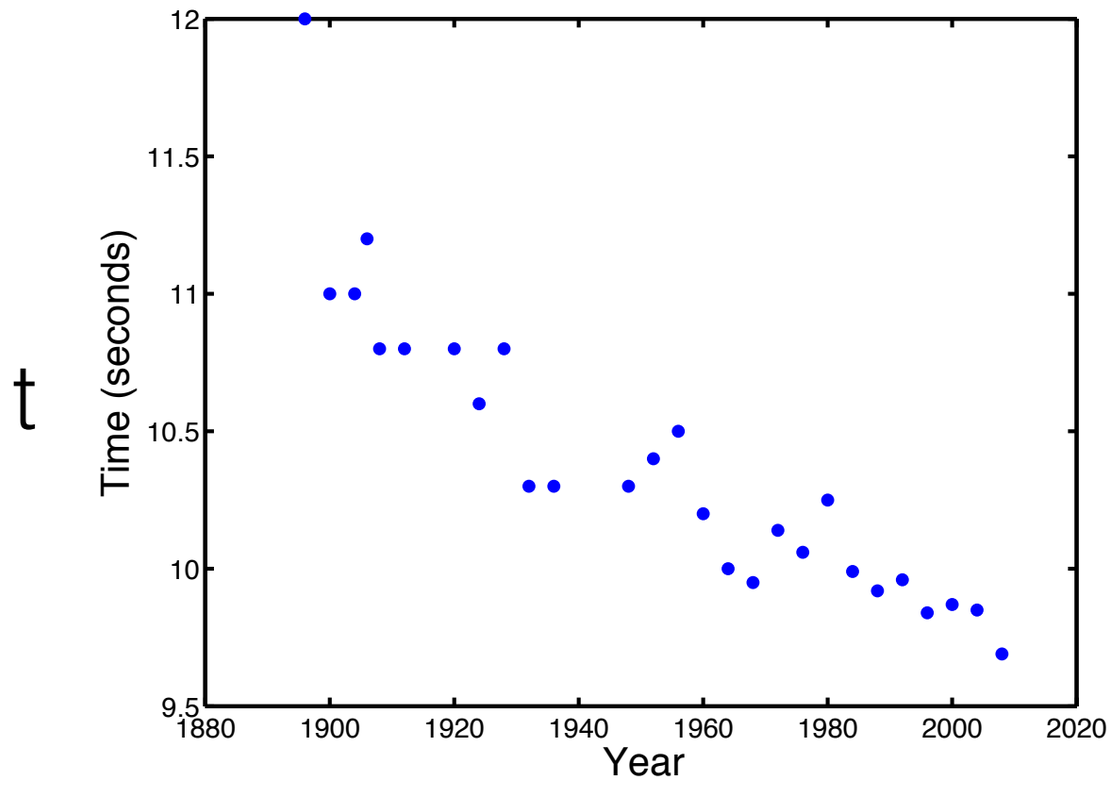
"Goodness" Metric
(e.g. a distance of f(x;w) to t)

Learning Program

Parameters $\mathbf{w}$ (choose a specific hypothesis/mapping)

Learning as parameter (hypothesis) inference

Revisit Olympics example: Univariate regression



Rogers & Girolami Ch. 1

t

X

Learn a mapping (function) of inputs x to outputs t

$$f(x; \mathbf{w}) : x \rightarrow t$$

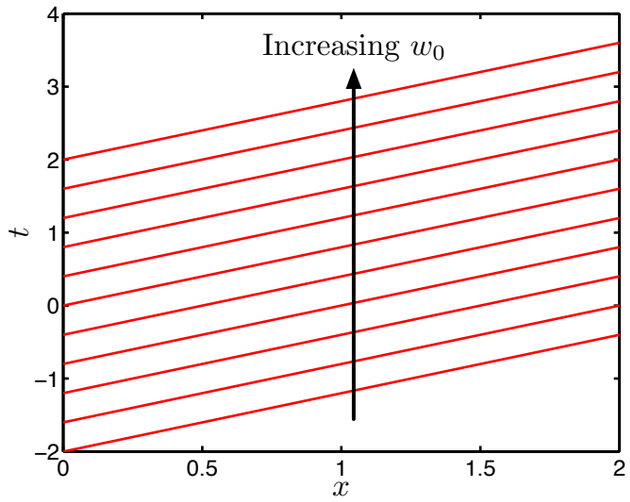
Hypothesis: linear relationship. So our hypothesis space is all lines in this space:

$$\hat{t} = f(x; \mathbf{w}) = w_0 + w_1 x$$

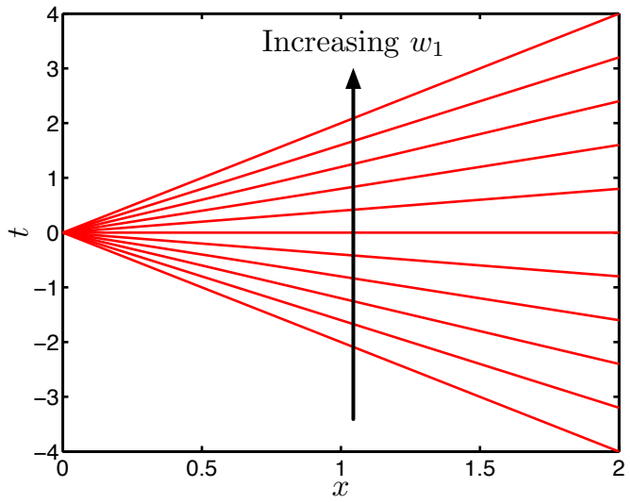
What is a good model?

$$\hat{t} = f(x; \mathbf{w}) = w_0 + w_1 x$$

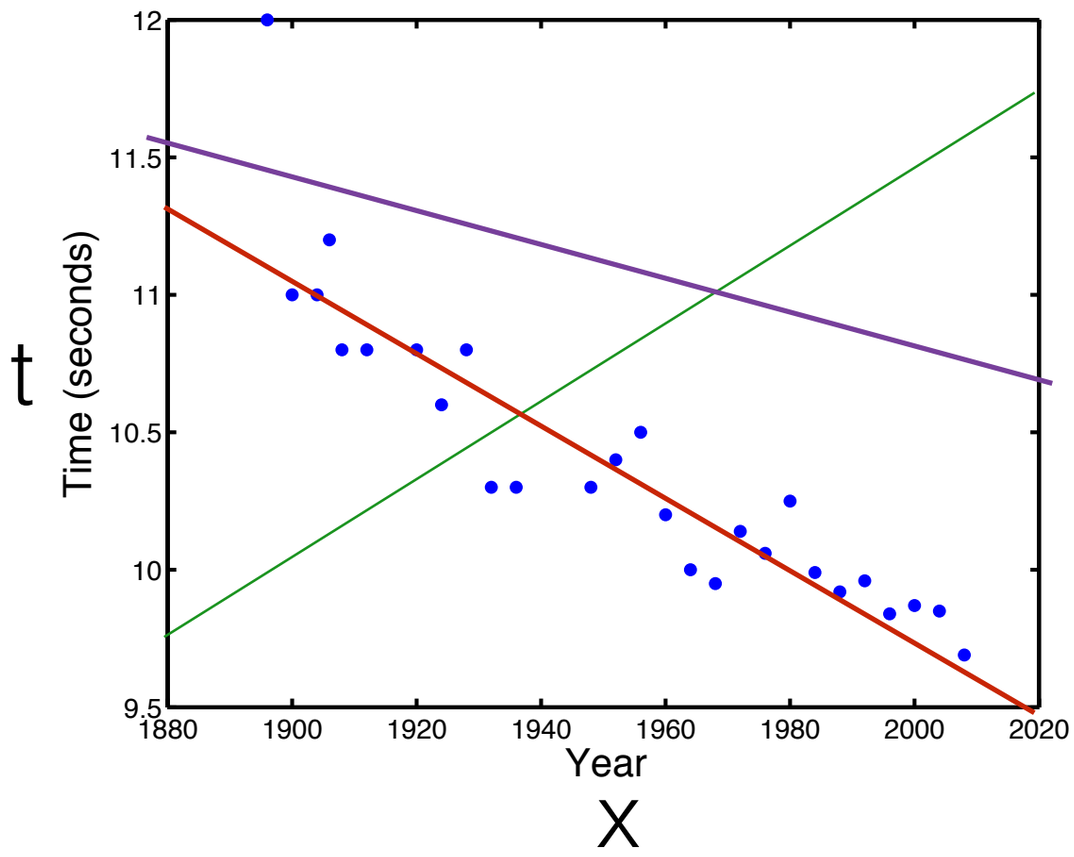
Different parameters
give us different lines/hypothesis



(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.



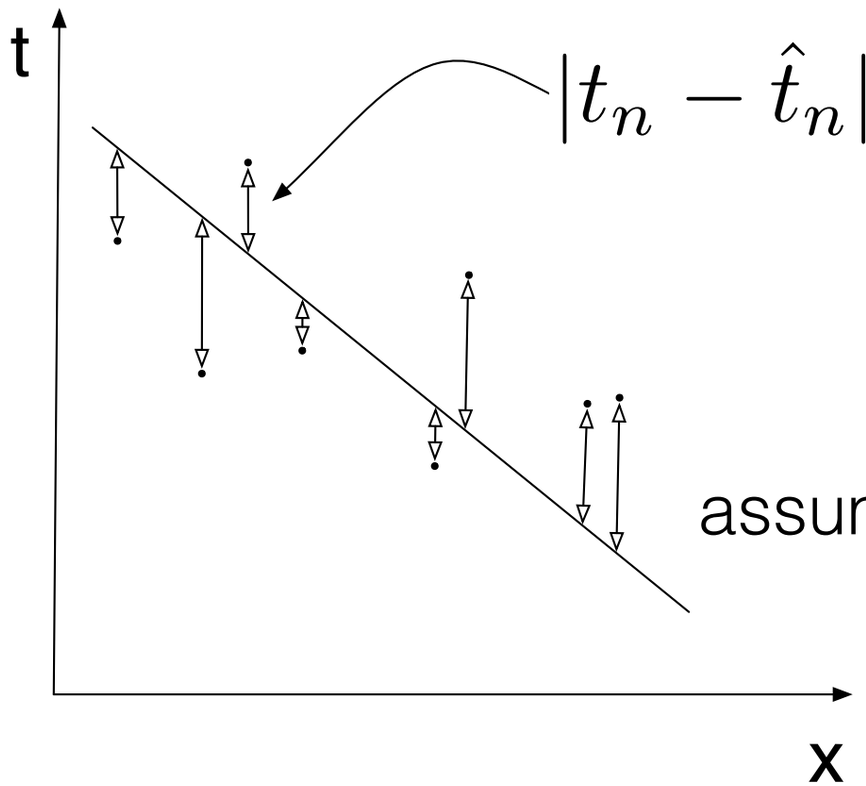
Ok so how do we find the right parameters?
What is one possible measure of “goodness”?

Loss functions

$$\hat{t} = f(x; \mathbf{w}) = w_0 + w_1 x$$

Line that passes as **close** as possible to all the points

n^{th} observation has $\{x_n, t_n\}$



$$|t_n - \hat{t}_n| = |t_n - f(x_n; \mathbf{w})|$$

assume "error" is in t direction

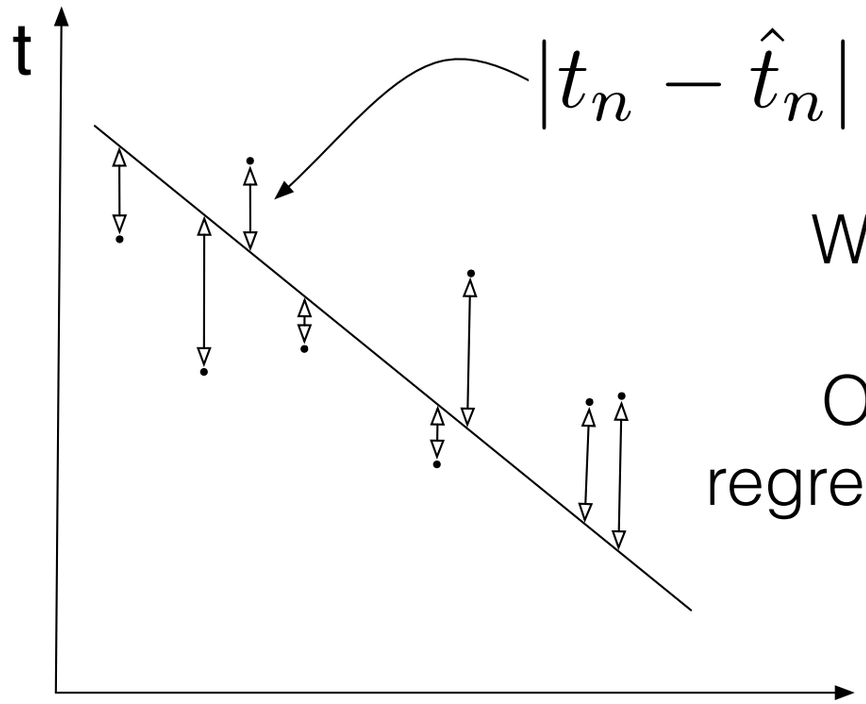
A natural choice is the ***squared loss*** function

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

Squared loss

Just square
the arrow length

a.k.a quadratic loss



$$|t_n - \hat{t}_n| = |t_n - f(x_n; \mathbf{w})|$$

Why vertical distance?

Orthogonal/Euclidean regression if we assume error is in both x and t

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

And the total loss for all inputs/observations?

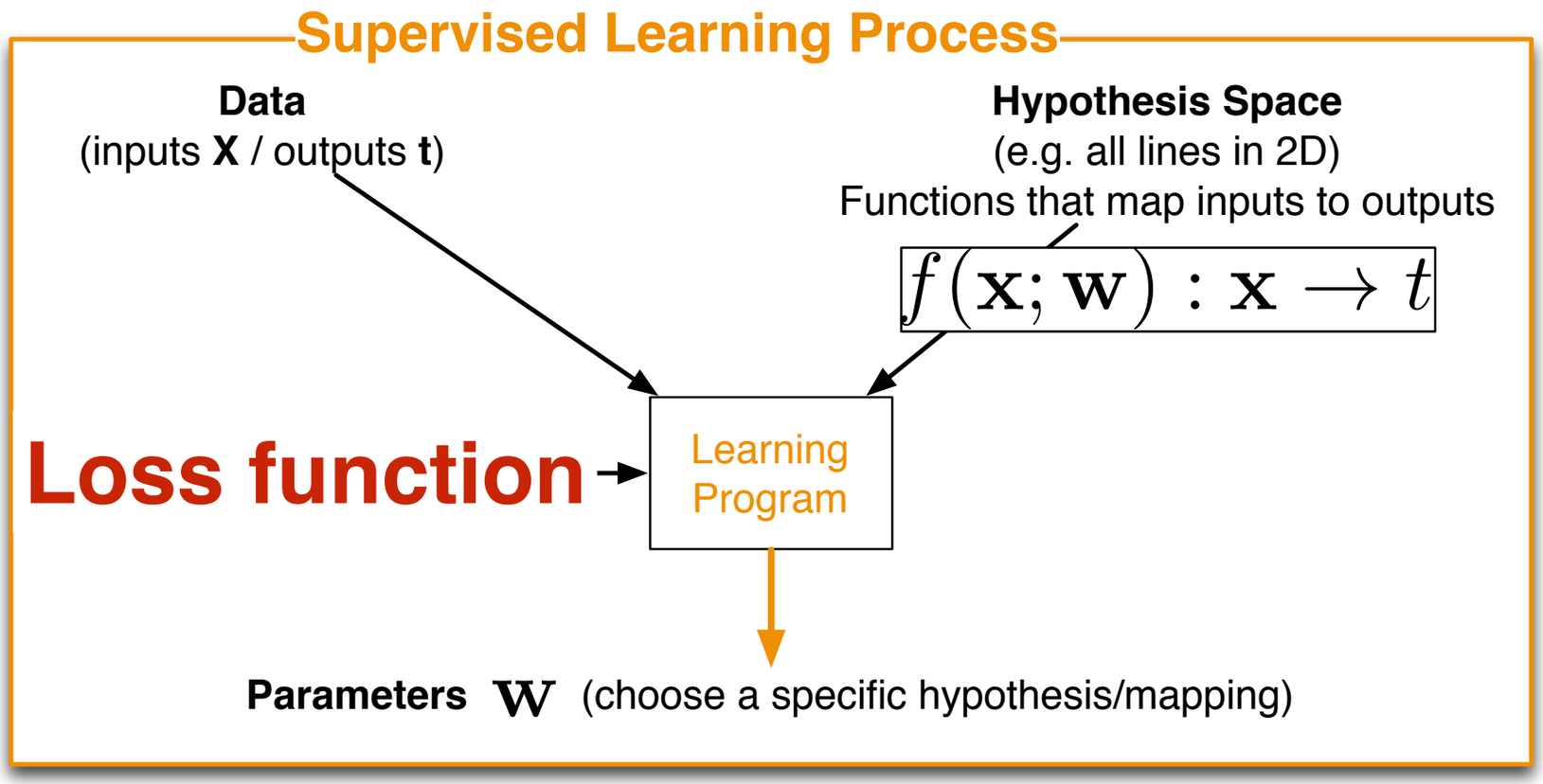
$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n$$

Any other Loss function you can think of?

$$\hat{\mathbf{t}} = \mathbf{X}\mathbf{w}$$

vector-matrix notation
see R&G p15-20!

Recap



Why do we use the squared error Loss?

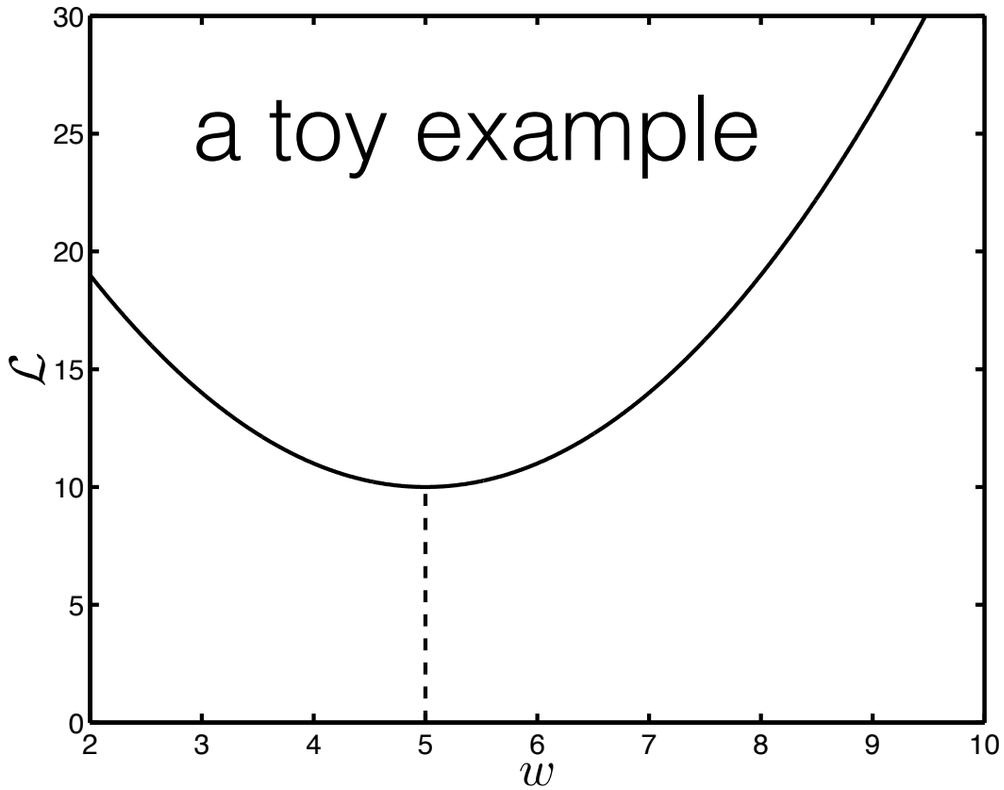
Because we can **analytically** derive (by simple math get a final expression by hand) the best* parameters/hypothesis.

Different lines (parameters) different Loss. Do you like loosing?

Final step: Minimise the Loss

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

Find the parameters that minimise the Loss



What do we want? from words to math..

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

Optimisation problem

Some necessary math

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

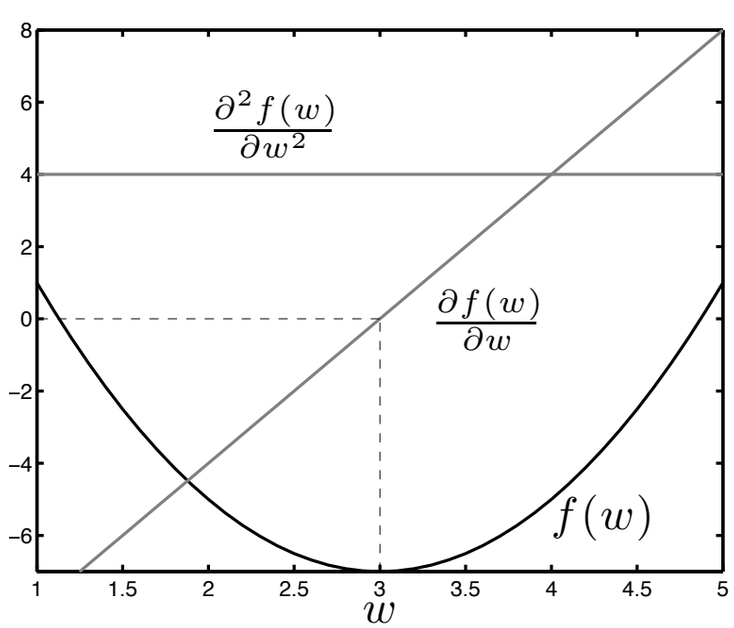
1. Replace with the expression for the specific Loss
2. Equate first derivative of Loss to zero to get **w** (find potential minima)
3. Examine second derivative of Loss at **w** to determine if unique minima

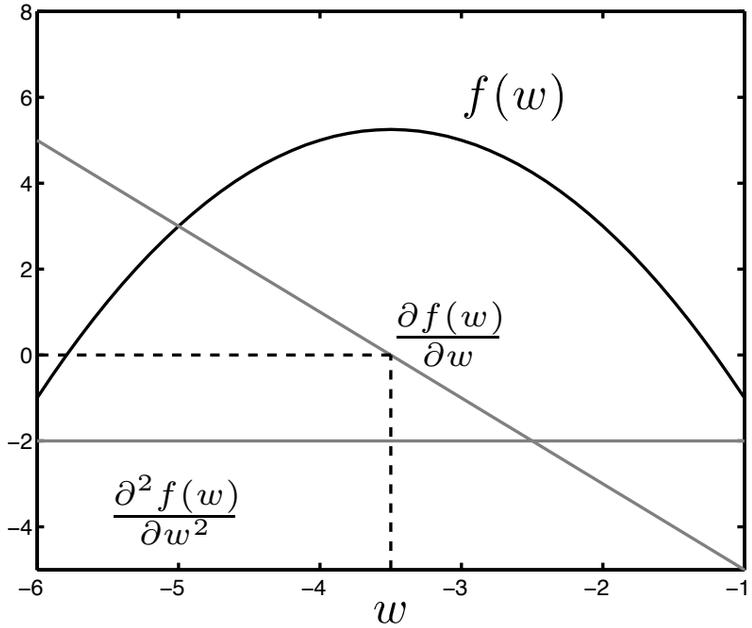
Find OLS parameters by $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} \rightarrow 0$

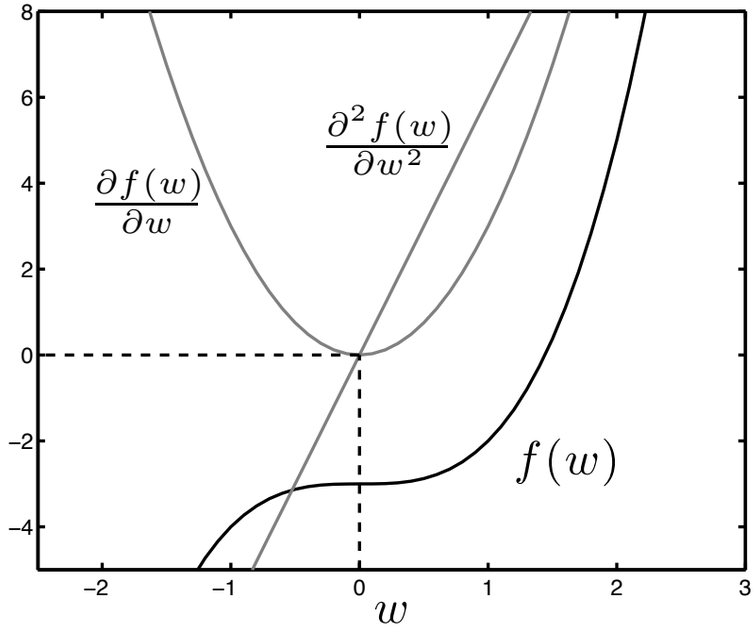
$$\text{if} \quad \frac{\partial^2 \mathcal{L}}{\partial^2 \mathbf{w}} > 0 \quad \text{we are at a minima}$$

The squared error Loss is **convex and smooth** so single global minima

Minima of a function







Derive OLS solutions (board/lab)

$$\widehat{w_0} = \bar{t} - w_1 \overline{x} = \frac{1}{N} \sum_{n=1}^{N} t_n - w_1 \sum_{n=1}^{N} x_n$$

Final form is:

$$\widehat{w_1} = \frac{\overline{xt} - \overline{x}\,\overline{t}}{\overline{x^2} - (\overline{x})^2}$$

Thats it!
Learning from data

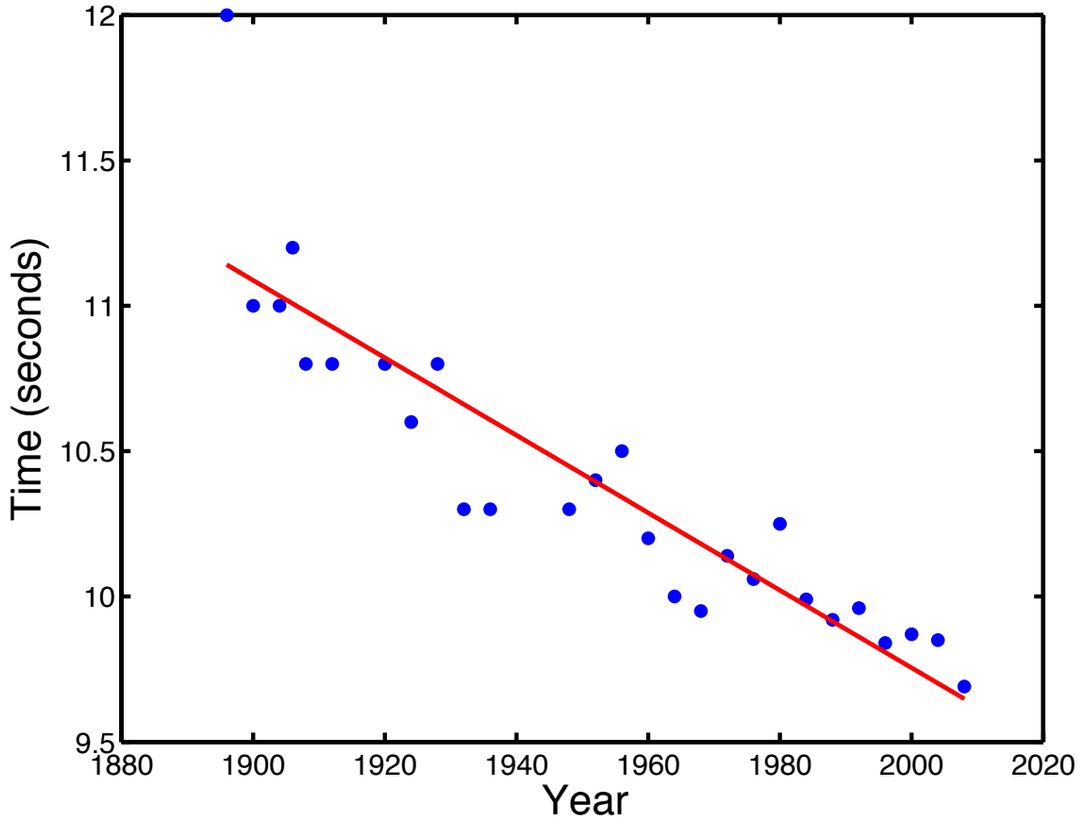
Department of COMPUTER SCIENCE

Breath!

You have just looked inside the box of your first ML model!

Making Predictions

The Ordinary Least Squares (OLS) fit to our training data

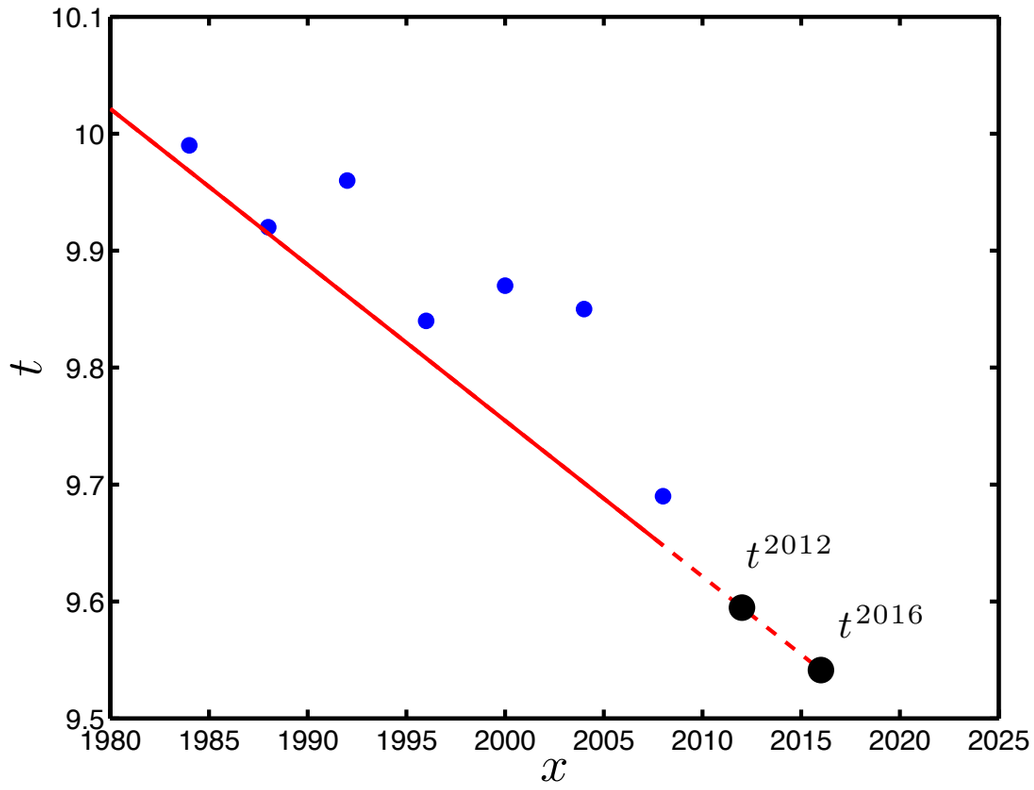


$$\hat{t} = f(x; \mathbf{w}) = w_0 + w_1 x$$

but we now have learned values for w so write:

$$\hat{t} = f(x; \mathbf{w}) = \widehat{w_0} + \widehat{w_1} x$$

What would our model predict for future?

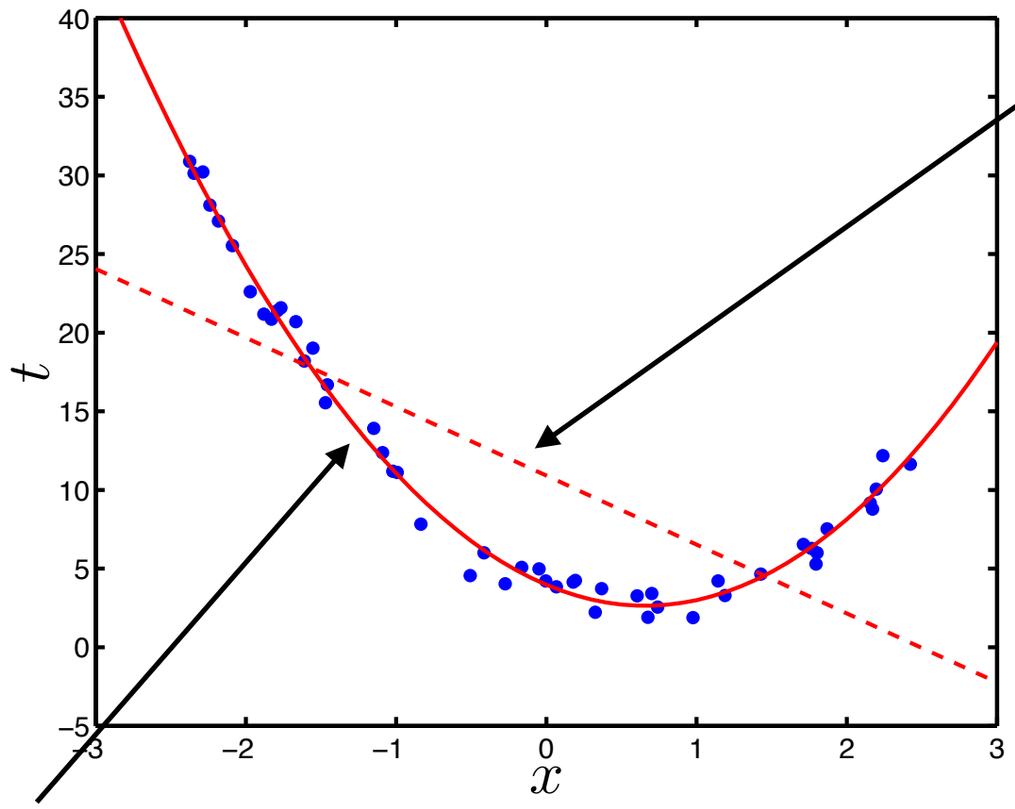


our prediction is $t^* = \widehat{w_0} + \widehat{w_1} x^*$

Can we create non-linear responses from a linear model?

linear regression (OLS): **linear in w**, linear in data x

$$f(x; \mathbf{w}) = w_0 + w_1 x$$



Pretty restrictive hypothesis space?

Demo

linear regression (OLS): **linear in w**, non-linear in data x

$$f(x; \mathbf{w}) = w_0 + w_1 x + w_2 x^2 = w_0 + w_1 \mathrm{x}_1 + w_2 \mathrm{x}_2$$

Still a linear structure (hyper-plane) but in a higher dimensional space

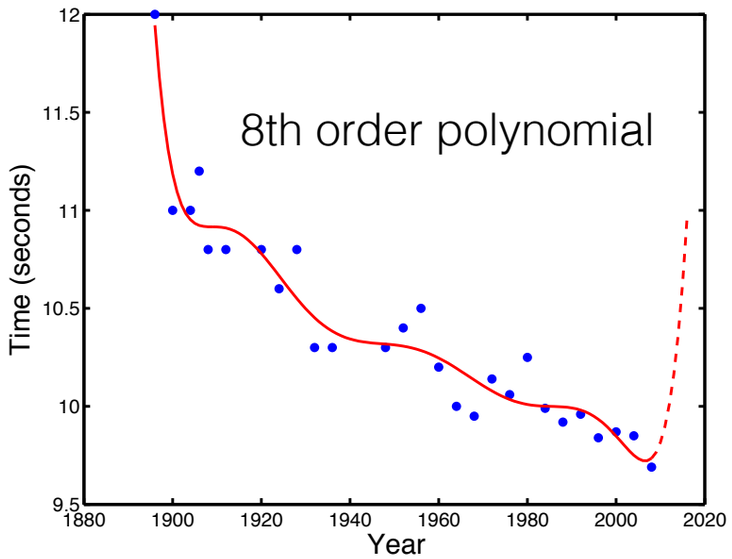
Feature engineering! A black art

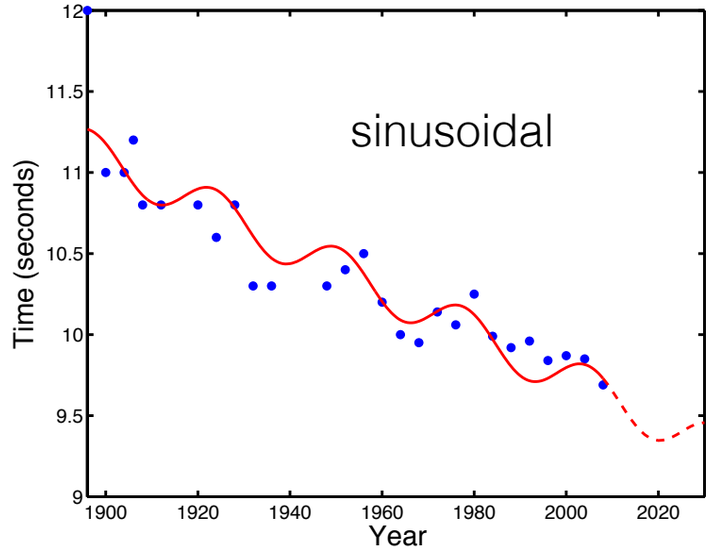
we map input space into new space called feature space

$$\mathbf{X} \rightarrow \Phi(\mathbf{X}) \quad \text{where} \quad \mathbf{X} \in \Re^{N \times D} \quad \Phi(\mathbf{X}) \in \Re^{N \times D*} \quad D* \gg D$$

From simple polynomial expansions to other functions

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





Whole fields of AI that deal with this problem for specific data types:
Computer Vision: Images, Video
NLP: Documents, Text
Acoustics: Sound

Matrix Derivation of OLS solution (see Ch1. R&G book p15)

We have input data **X** with N observations (rows) and D attributes (columns), and also have N outputs/targets **t**, one for every input.
e.g. If N=3 and D=2 our data would look like:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} \quad \text{and} \quad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

We will fit a linear model: line or (hyper-)plane in higher dimensions:

$$f(\mathbf{x}_n; \mathbf{w}) = w_0 + w_1 x_{n1} + \cdots + w_d x_{nd} + \cdots + w_D x_{nD}$$

for the example above this becomes:

$$f(\mathbf{x}_n; \mathbf{w}) = w_0 + w_1 x_{n1} + w_2 x_{n2}$$

And we decided to use the squared error or quadratic loss:

$$\mathcal{L}_n(t_n, \hat{t}_n) = (t_n - f(\mathbf{x}_n; \mathbf{w}))^2$$

Matrix Derivation of OLS solution (see Ch1. R&G book p15)

we can transform our **X** to include a column of 1s in the left and hence re-write our function in a compact vector-matrix format

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \end{bmatrix} \quad \text{and} \quad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

we could use another symbol instead of X to denote this but we are "overloading" notation and still call it **X**.

$$f(\mathbf{x}_n; \mathbf{w}) = w_0 + w_1 x_{n1} + w_2 x_{n2} = \mathbf{x}_n \mathbf{w}$$

where $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ So for all n our linear model is: $\mathbf{X}\mathbf{w}$

Matrix Derivation of OLS solution (see Ch1. R&G book p15)

So our squared error Loss function that we want to minimise can be written as:

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

We want to differentiate the loss wrt to the parameters and set it to zero:

expanding the brackets to re-write the loss

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}\left(\frac{1}{N}(\mathbf{t}^{\mathrm{T}}\mathbf{t} - \mathbf{t}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w})\right) = 0$$

noting that second and third term inside parenthesis are equal
and following rules of differentiation (see p21, table 1.4):

$$-2\mathbf{X}^{\mathrm{T}}\mathbf{t} + 2\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} = 0$$

we get the final solution: $\mathbf{w} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$