

Machine Learning

CS342

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

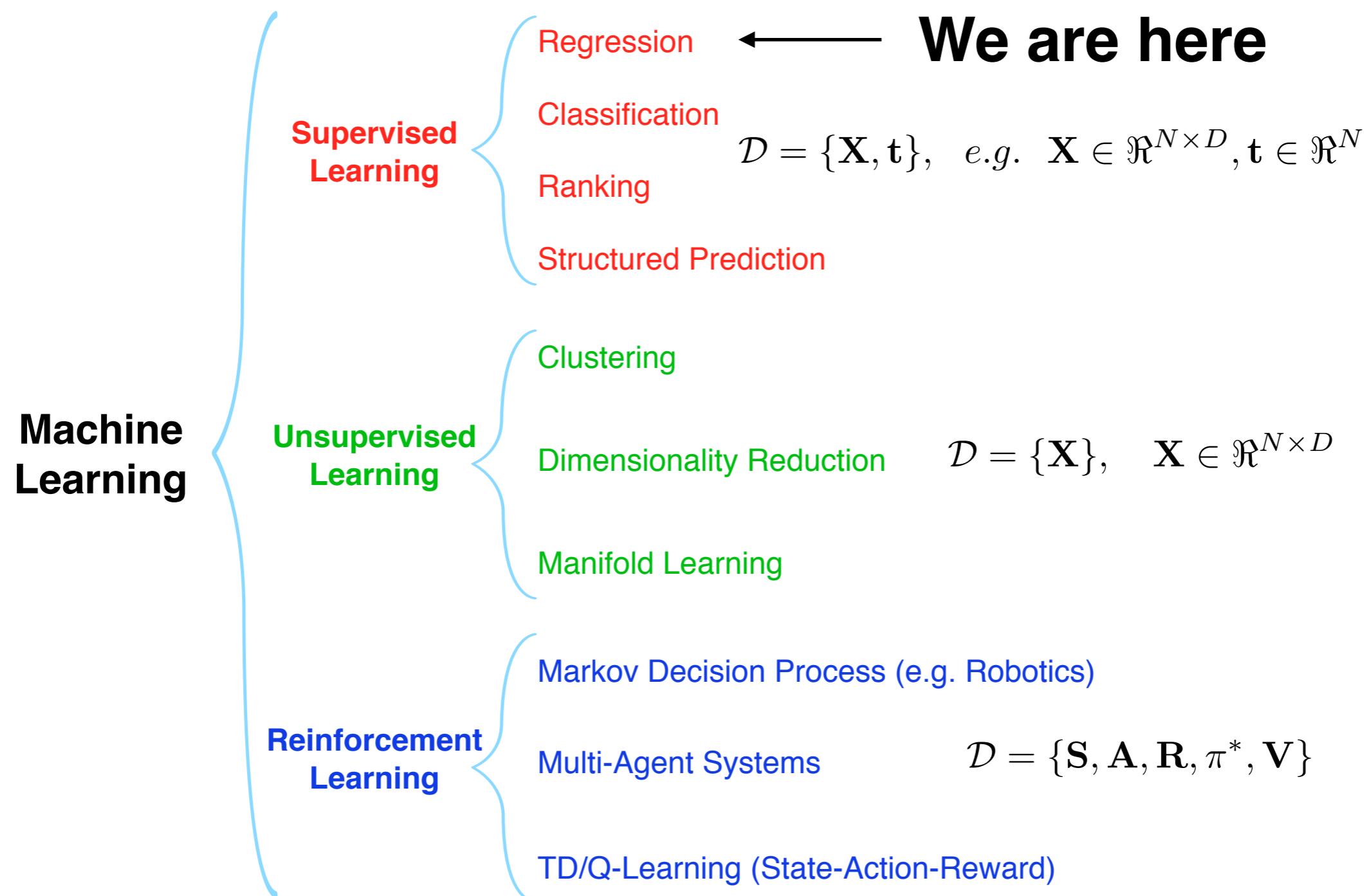
Dr. Theo Damoulas
T.Damoulas@warwick.ac.uk

Office hours (CS 307)

Mon 16:00-17:00

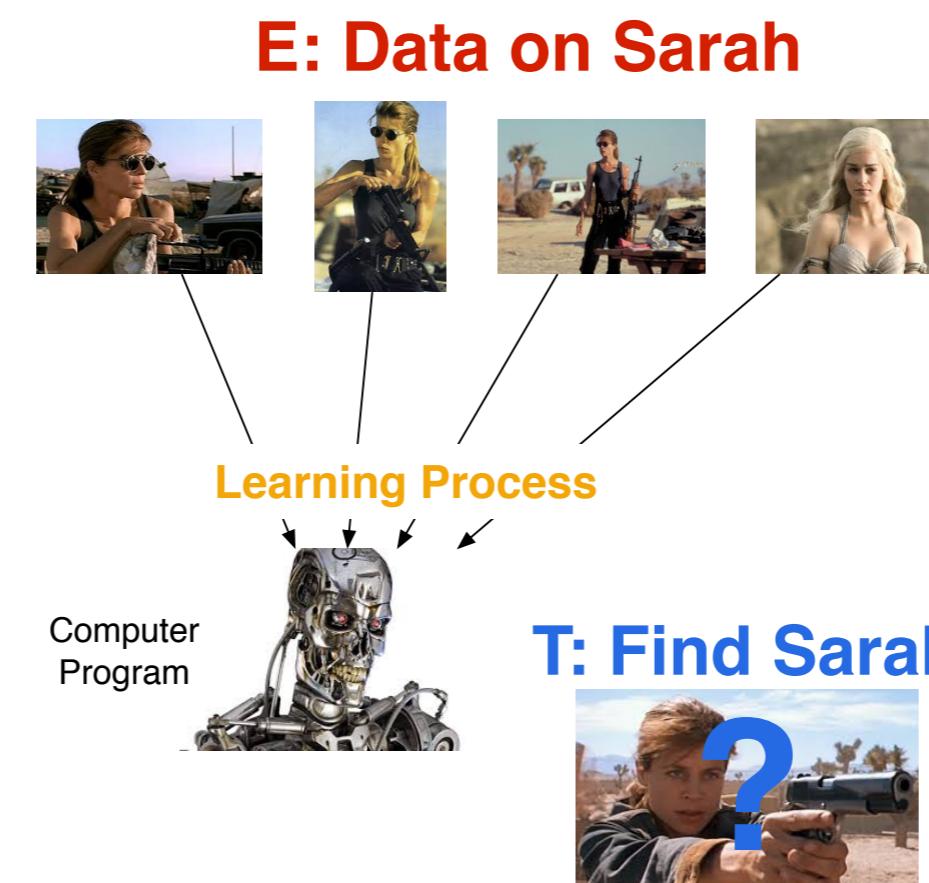
Fri 16:00-17:00

Recap: ML subfields



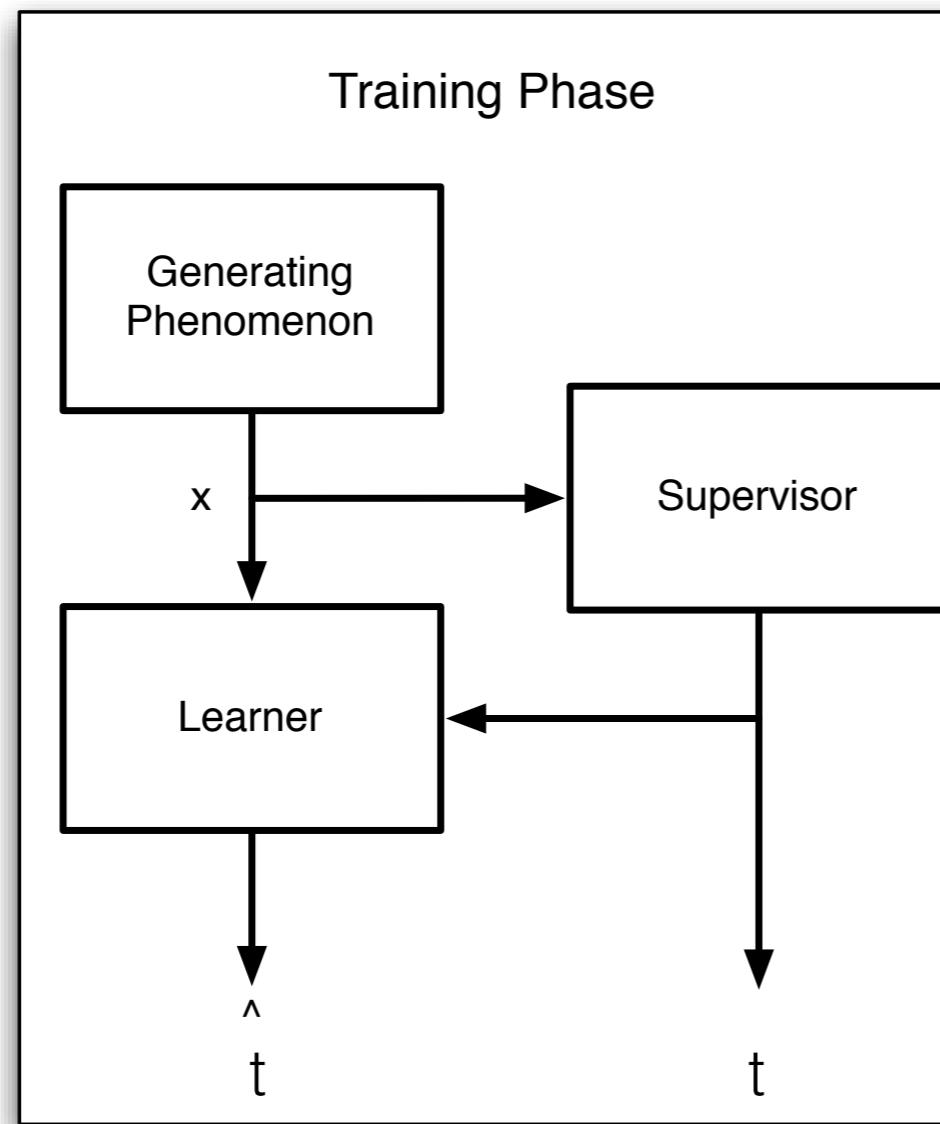
Recap: Definition of “learning from data”

- 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]

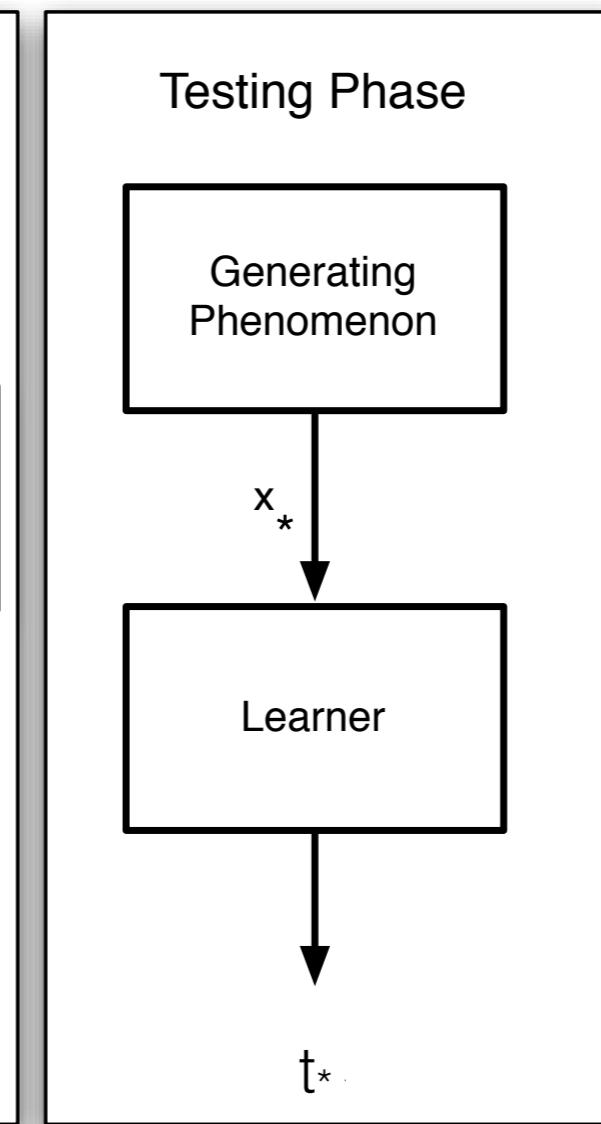


Recap: Schematic of Learning (Training) and Prediction phase

Learning



Prediction

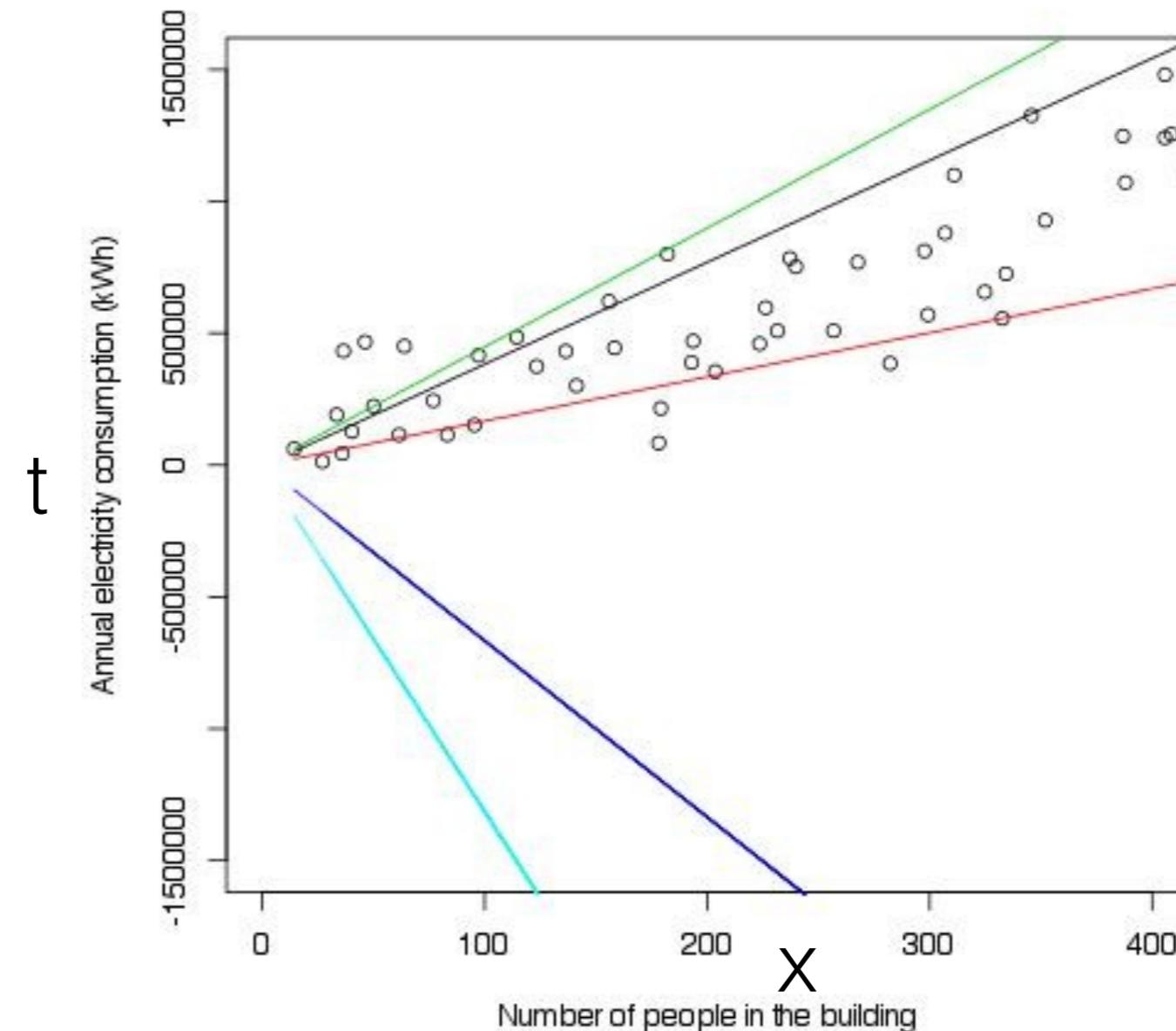


$$D_{\text{training}} = \{\mathbf{X}, \mathbf{t}\} = \{\mathbf{x}_n, t_n\}_{n=1}^N$$

$$D_{\text{testing}} = \{\mathbf{X}_*\}$$

Recap: Linear Regression

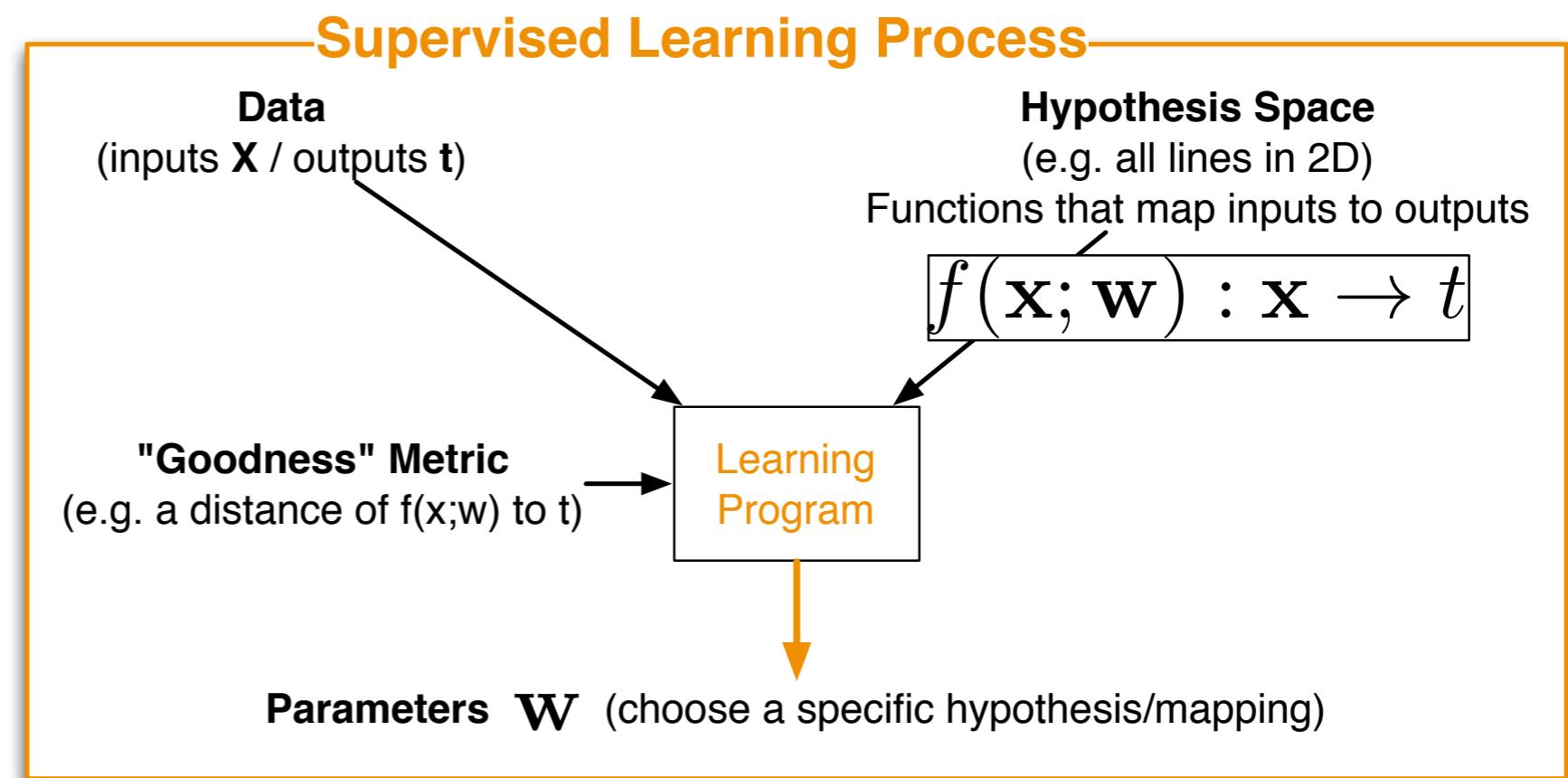
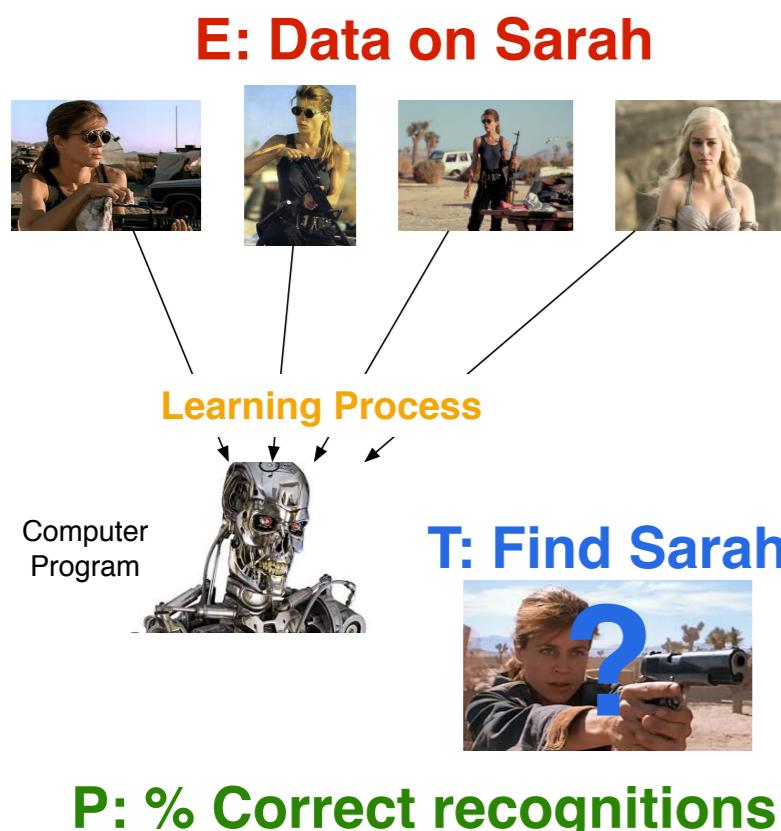
Regression:
t is continuous



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

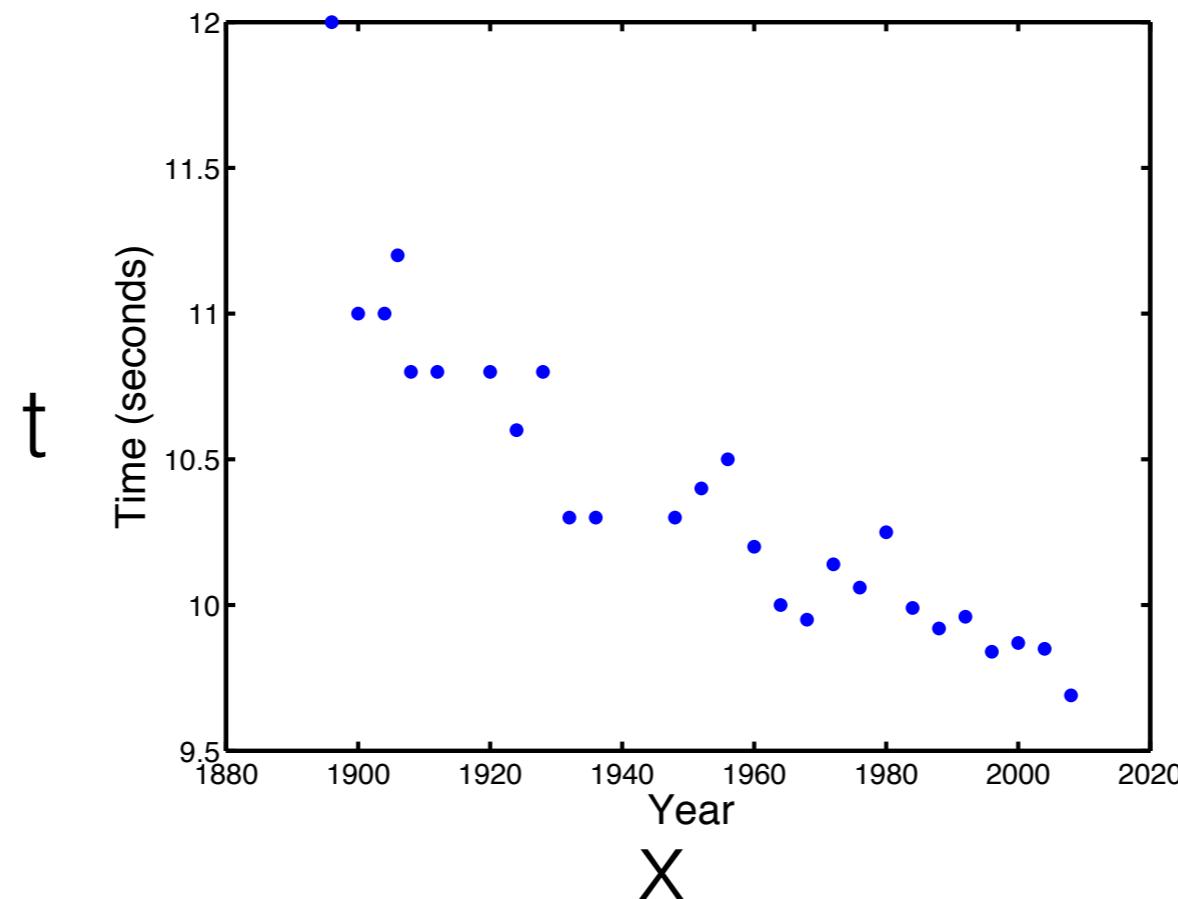
$$\hat{t} = f(x; w_0, w_1) = w_0 + w_1 x$$

Recap: Supervised Learning components



Learning as parameter (hypothesis) inference

Revisit Olympics example: Univariate regression



Rogers & Girolami Ch. 1

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

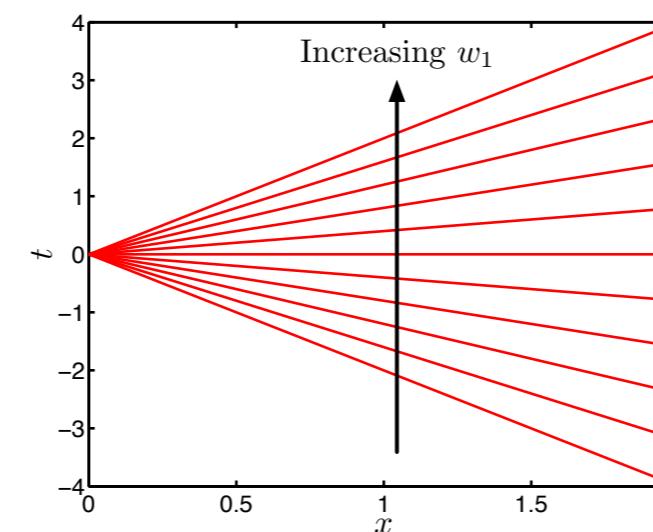
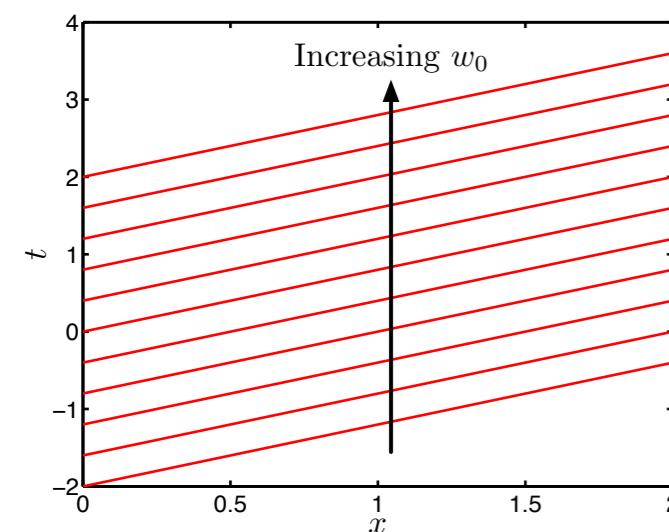
$$f(x; \mathbf{w}) : x \rightarrow t$$

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

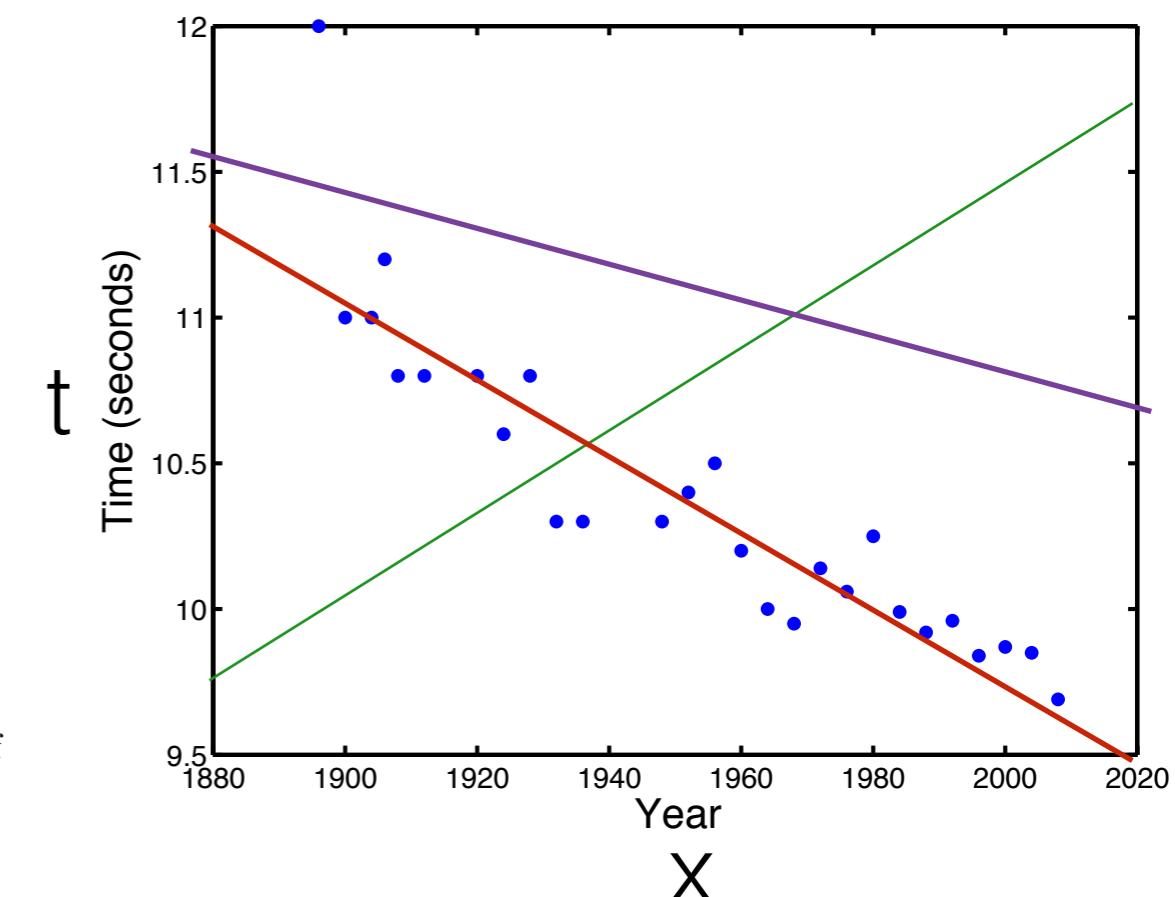
$$\hat{t} = f(x; \mathbf{w}) = w_0 + w_1 x$$

What is a good model?

Different parameters
give us different lines/hypothesis



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



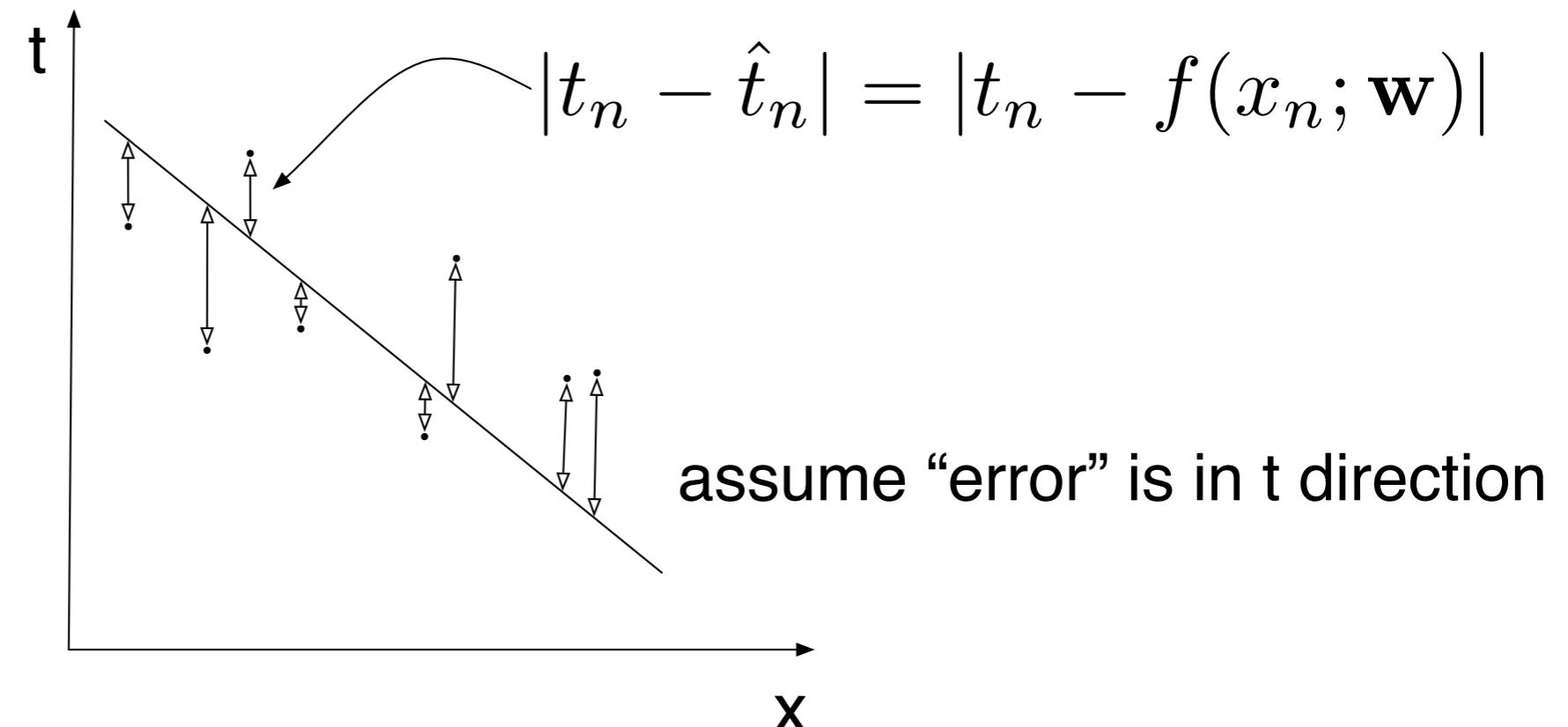
Ok so how do we find the right parameters?
What is one possible measure of “goodness”?

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 $\{x_n, t_n\}$



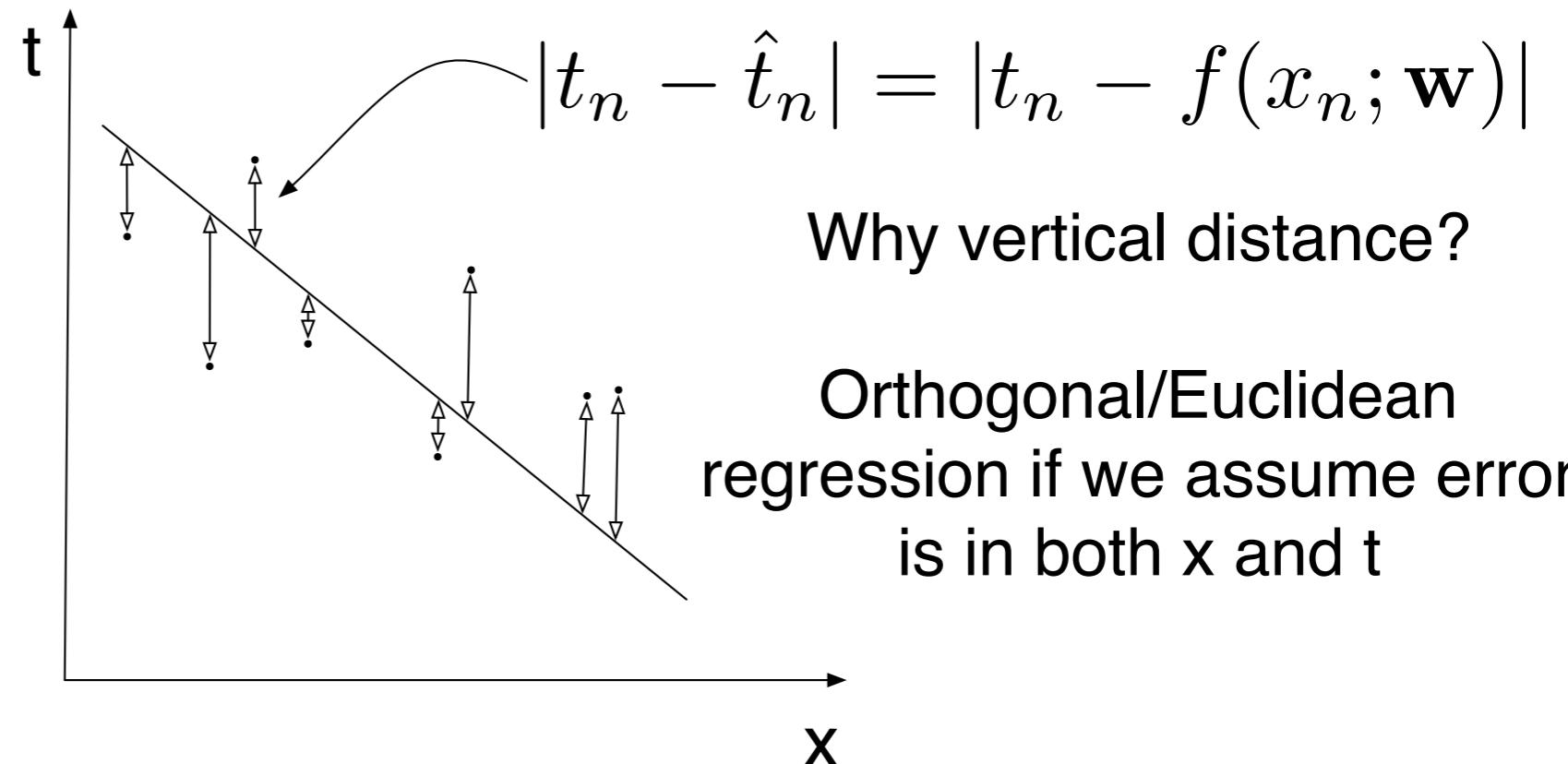
A natural choice is the **squared loss** function

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

Squared loss

Just square
the arrow length

a.k.a quadratic loss



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?

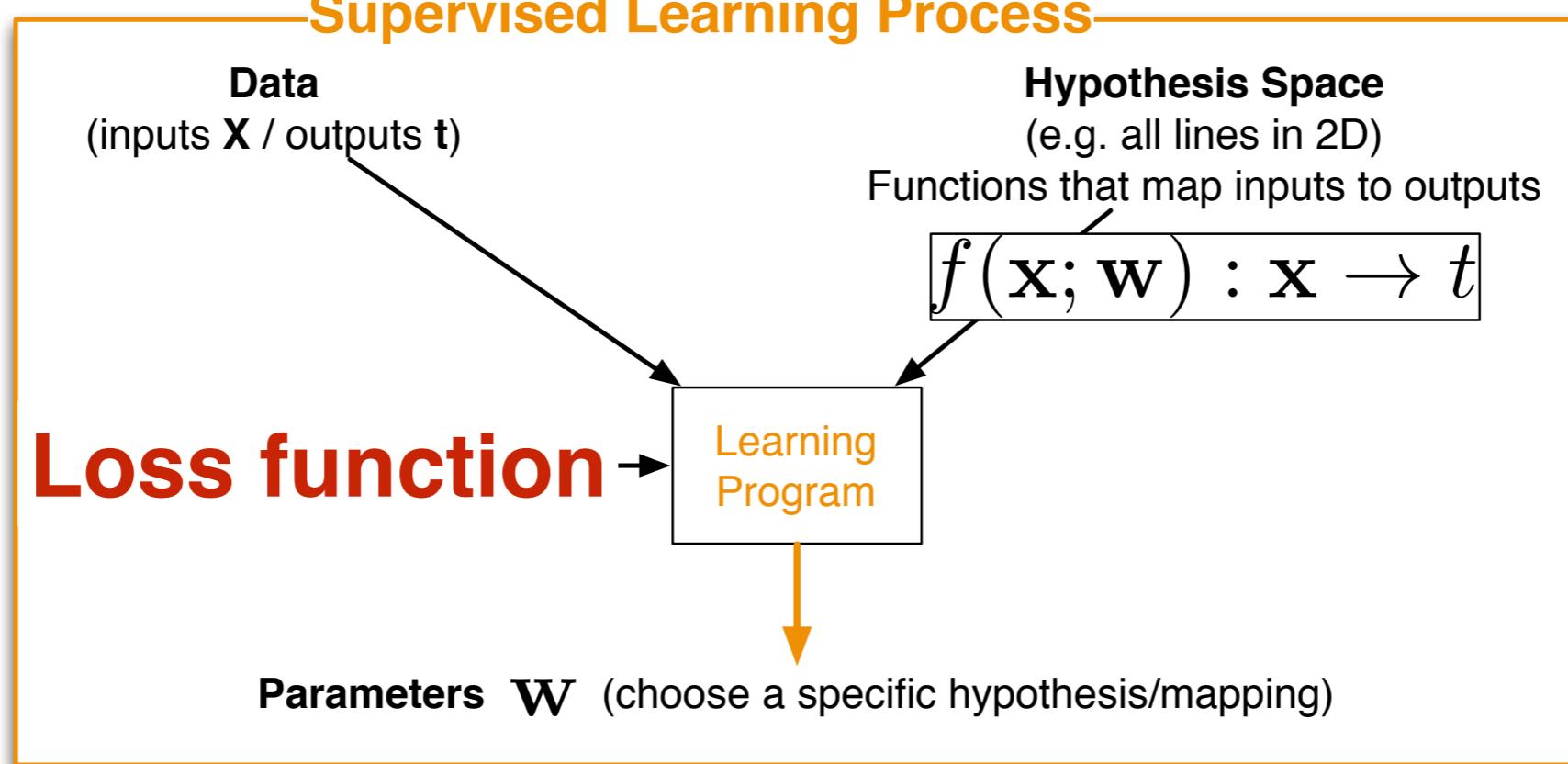
$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}_n$$

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?

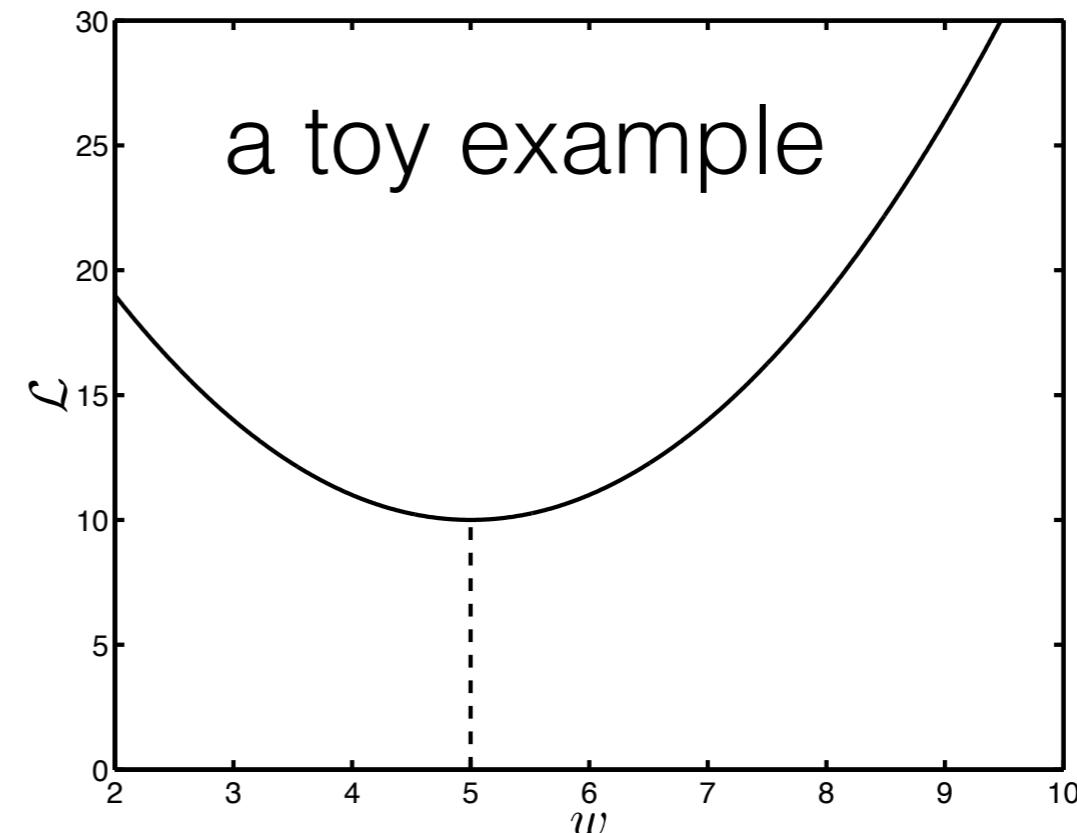
Because we can **analytically** derive (by simple math get a final expression by hand) the best* parameters/hypothesis.

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..

$$\widehat{w}_0, \widehat{w}_1 \leftarrow \operatorname{argmin}_{w_0, w_1} \frac{1}{N} \sum_{n=1}^N \mathcal{L}_n(t_n, f(x_n; w_0, w_1))$$

Optimisation problem

Some necessary math

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

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

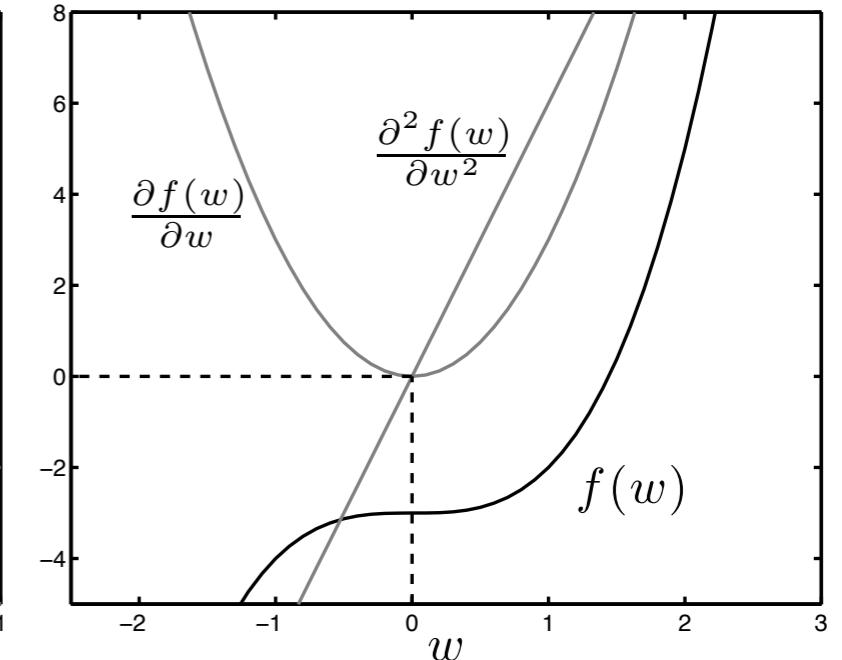
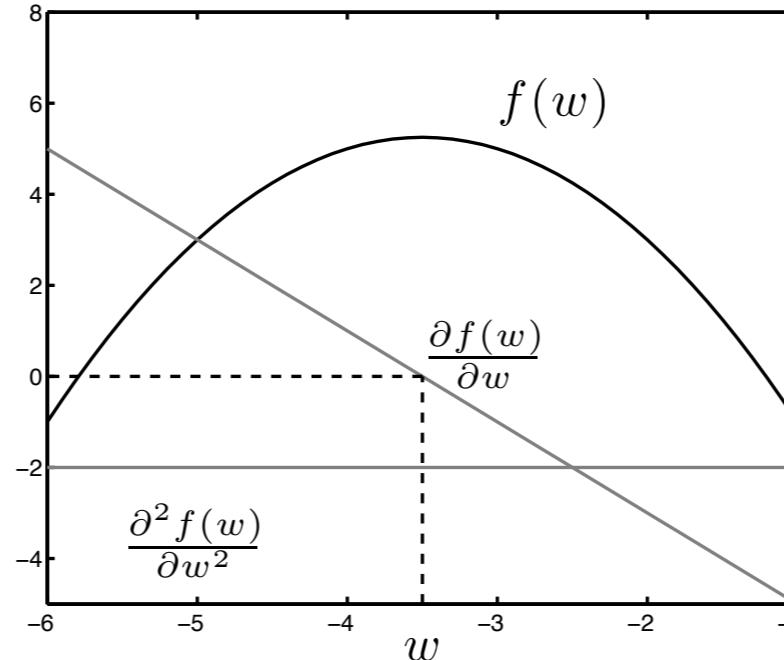
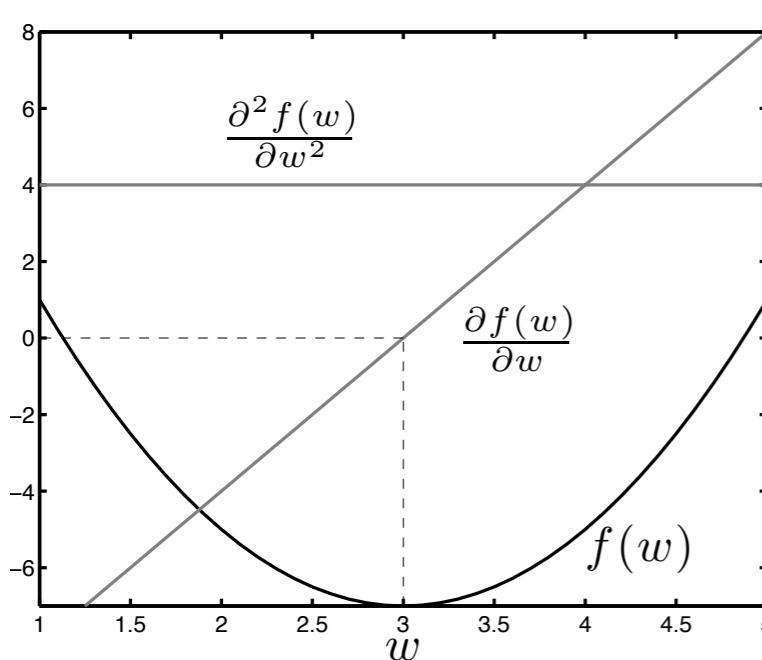
Find OLS parameters by

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} \rightarrow 0$$

if $\frac{\partial^2 \mathcal{L}}{\partial^2 \mathbf{w}} > 0$ 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)

$$\widehat{w}_0 = \bar{t} - w_1 \bar{x} = \frac{1}{N} \sum_{n=1}^N t_n - w_1 \sum_{n=1}^N x_n$$

Final form is:

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

Thats it!
Learning from data

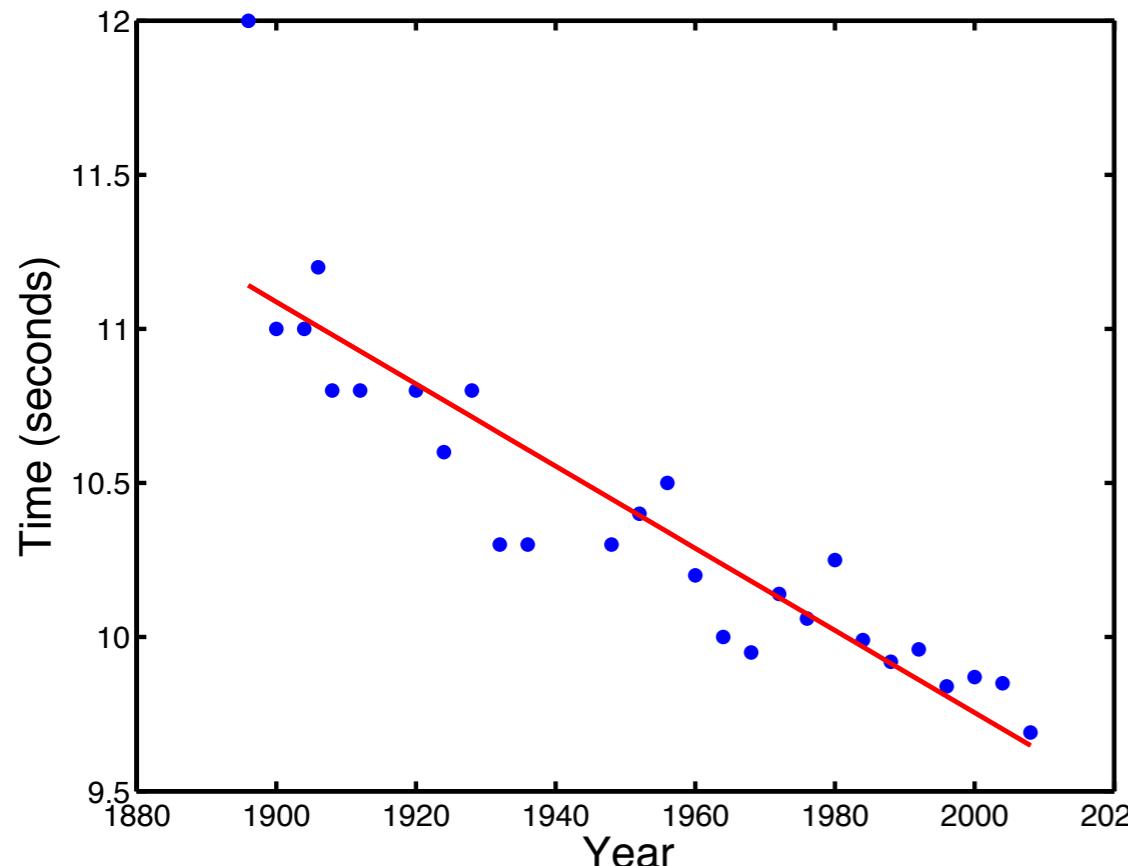
Breath!

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



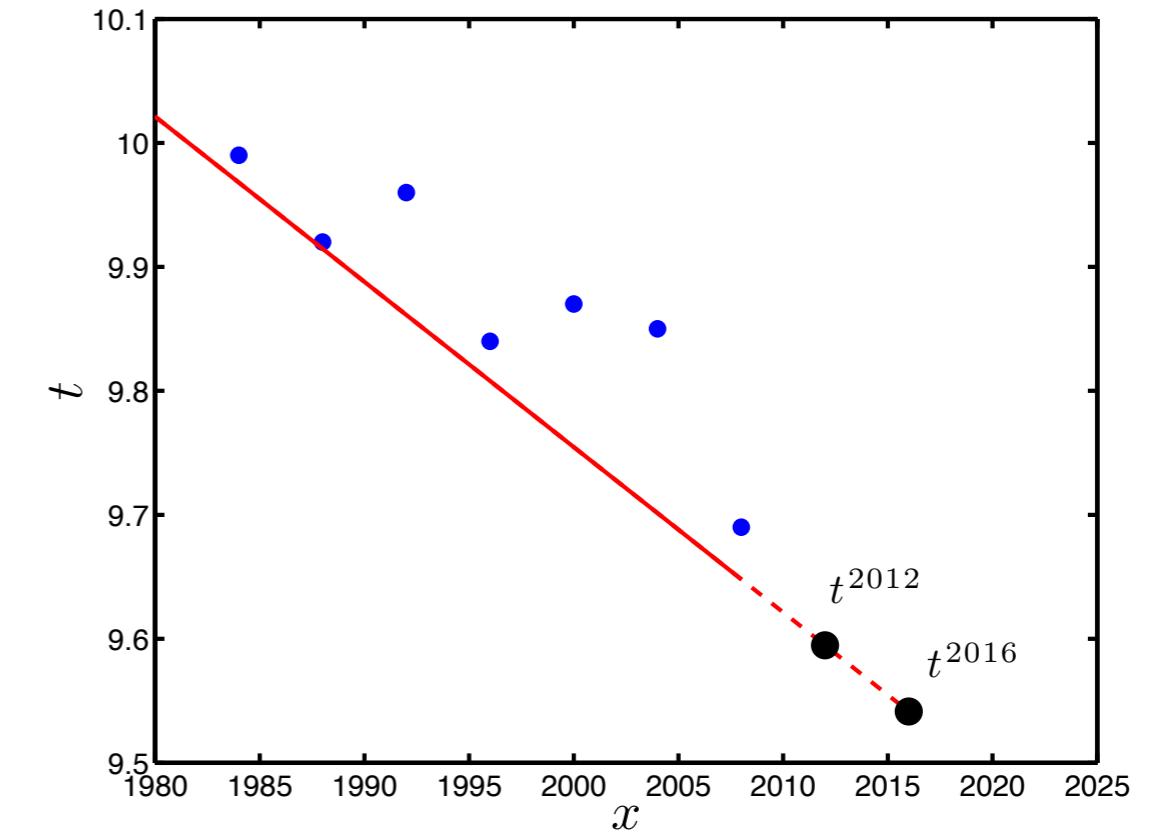
Making Predictions

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



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

What would our model predict for future?



but we now have learned values for w so write:

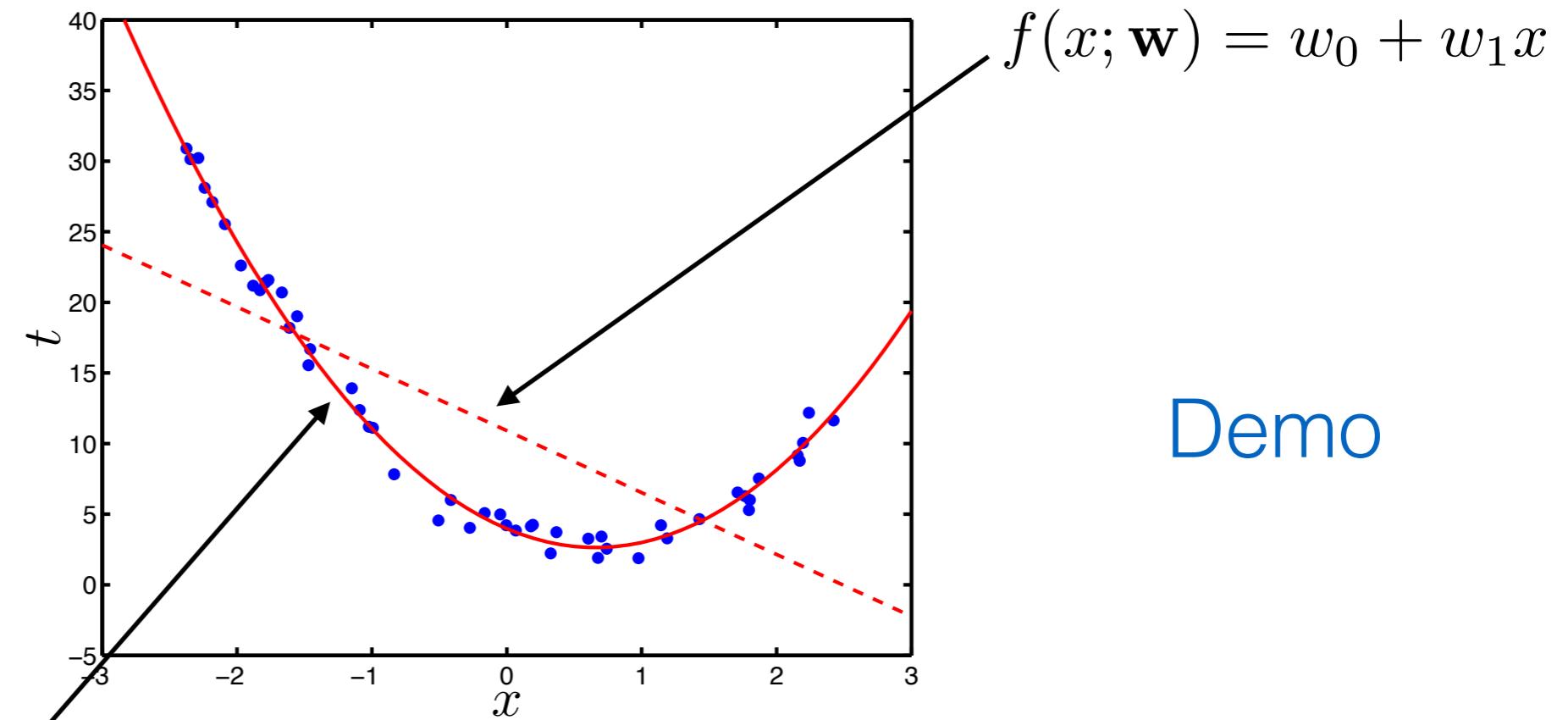
$$\hat{t} = f(x; \mathbf{w}) = \hat{w}_0 + \hat{w}_1 x$$

our prediction is $t^* = \hat{w}_0 + \hat{w}_1 x^*$

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

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

Pretty restrictive
hypothesis space?



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

$$f(x; \mathbf{w}) = w_0 + w_1 x + w_2 x^2 = w_0 + w_1 x_1 + w_2 x_2$$

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

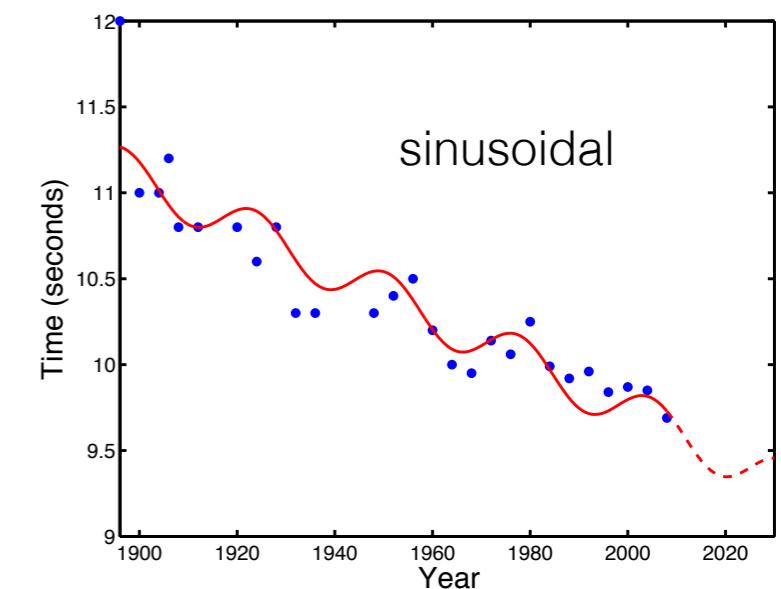
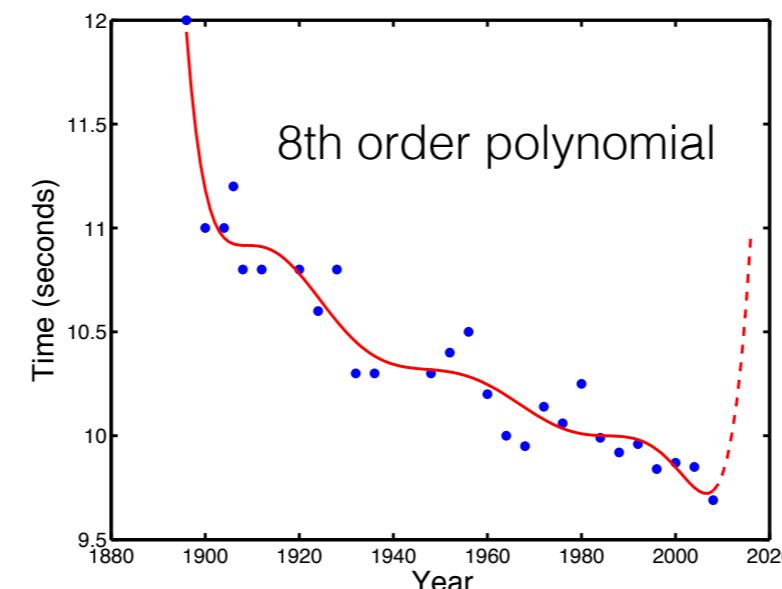
Feature engineering! A black art

we map input space into new space called feature space

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

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 \mathbf{X} with N observations (rows) and D attributes (columns), and also have N outputs/targets \mathbf{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 \mathbf{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 \mathbf{X} .

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

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

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:

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

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

expanding the brackets to re-write the loss

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

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

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

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