University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 3 - Linear Regression - Basis functions - Maximum Likelihood - Stochastic Gradient Descent - Underfitting and Overfitting - Regularized Least Squares

Erik Bekkers

(Bishop 3.1)

Image credit: Kirillm | Getty Images

University of Amsterdam
Informatics Institute

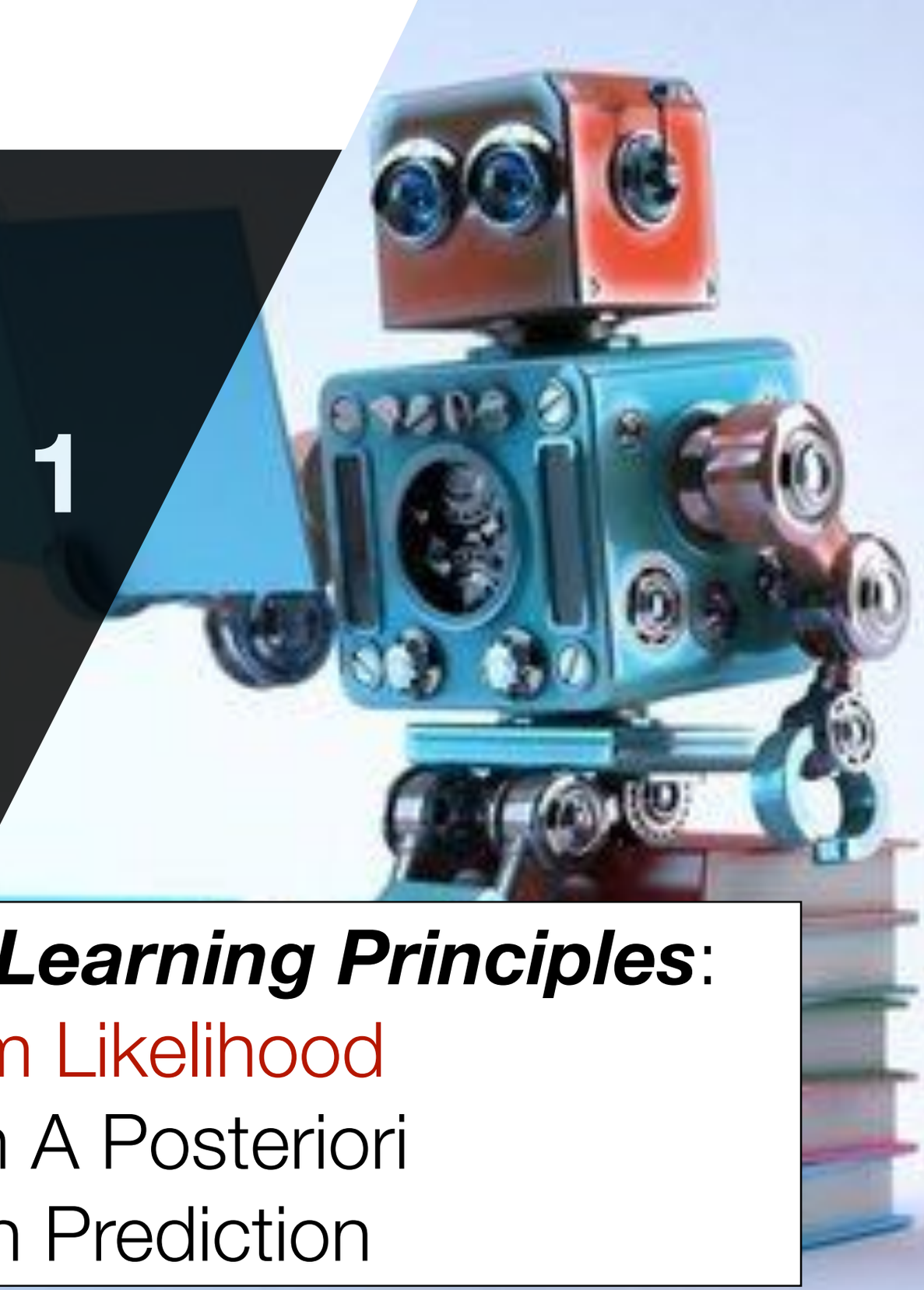
AMLAB Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 2.3 - Maximum Likelihood

Erik Bekkers

(Bishop 1.2.3 - 1.2.5)



Three Statistical Learning Principles:

Maximum Likelihood

Maximum A Posteriori

Bayesian Prediction

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Maximum Likelihood Principle

- **Maximum Likelihood Optimisation**: Consider $D = \{x_1, x_2, \ldots, x_N\} \sim p(D \mid \mathbf{w})$, assuming it came from some parametric distribution $p(D \mid \mathbf{w})$. Then solve:

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(D \mid \mathbf{w}) \qquad \text{(find parameters } \mathbf{w} \text{ that maximise the likelihood } p(D \mid \mathbf{w}) \text{)}$$

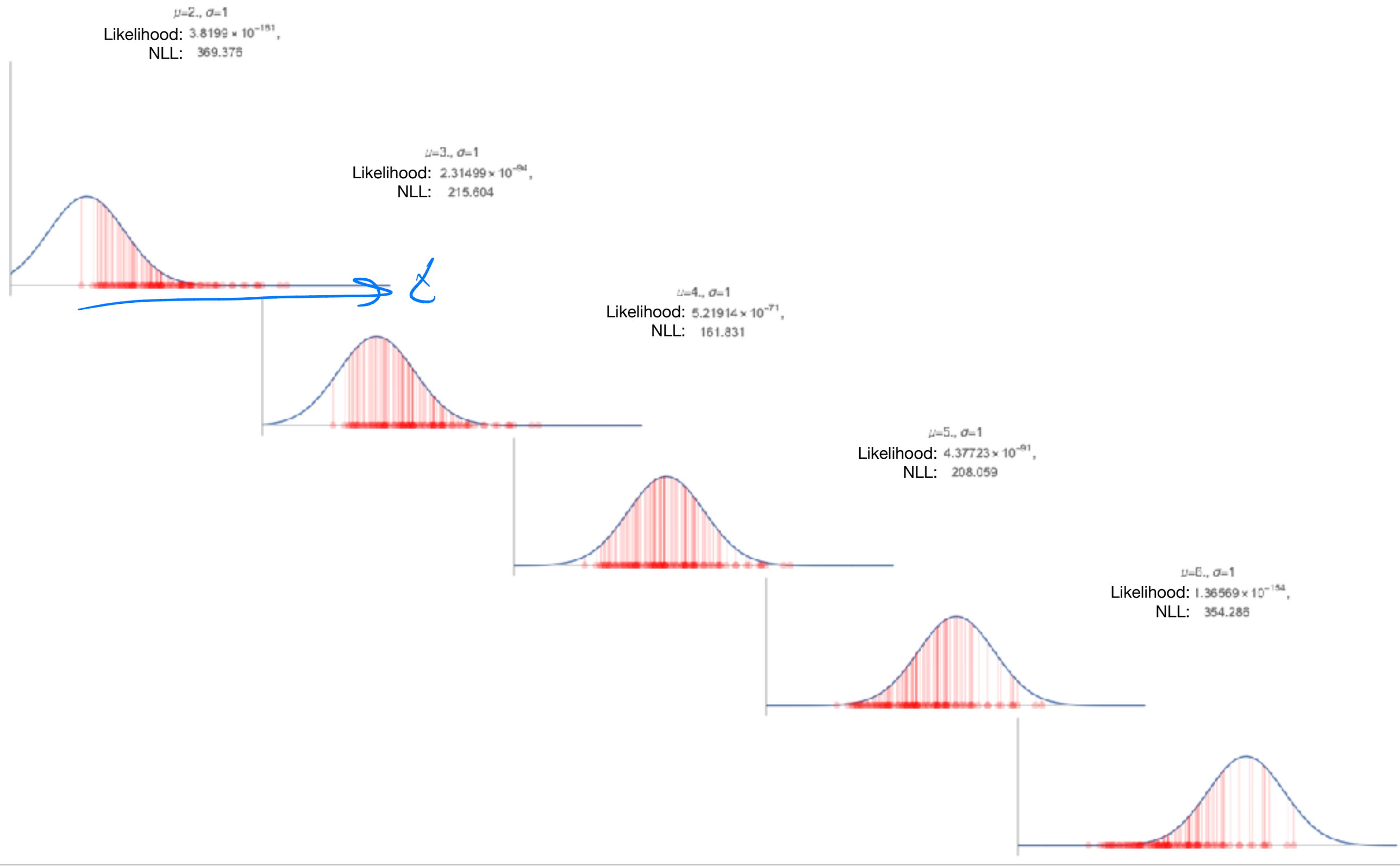
- Assume samples in D to be **independent, identically distributed** (i.i.d), meaning each $x_i \sim p(x_i \mid \mathbf{w})$:

$$\begin{aligned} p(D \mid \mathbf{w}) &= p(x_1, x_2, \ldots, x_N \mid \mathbf{w}) \\ &= p(x_1 \mid \mathbf{w}) p(x_2 \mid \mathbf{w}) \ldots p(x_3 \mid \mathbf{w}) \\ &= \prod_{i=1}^{N} p(x_i \mid \mathbf{w}) \end{aligned}$$

- Define equivalent problem of minimising **negative log likelihood (NLL)**:

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmin}} -\log p(D \mid \mathbf{w}) \quad \overset{i.i.d.}{\Longrightarrow} \quad \mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmin}} -\sum_{i=1}^{N} \log p(x_i \mid \mathbf{w})$$

Maximum Likelihood Principle



Maximum Likelihood Principle

Analytic solutions for Gaussians, i.e., the following can be solved in closed form

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmin}} - \sum_{i=1}^{N} \log p(x_i \,|\, \mathbf{w})$$

The solution being:

μ_{ML} is Sample mean:

$$\mu_{ML} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

σ^2_{ML} is sample variance:

$$\sigma^2_{ML} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_{ML})^2$$

Is the estimator unbiased?

$$\mathbb{E}_{D \sim p(D|\mu,\sigma)}[\mu_{ML}] = \mu$$

$$\mathbb{E}_{D \sim p(D|\mu,\sigma)}[\sigma^2_{ML}] \neq \frac{N-1}{N}\sigma^2$$

biased!

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 2.4 - Maximum Likelihood: An Example

Erik Bekkers

(Bishop 1.2.3 - 1.2.5)

Slide credits: Patrick Forré and Rianne van den Berg

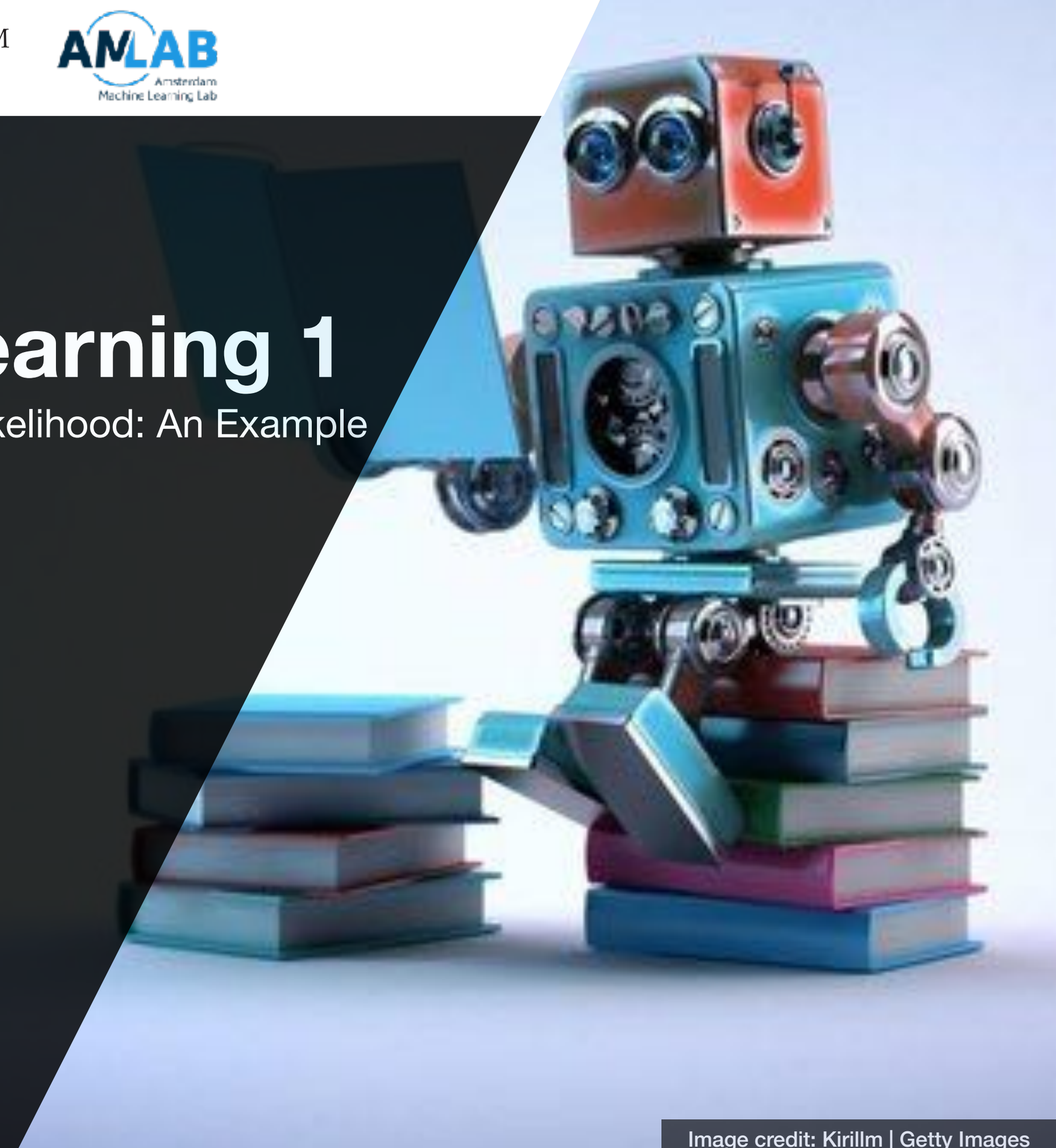


Image credit: Kirillm | Getty Images

Curve Fitting: Maximum Likelihood Estimates

- Data $D = \{(x_1, t_1), \ldots, (x_N, t_N)\} = \{\mathbf{x}, \mathbf{t}\}$

- Assume data is generated through underlying model

$$t = y(x, \mathbf{w}) + \sigma\,\epsilon$$

with Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, 1)$

- Define precision β such that $\beta^{-1} = \sigma^2$

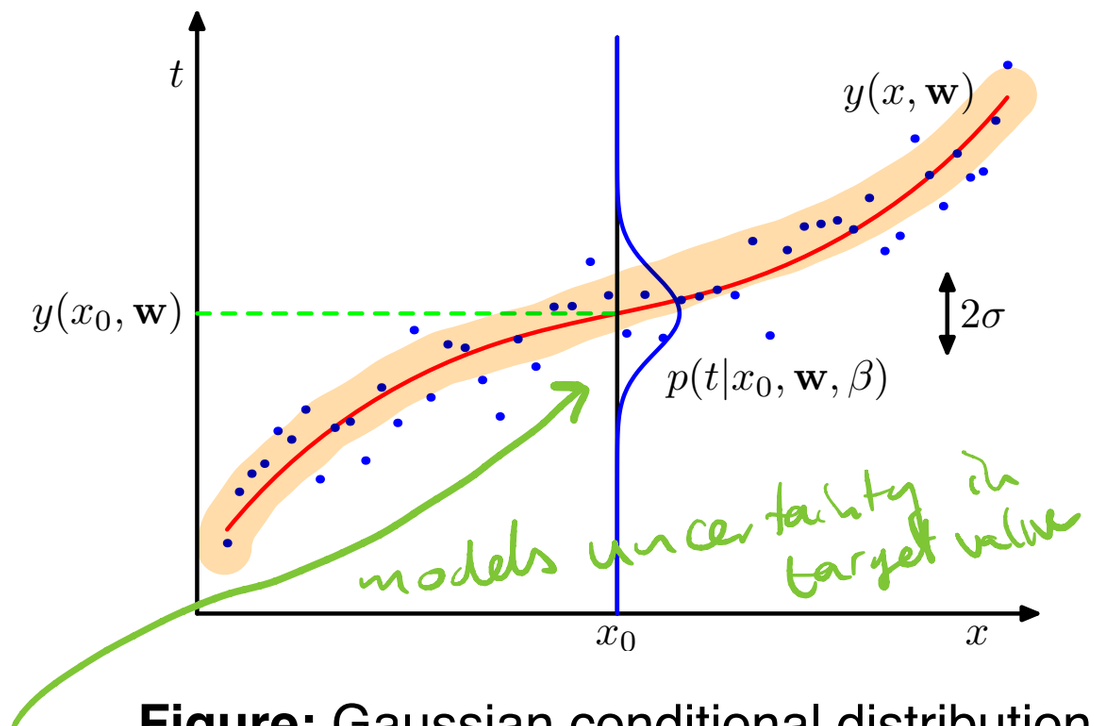


Figure: Gaussian conditional distribution (Bishop 1.16)

- Under these model assumptions we have target distribution:

$$p(t \mid x, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(x, \mathbf{w}), \beta^{-1})$$

$y(x, \mathbf{w})$: mean per position; β^{-1}: σ^2

- Log likelihood:

$$\log p(\mathbf{t} \mid \mathbf{x}, \mathbf{w}, \beta) = \log \prod_{i}^{N} \mathcal{N}(t_i \mid y(x_i, \mathbf{w}), \beta^{-1})$$

$$= \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi - \frac{\beta}{2} \sum_{i=1}^{N} (t_i - y(x_i, \mathbf{w}))^2$$

Curve Fitting: Maximum Likelihood Estimates

- ML: minimize $E(\mathbf{x}, \mathbf{t}, \mathbf{w}, \beta) = -\log p(\mathbf{t} \,|\, \mathbf{x}, \mathbf{w}, \beta)$ w.r.t. $\mathbf{w}$ and β

$$E(\mathbf{x}, \mathbf{t}, \mathbf{w}, \beta) = \frac{\beta}{2} \sum_{i=1}^{N} (y(x_i, \mathbf{w}) - t_i)^2 \; - \; \frac{N}{2} \log \beta \; + \; \frac{N}{2} \log 2\pi$$

does not depend on $\mathbf{w}$

- Maximum likelihood solution (least squares fit!):

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\beta}{2} \sum_{i=1}^{N} (y(x_i, \mathbf{w}) - t_i)^2$$

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{i=1}^{N} (y(x_i, \mathbf{w}_{ML}) - t_i)^2$$

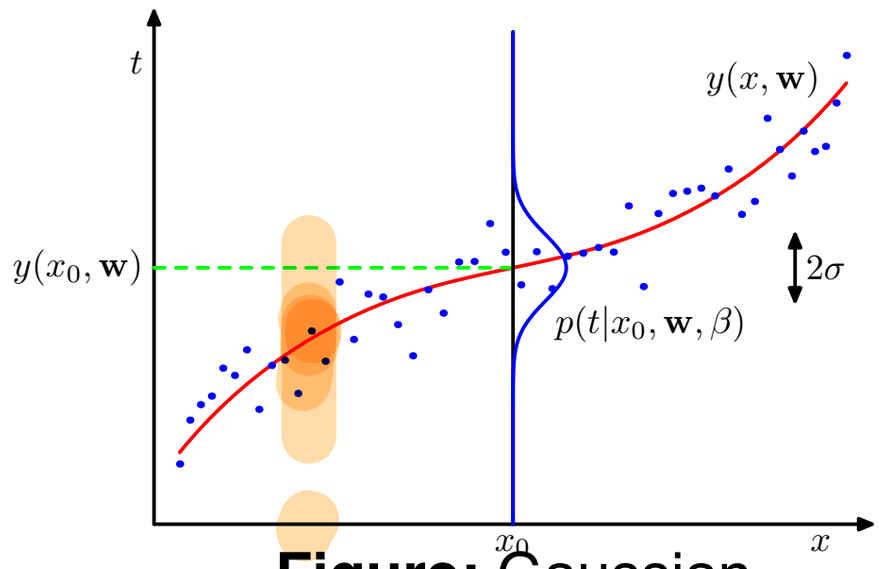


Figure: Gaussian conditional distribution (Bishop 1.16)

- **Predictive distribution**:

best point estimate

$$p(t' \,|\, x', \mathbf{w}_{ML}, \beta_{ML}) = \mathcal{N}(t' \,|\, y(x', \mathbf{w}_{ML}), \beta_{ML}^{-1})$$

$$\mathbb{E}[t' \,|\, x', \mathbf{w}_{ML}, \beta_{ML}] = y(x', \mathbf{w}_{ML})$$

University of Amsterdam
Informatics Institute

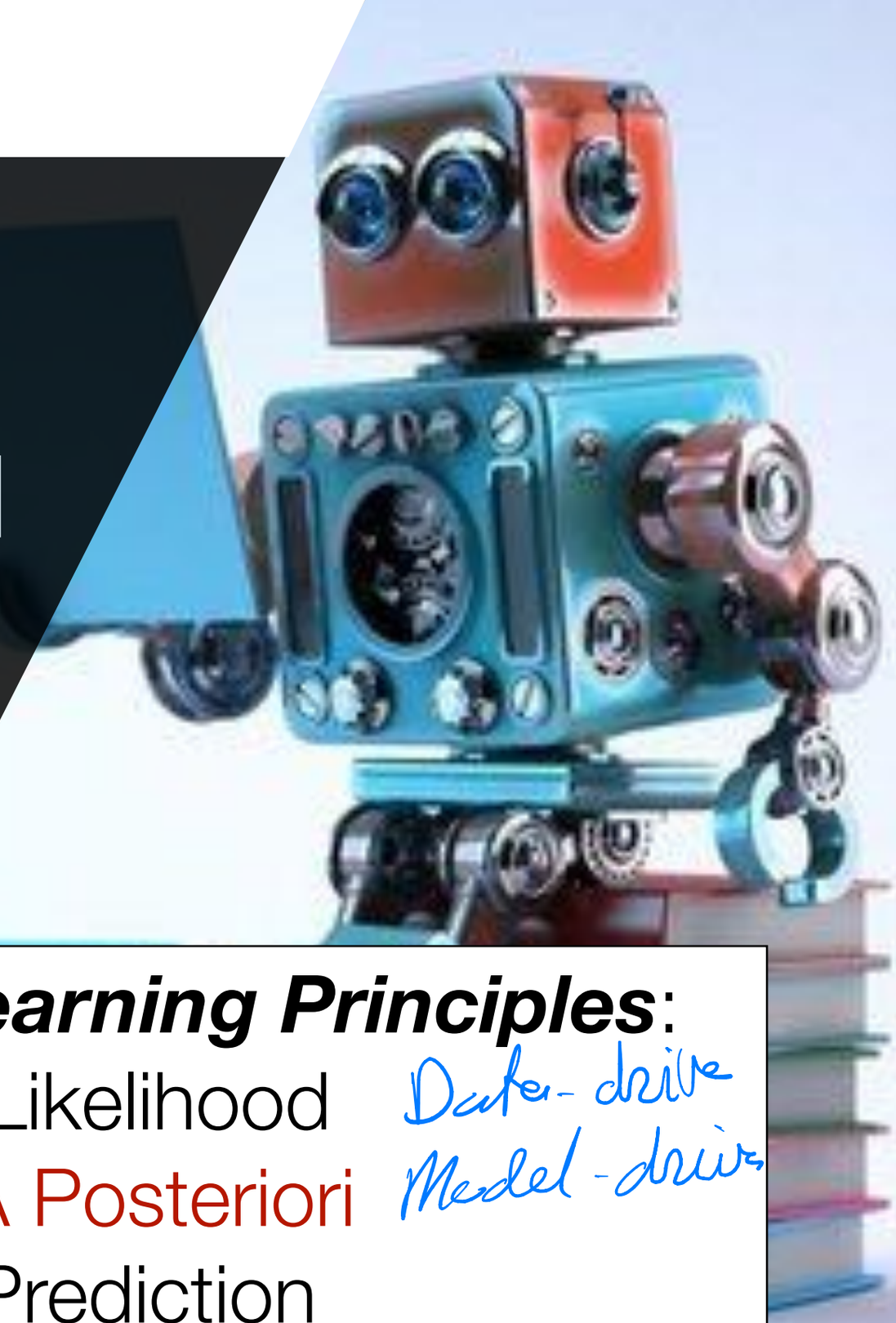
AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 2.5 - Maximum A Posteriori

Erik Bekkers

(Bishop 1.2.5 - 1.2.6)



Three Statistical Learning Principles:

Maximum Likelihood

Maximum A Posteriori

Bayesian Prediction

Data-driven

Model-driven

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

The posterior

- Dataset $D = (x_1, x_2, \ldots, x_N)$ of N independent observations.
- Model the data distribution with some $p(D \mid \mathbf{w})$ parametrized by $\mathbf{w}$
- MAP estimate: choose most probable $\mathbf{w}$ given the data.

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w} \mid D)$$

- With posterior $p(\mathbf{w} \mid D)$ obtained via Bayes' rule

$$p(\mathbf{w} \mid D) = \frac{p(D \mid w)\, p(w)}{p(D)}$$

Maximum A Posteriori Estimates

- Dataset $D = (x_1, x_2, \ldots, x_N)$ of N independent observations.
- Model the data distribution with some $p(D \mid \mathbf{w})$ parametrized by $\mathbf{w}$
- MAP estimate: choose most probable $\mathbf{w}$ given the data.

$$\mathbf{w}_{MAP} = \operatorname{argmax}_{\mathbf{w}} p(\mathbf{w} \mid D)$$

Maximum A Posteriori Estimates

- Dataset $D = (x_1, x_2, \ldots, x_N)$ of N independent observations.
- Model the data distribution with some $p(D \mid \mathbf{w})$ parametrized by $\mathbf{w}$
- MAP estimate: choose most probable $\mathbf{w}$ given the data.

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w} \mid D)$$

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, \frac{p(D \mid \mathbf{w})\, p(\mathbf{w})}{p(D)}$$

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(D \mid \mathbf{w})\, p(\mathbf{w})$$

Maximum A Posteriori Estimates

- Dataset $D = (x_1, x_2, \ldots, x_N)$ of N independent observations.
- Model the data distribution with some $p(D \mid \mathbf{w})$ parametrized by $\mathbf{w}$
- MAP estimate: choose most probable $\mathbf{w}$ given the data.

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w} \mid D)$$

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, \frac{p(D \mid \mathbf{w})\, p(\mathbf{w})}{p(D)}$$

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(D \mid \mathbf{w})\, p(\mathbf{w})$$

$\log$ trick

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmin}} - \log p(D \mid \mathbf{w}) \; - \; \log p(\mathbf{w})$$

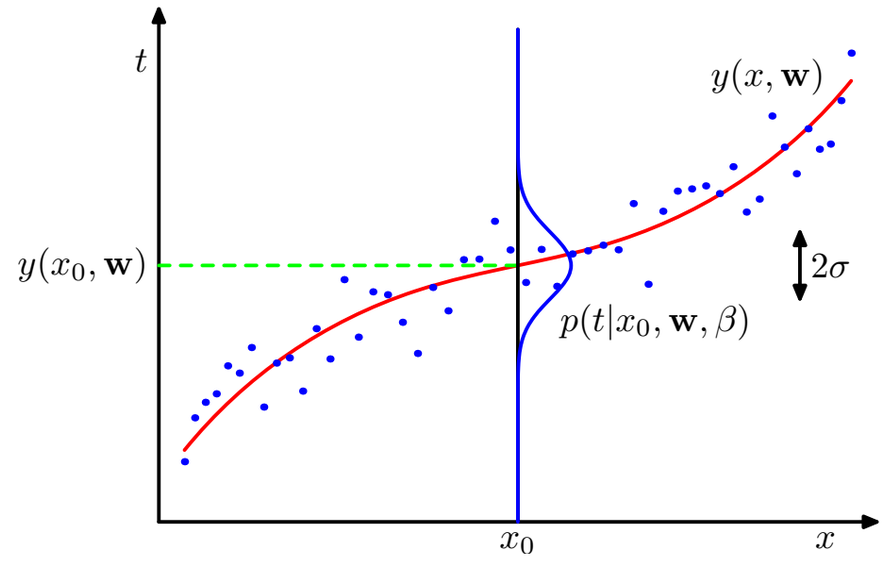
neg. log likelihood — log prior

MAP - Example Gaussian distribution

- **Likelihood**/Data model:

$$p(t \mid x, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(x, \mathbf{w}), \beta^{-1})$$

$$= \sqrt{\frac{\beta}{2\pi}} \exp\left[-\frac{\beta}{2}(t - y(x, \mathbf{w}))^2\right]$$



$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \cdots & \cdots & \cdots \end{pmatrix}$$

covariance $[W_1, W_2] = 0$

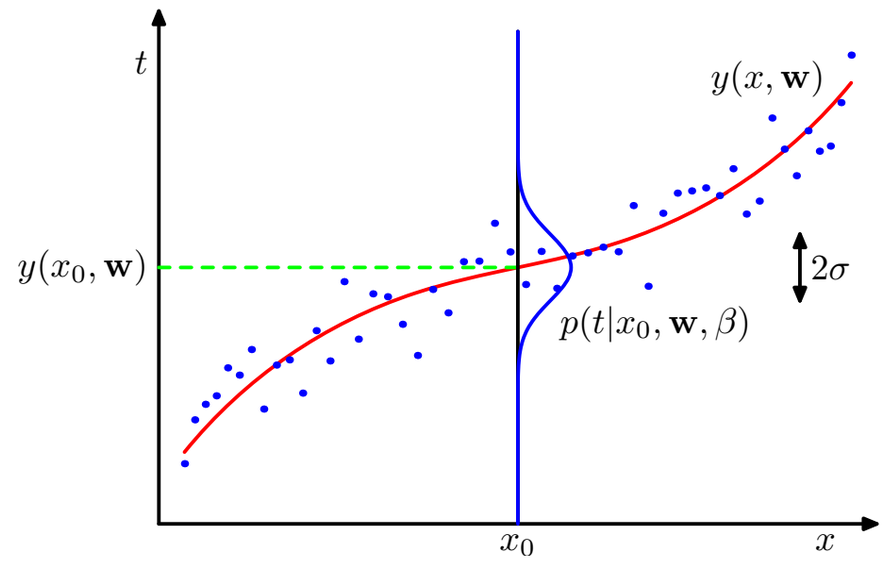
$$\Sigma = \begin{pmatrix} \Sigma_{11} & 0 & 0 \\ 0 & \Sigma_{22} & 0 \\ 0 & 0 & \Sigma_{33} \end{pmatrix}$$

$$= \begin{pmatrix} \alpha^{-1} & 0 & 0 \\ 0 & \alpha^{-1} & 0 \\ 0 & 0 & \alpha^{-1} \end{pmatrix} = \alpha^{-1} Id$$

MAP - Example Gaussian distribution

- **Likelihood**/Data model:

$$p(t \mid x, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(x, \mathbf{w}), \beta^{-1})$$
$$= \sqrt{\frac{\beta}{2\pi}} \exp\left[-\frac{\beta}{2}(t - y(x, \mathbf{w}))^2\right]$$



- **Prior** (assumption: $\mathbf{w}$ takes on small values with certainty α):

$$p(\mathbf{w} \mid \alpha) = \prod_{i=1}^{M} \mathcal{N}(w_i \mid 0, \alpha^{-1})$$
$$= \left(\frac{\alpha}{2\pi}\right)^{M/2} \prod_{i=1}^{M} e^{-\frac{\alpha}{2} w_i w_i}$$
$$= \left(\frac{\alpha}{2\pi}\right)^{M/2} e^{-\frac{\alpha}{2} \mathbf{w}^T \mathbf{w}}$$

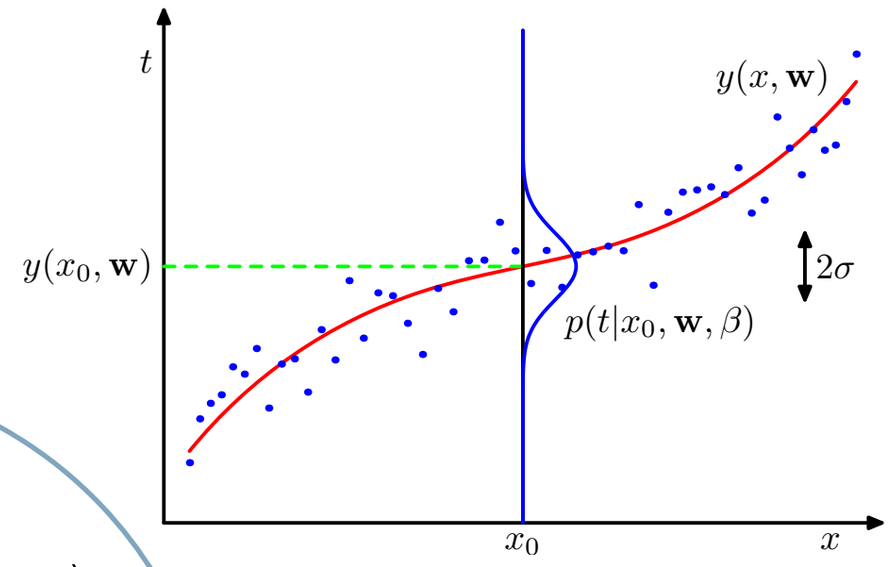
- **MAP solution** obtained by solving:

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmin}} -\log p(D \mid \mathbf{w}, \beta) - \log p(\mathbf{w}, \alpha)$$

MAP - Example Gaussian distribution

- **Likelihood**/Data model:

$$p(t \mid x, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(x, \mathbf{w}), \beta^{-1})$$
$$= \sqrt{\frac{\beta}{2\pi}} \exp\left[-\frac{\beta}{2}(t - y(x, \mathbf{w}))^2\right]$$



- **Prior** (assumption: $\mathbf{w}$ takes on small values with certainty α):

$$p(\mathbf{w} \mid \alpha) = \prod_{i=1}^{M} \mathcal{N}(w_i \mid 0, \alpha^{-1})$$
$$= \left(\frac{\alpha}{2\pi}\right)^{M/2} \prod_{i=1}^{M} e^{-\frac{\alpha}{2} w_i w_i}$$
$$= \left(\frac{\alpha}{2\pi}\right)^{M/2} e^{-\frac{\alpha}{2} \mathbf{w}^T \mathbf{w}}$$

$$\log p(D \mid \mathbf{w}, \beta) = -\frac{\beta}{2} \sum_{i=1}^{N} (y(x_i, \mathbf{w}) - t_i)^2 + \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi$$

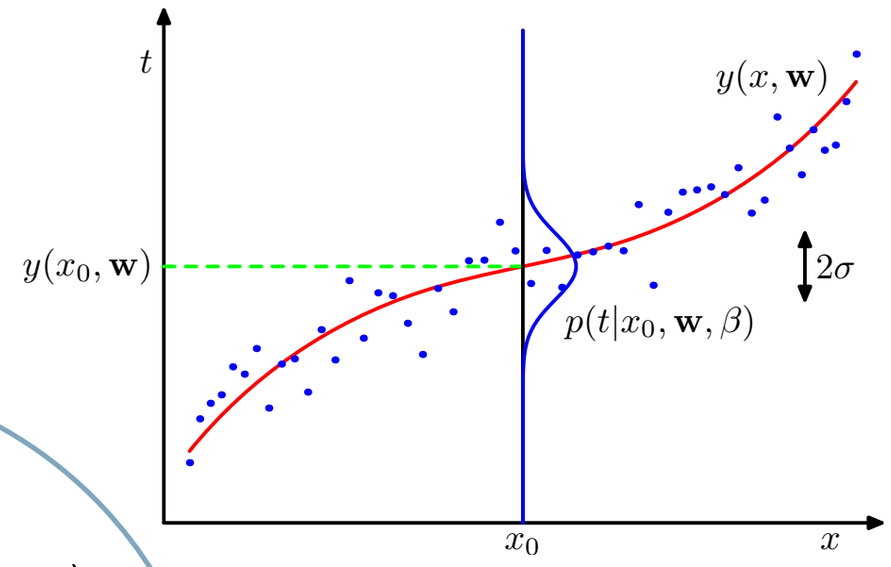
- **MAP solution** obtained by solving:

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmin}} \; \frac{\beta}{2} \sum_{i=1}^{N} (y(x_i, \mathbf{w}) - t_i)^2 - \log p(\mathbf{w}, \alpha)$$

MAP - Example Gaussian distribution

- **Likelihood**/Data model:

$$p(t \mid x, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(x, \mathbf{w}), \beta^{-1})$$
$$= \sqrt{\frac{\beta}{2\pi}} \exp\left[-\frac{\beta}{2}(t - y(x, \mathbf{w}))^2\right]$$



- **Prior** (assumption: $\mathbf{w}$ takes on small values with certainty α):

$$p(\mathbf{w} \mid \alpha) = \prod_{i=1}^{M} \mathcal{N}(w_i \mid 0, \alpha^{-1})$$
$$= \left(\frac{\alpha}{2\pi}\right)^{M/2} \prod_{i=1}^{M} e^{-\frac{\alpha}{2} w_i w_i}$$
$$= \left(\frac{\alpha}{2\pi}\right)^{M/2} e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}}$$

$$\log p(D \mid \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{i=1}^{N}(y(x_i, \mathbf{w}) - t_i)^2 + \frac{N}{2}\log\beta - \frac{N}{2}\log 2\pi$$

$$\log p(\mathbf{w} \mid \alpha) = -\frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \frac{N}{2}\log\alpha - \frac{N}{2}\log 2\pi$$

- **MAP solution** obtained by solving:

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\beta}{2}\sum_{i=1}^{N}(y(x_i, \mathbf{w}) - t_i)^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

University of Amsterdam
Informatics Institute

AMLAB Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 2.6 - Bayesian Prediction

Erik Bekkers

(Bishop 1.2.6)

Three Statistical Learning Principles:

Maximum Likelihood

Maximum A Posteriori

Bayesian Prediction

Slide credits: Patrick Forré and Rianne van den Berg

Bayesian Approach

- Dataset $D = (x_1, x_2, \ldots, x_N)$ of N independent observations.

- Frequentist approach: search for one optimal estimate of $\mathbf{w}$

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(D \mid \mathbf{w})$$

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w} \mid D)$$

} point estimate

- Bayesian approach: Given a prior belief over $\mathbf{w}$, $p(\mathbf{w})$, and our data D, we are interested in the entire posterior distribution

$$p(\mathbf{w} \mid D) = \frac{p(D \mid \mathbf{w})\, p(\mathbf{w})}{p(D)}$$

Here we do not pick one value for $\mathbf{w}$ but consider all of them! We will weigh each model parameterized with $\mathbf{w}$ according to their posterior probability

- $p(\mathbf{w} \mid D)$ reflects the plausibility of different $\mathbf{w}$, given our prior knowledge and how likely our data is generated using $\mathbf{w}$.

- We consider both x and $\underline{w}$ random variables according to joint prob $p(x', \underline{w} | D)$. I.e. $p(x', \underline{w} | D)$ gives the prob. of observing a new x' and model params $\underline{w}$ given the already observed dataset D.
- In the predictive setting we are only interested in x', and thus we compute the marginal $p(x' | D)$ by integrating out $\underline{w}$.
- Following the product rule $p(x', \underline{w} | D) = p(x' | \underline{w}, D)\, p(\underline{w} | D)$ the marginal get's the interpretation of a weighted average sum/integral. Weighting each predictive model $p(x' | \underline{w})$ with the posterior $p(\underline{w} | D)$ for that $\underline{w}$.

conditional

Three Statistical Learning Principles

Three general statistical learning principles to go from data to models (parametric predictive/proposal distributions):

I. Maximum likelihood

II. Maximum a posteriori

III. Bayesian prediction

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 3.1 - Supervised Learning

Linear Regression With Basis Functions

Erik Bekkers

(Bishop 3.1)

Slide credits: Patrick Forré and Rianne van den Berg

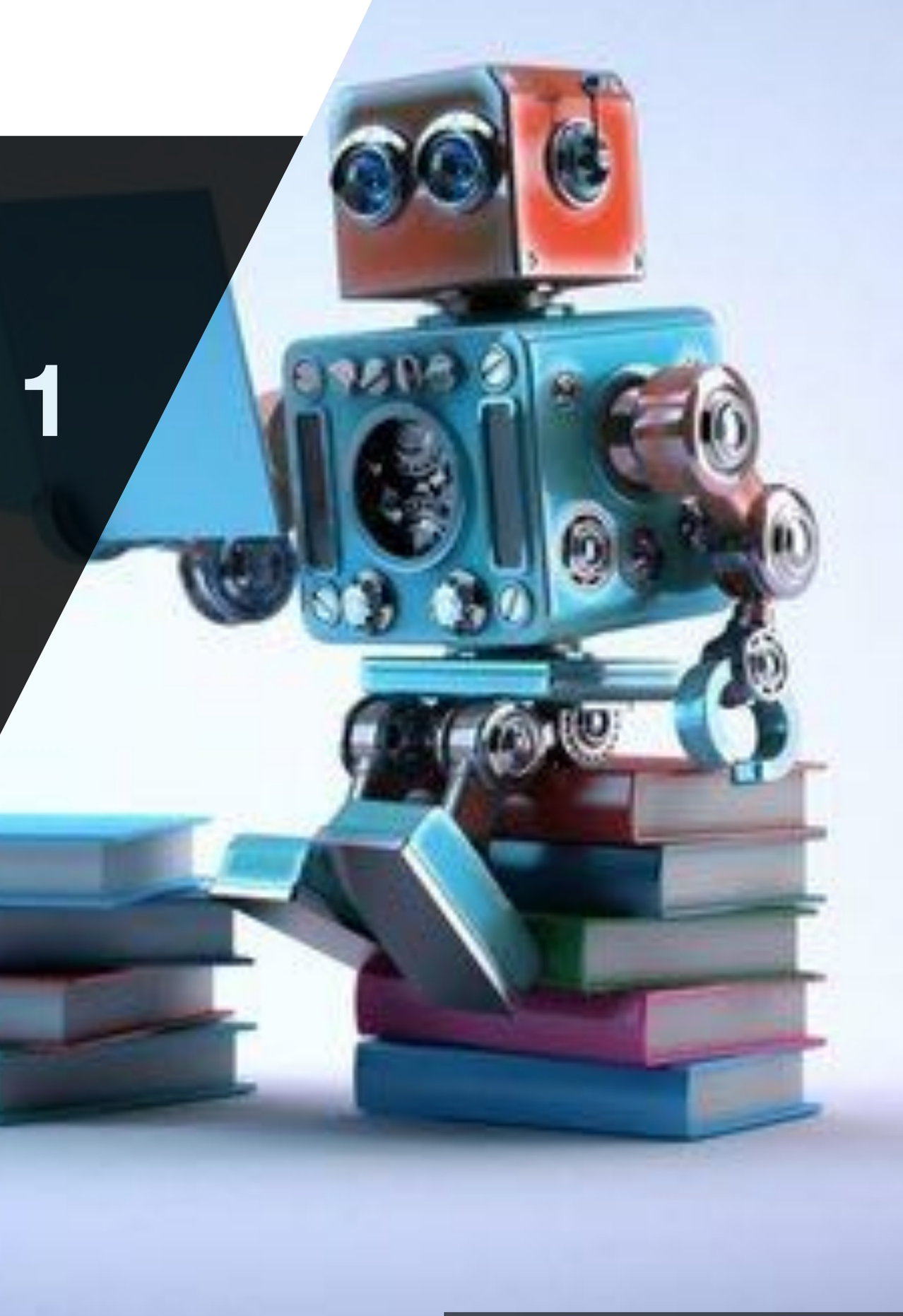


Image credit: Kirillm | Getty Images

University of Amsterdam
Informatics Institute

AMLAB Amsterdam Machine Learning Lab

Ma

Lecture

Linear R

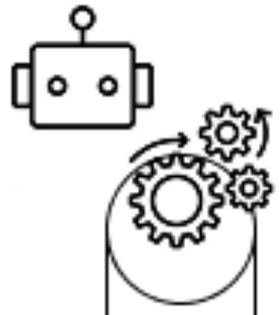
Erik Bek

(Bishop

Machine Learning 1

The probabilistic view

The algorithmic view



W7 Kernel methods

W6 Continuous Latent variable models

W5 Discrete Latent variable models

W4 Neural networks

W3 Classification

W2 Regression

W1

Probabilistic modeling	Optimization	Mathematical tools
Random variables, distributions, maximum likelihood, MAP, ...	Exact/analytic, (stochastic) gradient descent, constraint optimization, 2nd order	Multi-variate calculus, linear algebra

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Linear Regression

- Regression on dataset $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$
  - Input variables $\mathbf{x}_n \in \mathbb{R}^d$
  - Target variables $t_n \in \mathbb{R}$

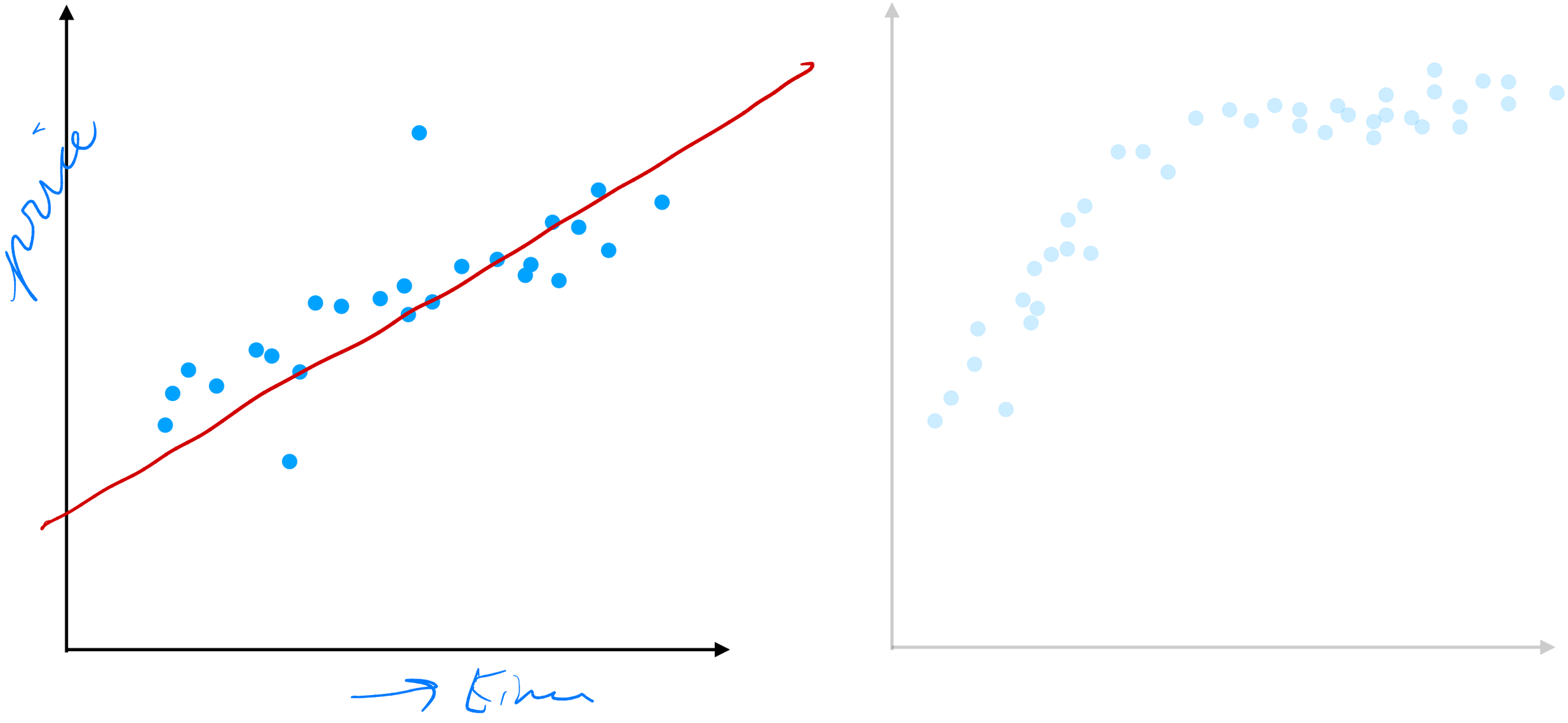
$$\underline{X} = \begin{pmatrix} x_{n1} \\ x_{n2} \\ x_{n3} \\ \vdots \\ x_{nd} \end{pmatrix} \in \mathbb{R}^d, \quad \underline{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}$$

- Simplest linear model:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 \cdot x_2 + \ldots + w_D x_D$$

bias

Linear Regression



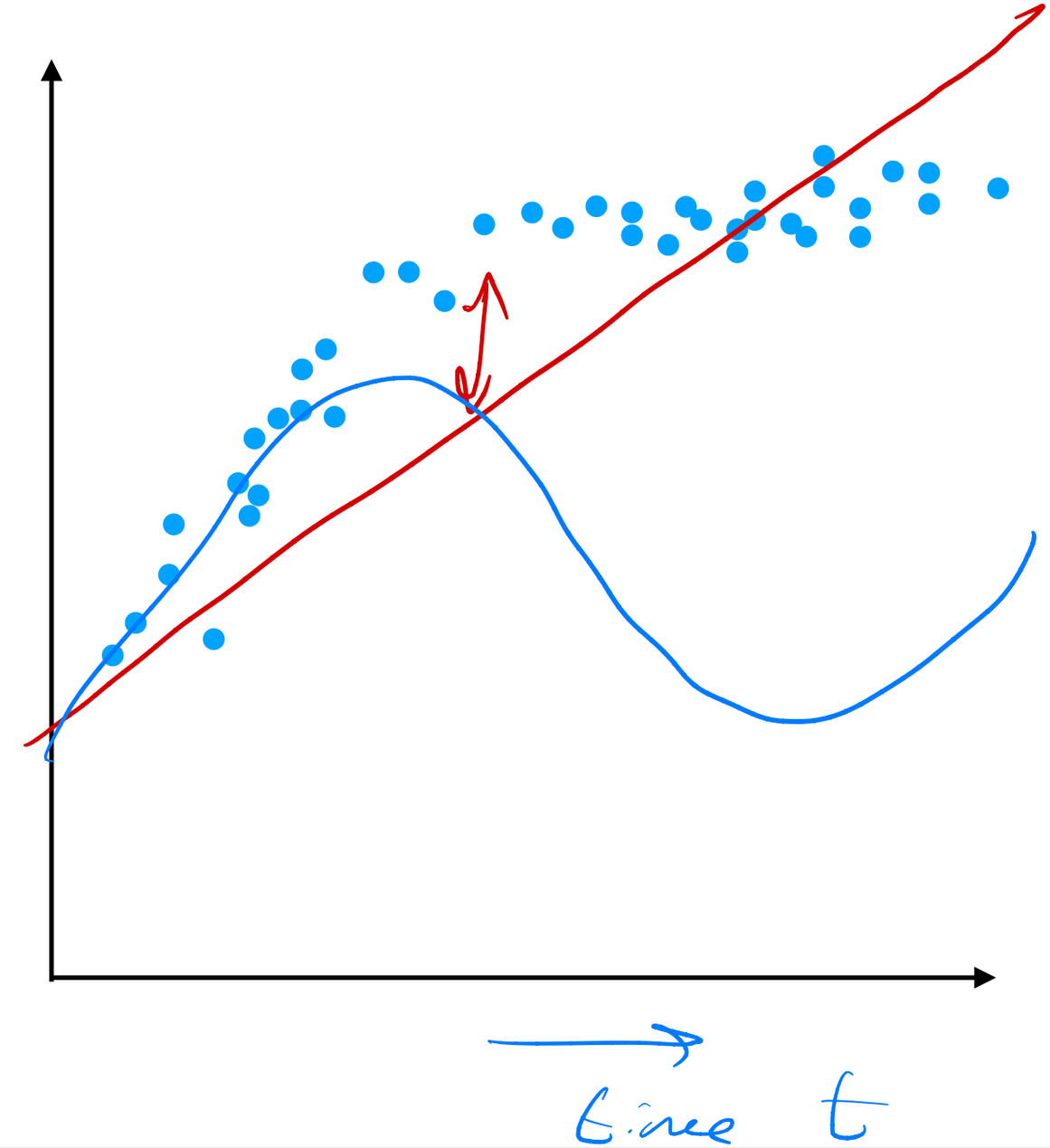
Linear Regression

$\phi_1(t) = t$

$\phi_2(t) = t^2$

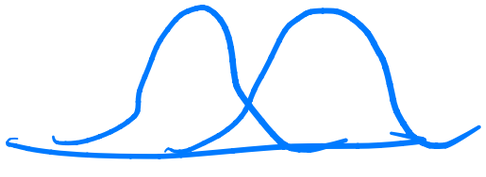
$\phi_3(t) = \sin(t)$

$y(t) = w_1 \cdot t + w_2 \cdot t^2 + w_3 \sin(t)$



Example: Basis Functions (II)

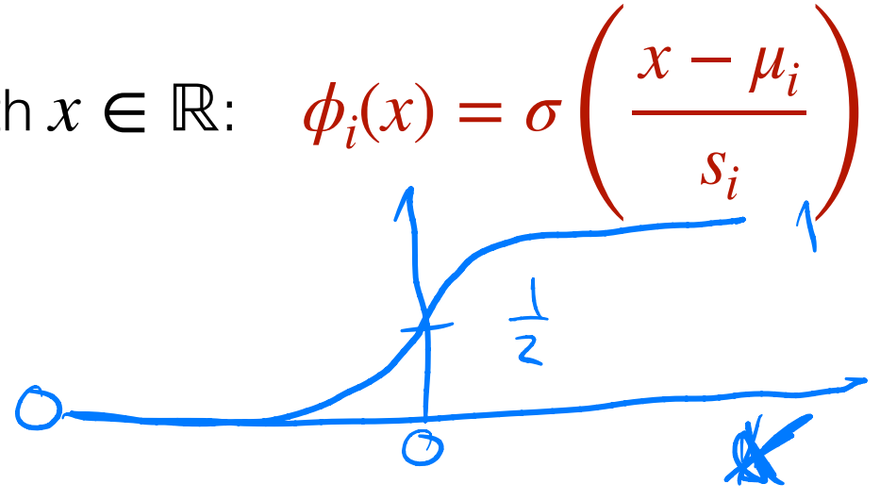
- Gaussian basis functions with $\mathbf{x} \in \mathbb{R}^D$: $\phi_i(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right)$



$$y(\mathbf{x},\mathbf{w}) = w_0 + \sum_{i=1}^{M-1} w_i \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right)$$

- Logistic sigmoid functions with $x \in \mathbb{R}$: $\phi_i(x) = \sigma\left(\frac{x-\mu_i}{s_i}\right)$

with $\sigma(x) = \frac{1}{1+e^{-x}}$



$$y(\mathbf{x},\mathbf{w}) = w_0 + \sum_{i=1}^{M-1} w_i \sigma\left(\frac{x-\mu_i}{s_i}\right)$$

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 3.2 - Supervised Learning

Linear Regression via Maximum Likelihood Optimization

Erik Bekkers

(Bishop 3.1.1)

Slide credits: Patrick Forré and Rianne van den Berg

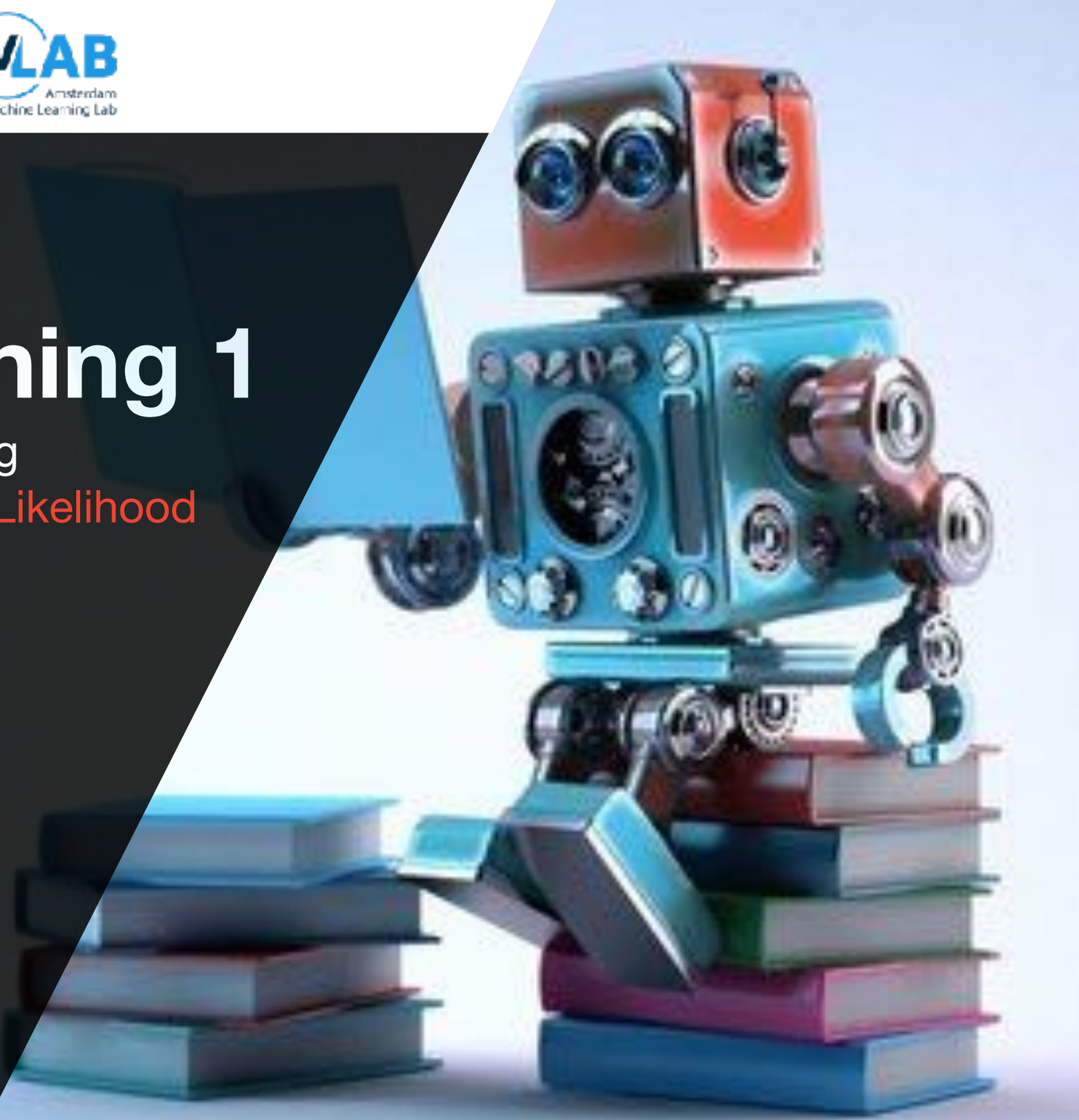


Image credit: Kirillm | Getty Images

Linear Regression

- Regression $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$
  - Input variables $\mathbf{x}_i \in \mathbb{R}^D$
  - Target variables t_i

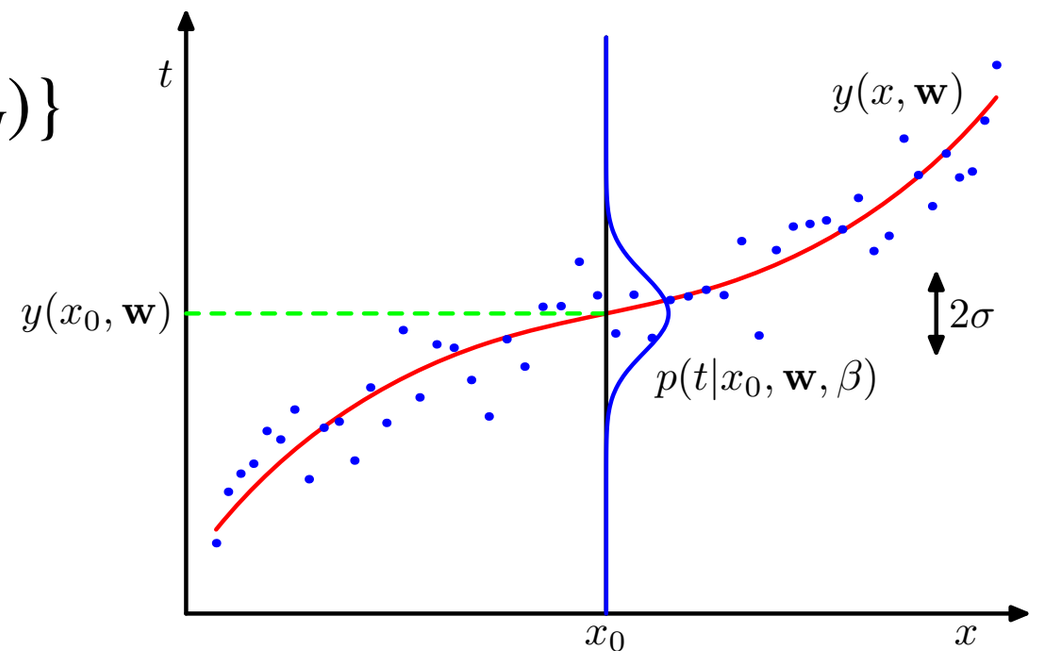


Figure: Gaussian conditional distribution (Bishop 1.16)

- Linear model with basis functions

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

$\mathbb{R}^M$

Maximum Likelihood

- Model assumptions/choices:
  - Data is generated through model $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$
  - Observations t_i are subject to Gaussian noise

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \quad , \quad \varepsilon \sim \mathcal{N}(0, \beta^{-1})$$

- Then data distribution (likelihood):

$$p(t \,|\, \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t \,|\, y(x, w), \beta^{-1})$$

$$\begin{pmatrix} - & x_1 & - \\ - & x_i & - \\ & i & \end{pmatrix} \in \mathbb{R}^{N \times D}$$

- Dataset: $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$ and $\mathbf{t} = (t_1, \ldots, t_N)^T \in \mathbb{R}^N$
- Likelihood function:

$$p(\mathbf{t} \,|\, \mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \sqrt{\frac{\beta}{2\pi}} e^{-\frac{\beta}{2}(t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2}$$

ML: Sum-of-Squares Error

- Likelihood:

$$p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(t_i \mid \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i), \beta^{-1})$$

- Log likelihood

$$\log p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = \frac{N}{2} \log \beta - \frac{N}{2} \log 2\pi - \frac{\beta}{2} \sum_{i=1}^{N} (t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2$$

- Sum-of-squares error:

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2$$

- For comparison of different dataset sizes N

$$E_D^{RMSE}(\mathbf{w}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2}$$

no dependency on dataset size N

sensible units, e.g. error in euros instead of squared euros

Example: Sum-of-Squares Error

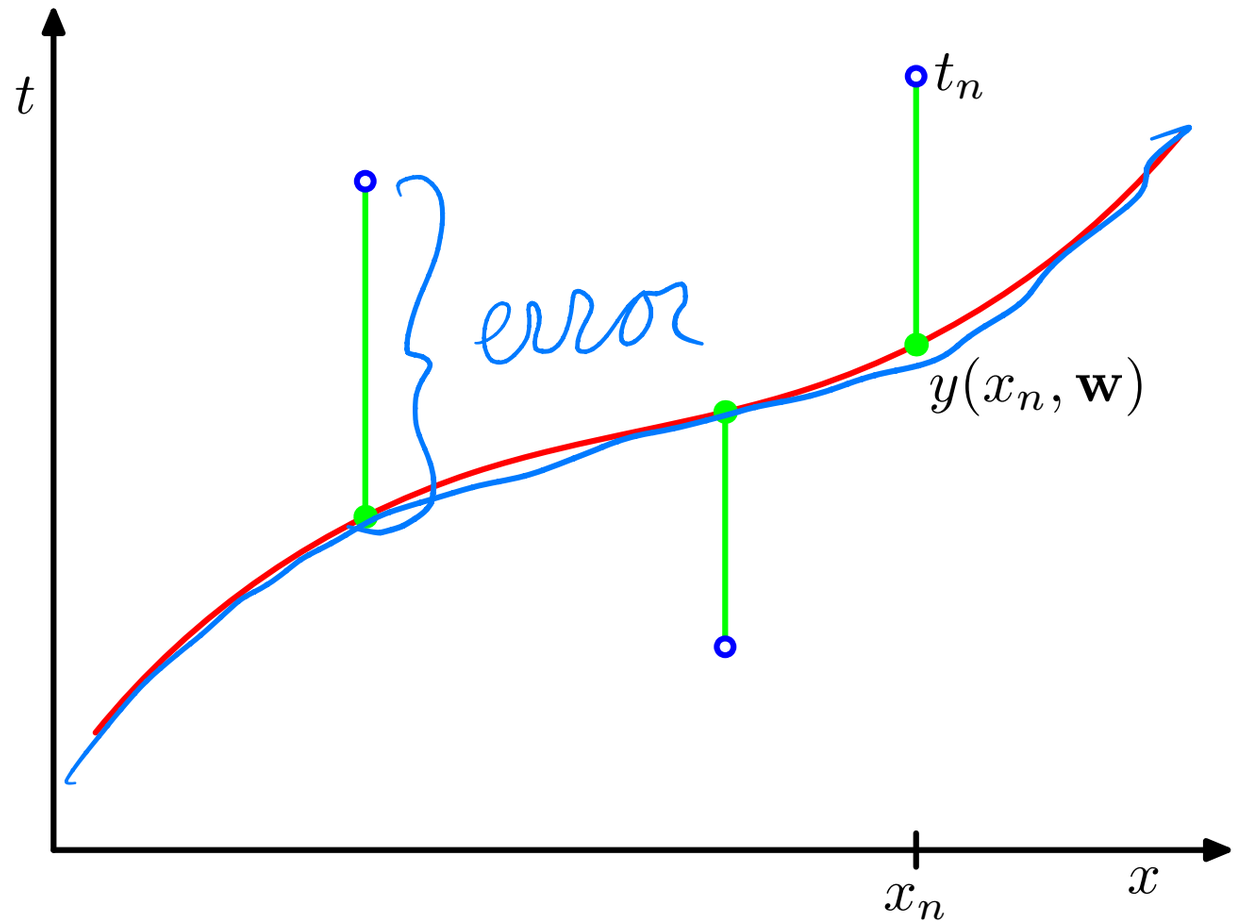


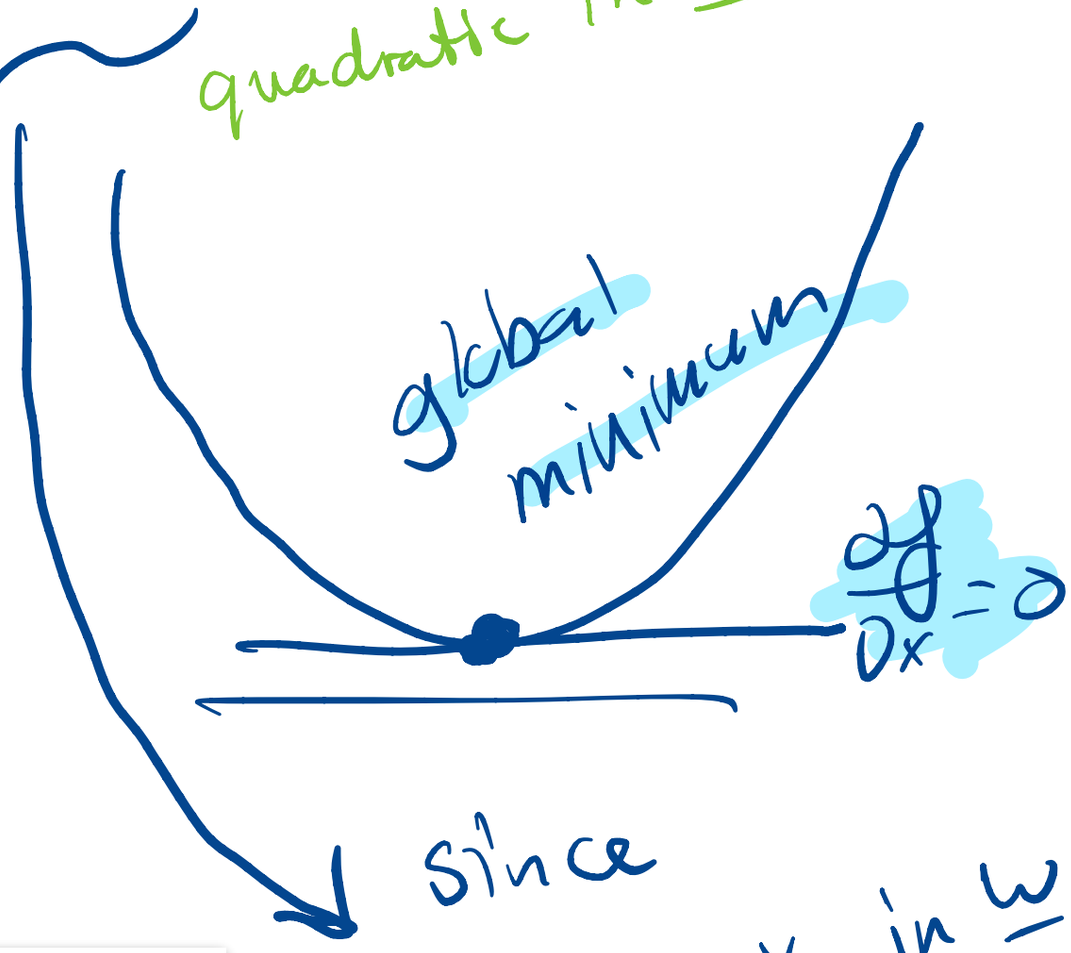
Figure: Errors are given by half the squares of green bars (Bishop 1.3)

Maximum Likelihood Estimates

- Maximize the log likelihood $\log p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta)$
- Analytic solution by solving: $\frac{\partial}{\partial \mathbf{w}} \log p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = 0$ for $\mathbf{w} \in \mathbb{R}^M$

$$\frac{\partial}{\partial \mathbf{w}} \log p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = -\beta \frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} \{t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\}^2 = \mathbf{0}^T$$

quadratic in $\mathbf{w}$ $\Rightarrow$ convex



Convention $\nabla_{\mathbf{x}} f(\mathbf{x}) := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} := \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots \right)$ is a row vector

Maximum Likelihood Estimates

- Maximize the log likelihood $\log p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta)$
- Analytic solution by solving: $\frac{\partial}{\partial \mathbf{w}} \log p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = 0$ for $\mathbf{w}$

$$\frac{\partial}{\partial \mathbf{w}} \log p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = -\beta \frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} \{t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\}^2 = \mathbf{0}^T$$

(β>0, and apply derivative)

$\Leftrightarrow$

$$\sum_{i=1}^{N} \{t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\} \boldsymbol{\phi}(\mathbf{x}_i)^T = \mathbf{0}^T$$

(rearrange terms)

$\Leftrightarrow$

$$\sum_{i=1}^{N} \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) \boldsymbol{\phi}(\mathbf{x}_i)^T = \sum_{i=1}^{N} t_i \boldsymbol{\phi}(\mathbf{x}_i)^T$$

Convention $\nabla_{\mathbf{x}} f(\mathbf{x}) := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} := \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots \right)$ is a row vector

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 3.3 - Supervised Learning

Stochastic Gradient Descent

Erik Bekkers

(Bishop 3.1.3)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Stochastic gradient descent

- For $N \gg 1$ solution $\mathbf{w}_{ML} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}$ is very costly to compute!
  - Needs to process all data $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ at once.
  - Matrix inversion of M x M matrix: $O(M^3)$ Expensive
- Loss is a sum of error terms for each datapoint:
  - $E_D(\mathbf{w}) = \sum_{i=1}^{N} E(\mathbf{x}_i, t_i, \mathbf{w})$
  - $E(\mathbf{x}_i, t_i, \mathbf{w}) = \frac{1}{2}(t_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i))^2$
- Approach for large dataset: stochastic gradient descent

idea approximate E_D with few data points

Recap: The Gradient (see e.g. Bishop App. E)

- The gradient encodes all directional derivatives via scalar product

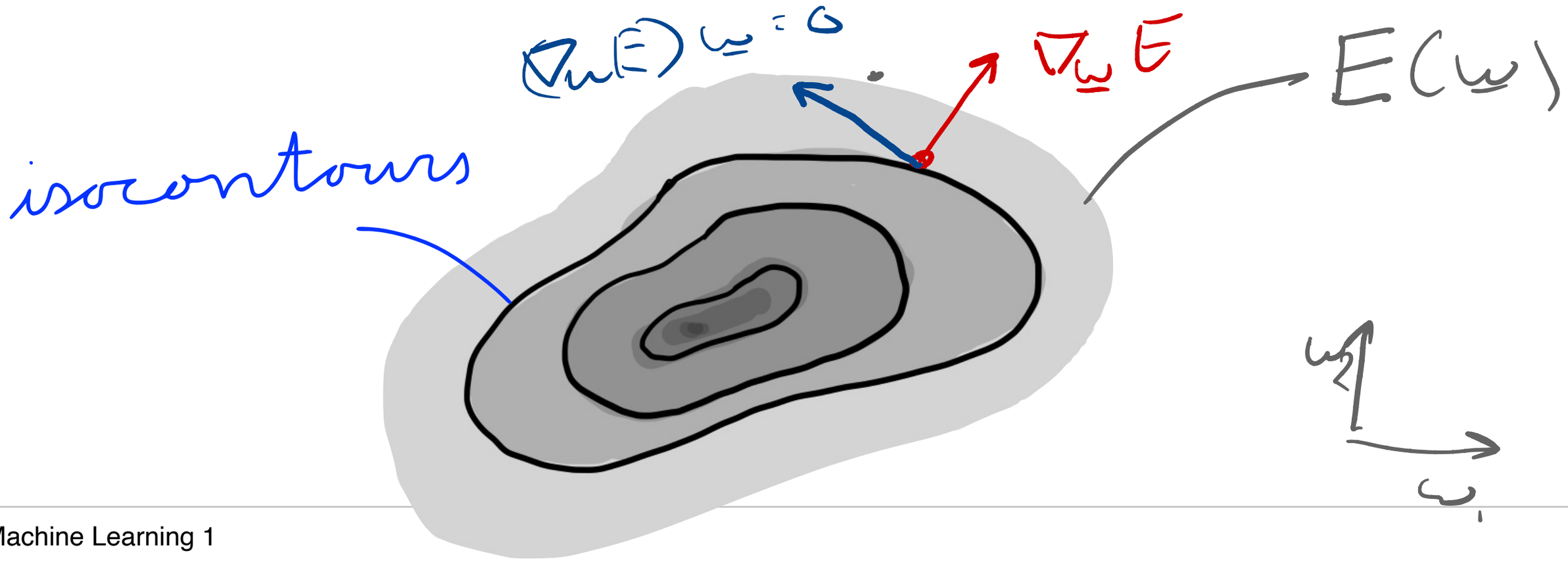
$\nabla_{\mathbf{w}} E := \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots \right)$ a row vector

directional derivative in direction $\mathbf{v}$ is given by $(\nabla_{\mathbf{w}} E)\mathbf{v}$

- The gradient always points in the direction of steepest ascent

directional derivative is maximized when $\mathbf{v} = \nabla_{\mathbf{w}} E$

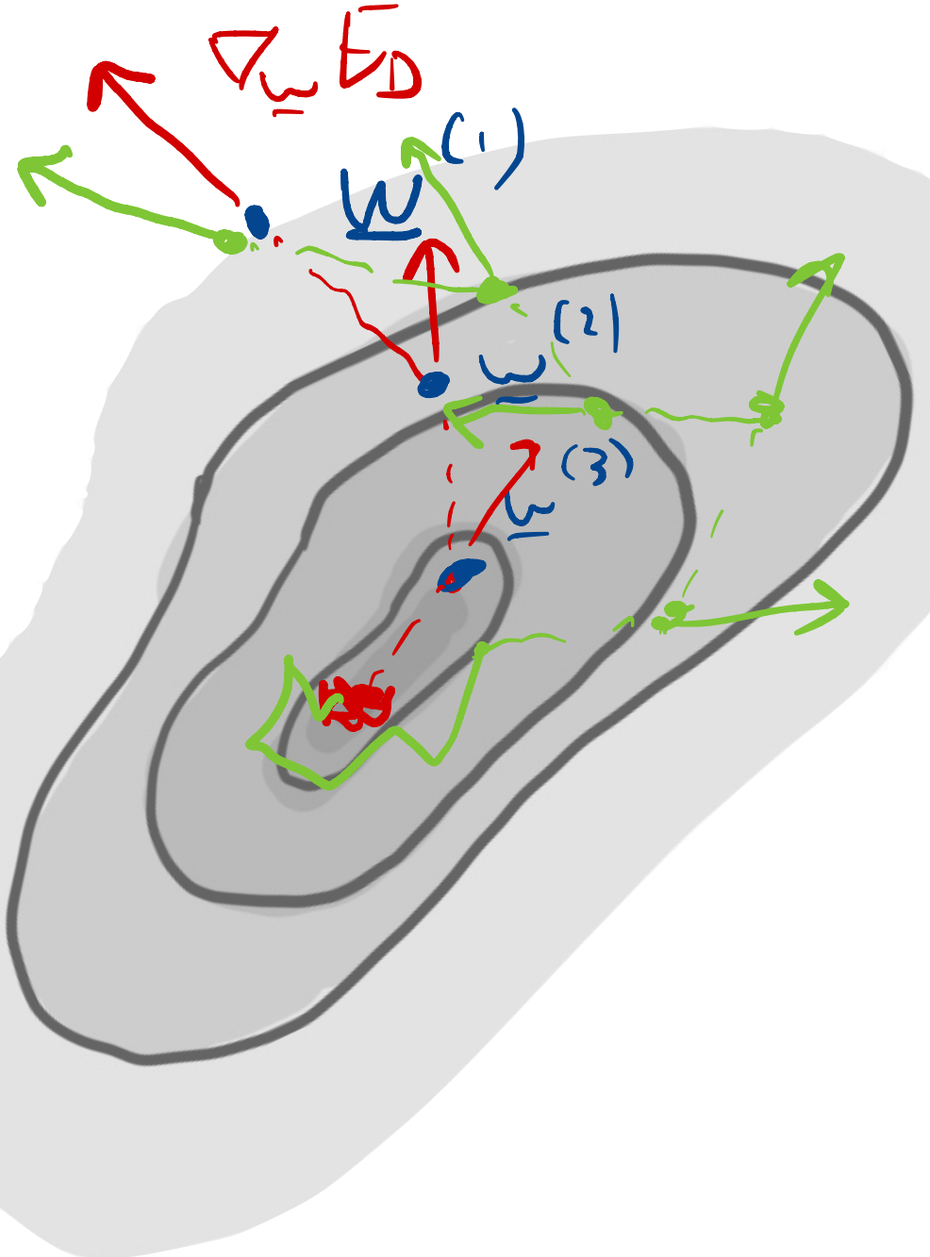
- The gradient is always perpendicular to the contours of a function



Stochastic gradient descent

- Objective: minimize $E_D(\mathbf{w}) = \sum_{i=1}^{N} E(\mathbf{x}_i, t_i, \mathbf{w})$, with $E(\mathbf{x}_i, t_i, \mathbf{w}) = \frac{1}{2}(t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2$

True gradient



- Stochastic gradient descent (SGD):
  - Initialize $\mathbf{w}^{(0)}$, choose learning rate η
  - Iterate over data points, and update

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta(\nabla_{\mathbf{w}} E(\mathbf{x}_i, t_i, \mathbf{w}^{(\tau)}))^T$$

$$= \mathbf{w}^{(\tau)} - \eta(t_i - \boldsymbol{\phi}(\mathbf{x}_i)^T \mathbf{w}^{(\tau)})\boldsymbol{\phi}(\mathbf{x}_i)$$

Often you pick a (mini-) batch of samples B and compute the gradient on this batch instead of a single sample.

  - If $E_D(\mathbf{w})$ is convex in $\mathbf{w}$ and η is small enough: convergence