University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 5 - Classification

Recap/Catch-up:

Bias-Variance

Gaussian Posteriors,

Sequential Bayesian Learning,

Bayesian Predictive Distributions

Equivalent Kernel

Continue with:

~~Bayesian Model Comparison~~

~~Empirical Bayes~~

Classification

Decision Theory

Generative Models

Erik Bekkers

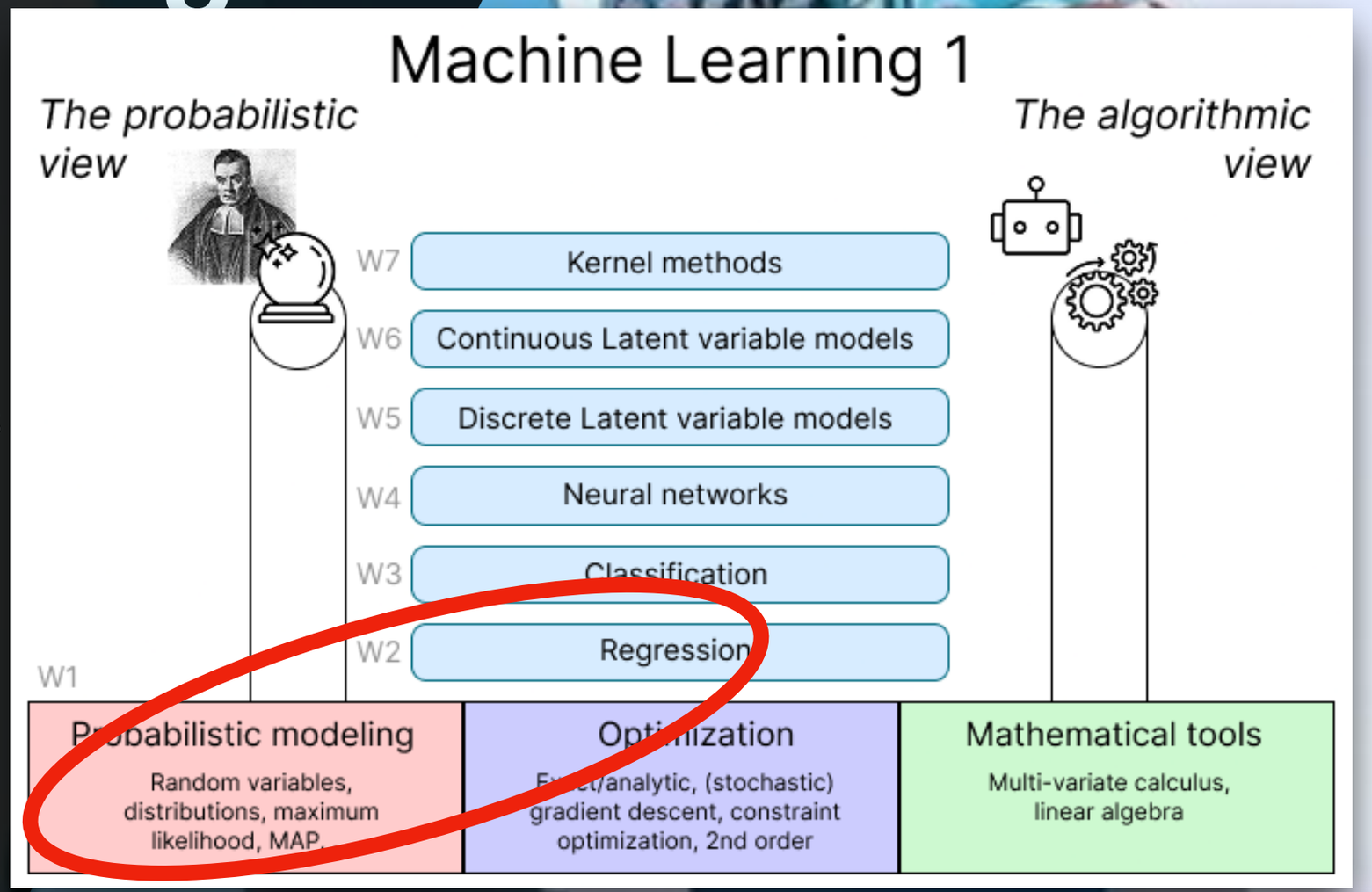


Image credit: Kirillm | Getty Images

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 4.2 - Supervised Learning

Bias Variance Decomposition

Erik Bekkers

(Bishop 1.5.5, 3.2)

Recap

Slide credits: Patrick Forré and Rianne van den Berg

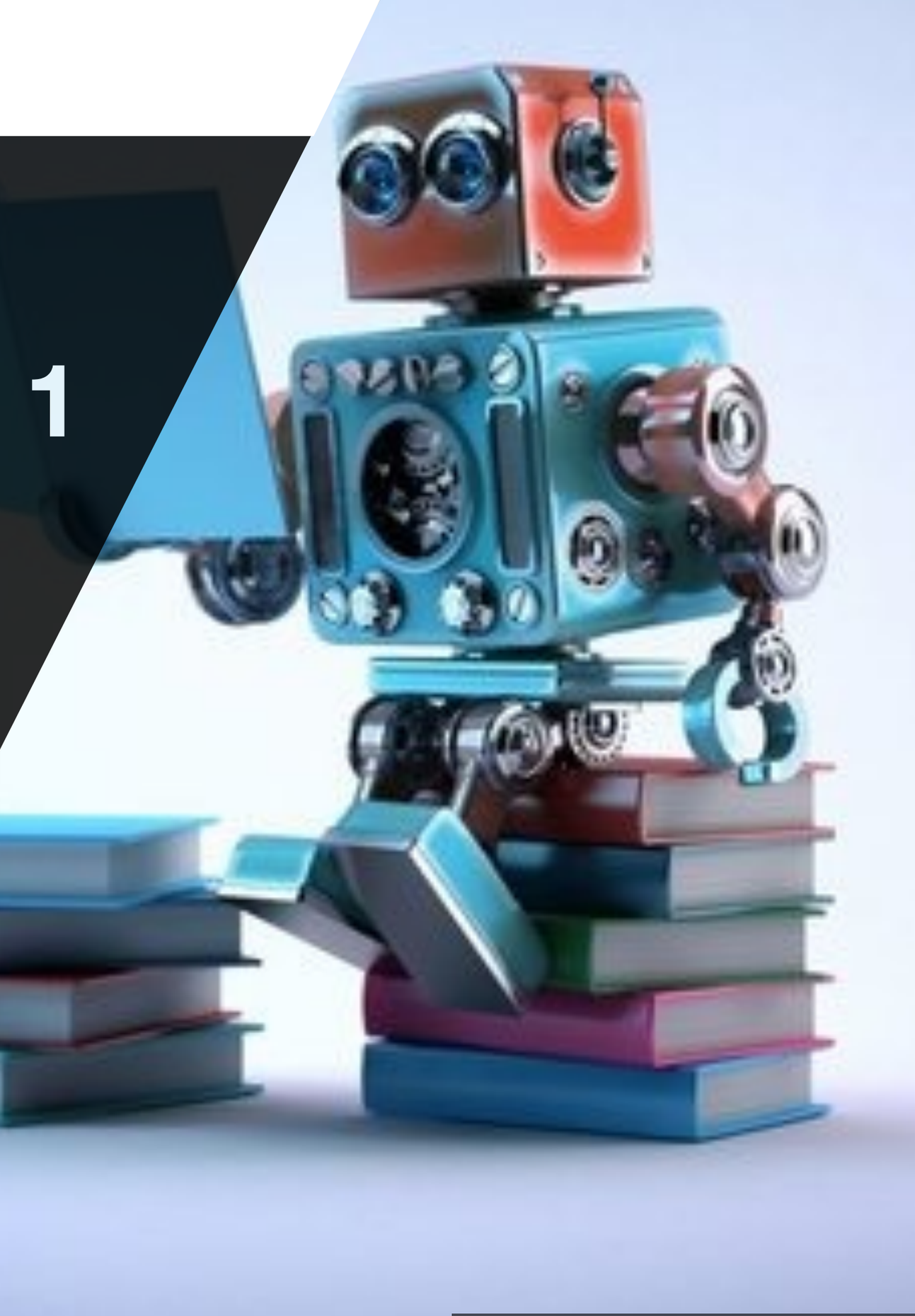


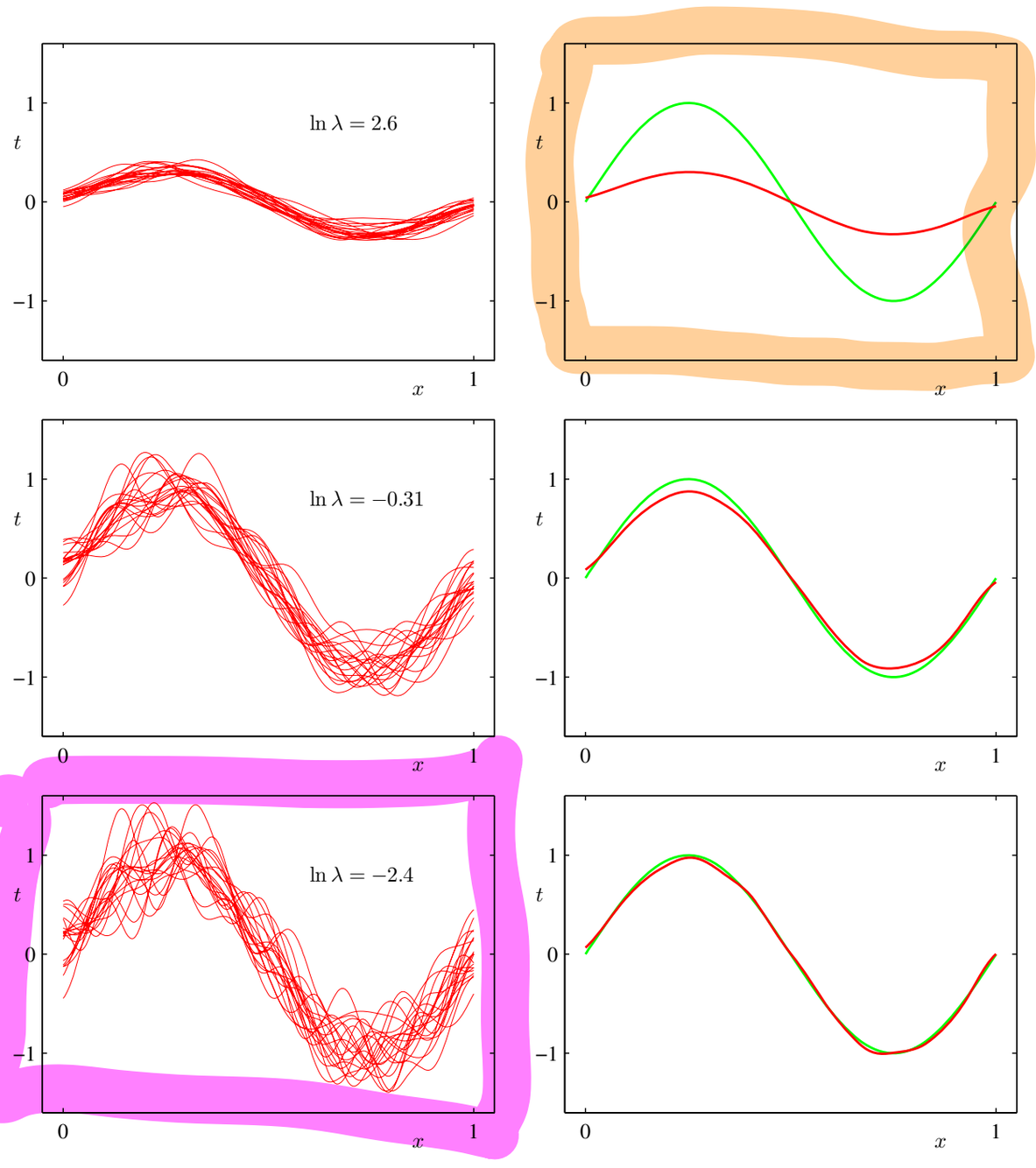
Image credit: Kirillm | Getty Images

Bias-Variance Summary slide

- The bias-variance decomposition is a way to **analyze a model's prediction error**.

- It tells us **in expectation** (over all possible datasets) a model's error consists of three parts:

Expected Loss
= (Bias)² + Variance + Noise



Expected Loss for Regression $\mathbb{E}_{(\mathbf{x},t)\sim p(\mathbf{x},t)}[L(t, y(\mathbf{x})]$

Why do models make errors?
What kind of errors can we expect?

- Consider dataset of observations $(\mathbf{x}, t) \sim p(\mathbf{x}, t)$ and a model $y(\mathbf{x})$

- The model makes errors (regression loss function):

$$L(t, y(\mathbf{x})) = (t - y(\mathbf{x}))^2$$

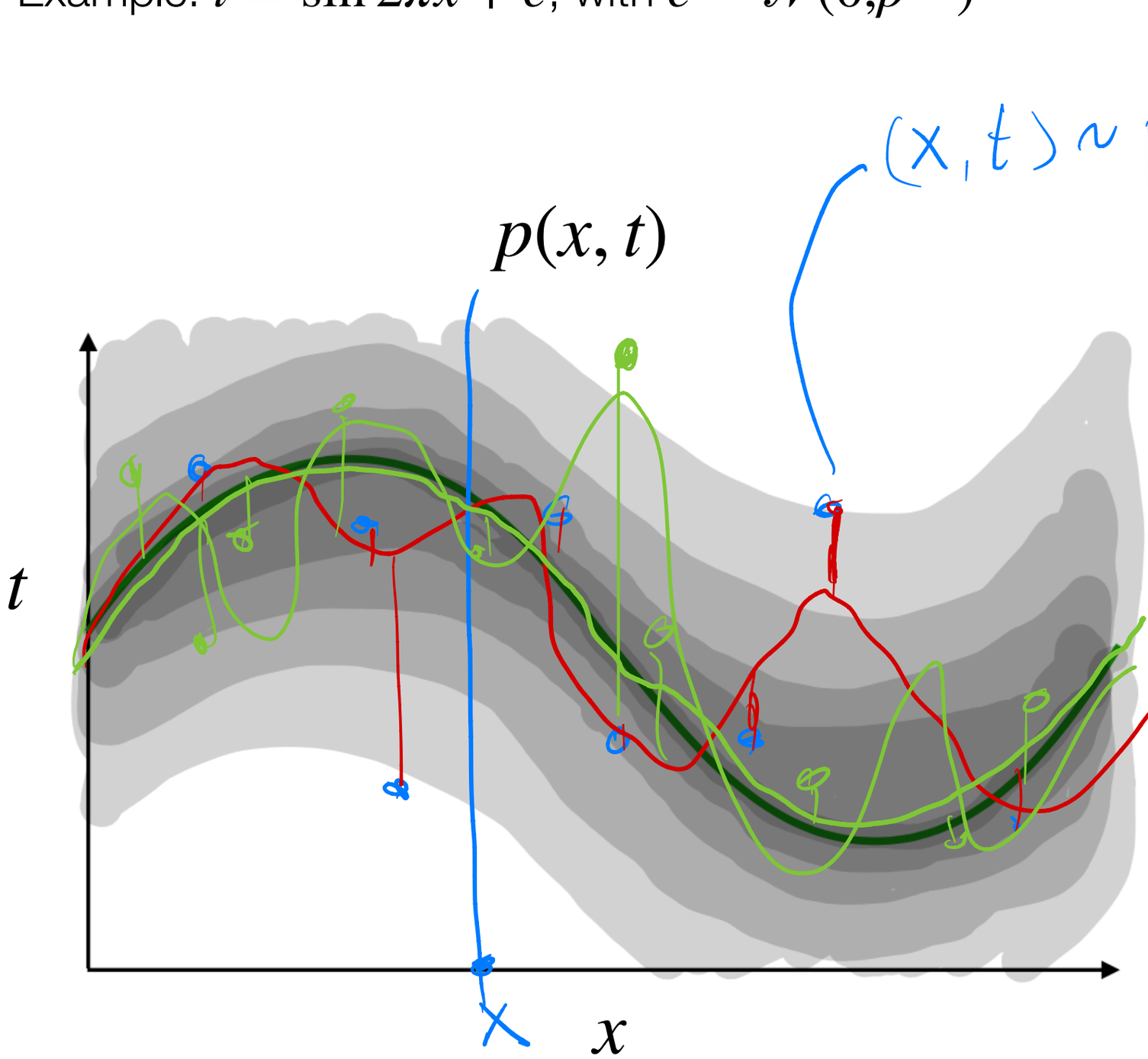
- Every time we make an observation of random variables $(\mathbf{x}, t)$ we make a different error. The best model we can possibly have (Bishop 1.5.5):

$$y(\mathbf{x}) = \mathbb{E}_{t\sim p(t|\mathbf{x})}[t \,|\, \mathbf{x}]$$

- We now consider the expected loss:

$$\mathbb{E}_{(\mathbf{x},t)\sim p(\mathbf{x},t)}[L(t, y(\mathbf{x})] = \underbrace{\int\int (t - y(\mathbf{x}))^2 p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t}_{\text{(Bias)}^2 + \text{Variance} + \text{Noise}}$$

Example: $t = \sin 2\pi x + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \beta^{-1})$



$(x, t) \sim p(x, t)$

$p(t|x)$

$y_{D_2}(x)$

$y_{D_1}(x)$

$\mathbb{E}_{t \sim p(t|x)}[t]$

Regression function

$D_1 = \{(x_i, t_i)\}_{i=1}^{N}$

$D_2 = \{(x_i, t_i)\}_{i=1}^{N}$

Bias-Variance Decomposition

- What can we say about the expected loss of y_D?

- On average (over datasets D) our model y_D will make three types of errors:

$$\mathbb{E}_D[\mathbb{E}[L]] = \int \mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}_D[y_D(\mathbf{x})])^2]\, p(\mathbf{x})\mathrm{d}\mathbf{x} \qquad \text{Variance}$$

$$+ \int (\mathbb{E}_D[y_D(\mathbf{x})] - \mathbb{E}[t\,|\,\mathbf{x}])^2\, p(\mathbf{x})\mathrm{d}\mathbf{x} \qquad \text{Bias}^2$$

$$+ \int \mathrm{var}[t\,|\,\mathbf{x}]\, p(\mathbf{x})\mathrm{d}\mathbf{x} \qquad \text{Noise}$$

→ expected model
average model

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 4.3 - Supervised Learning
Bayesian Linear Regression - Gaussian Posteriors

Erik Bekkers

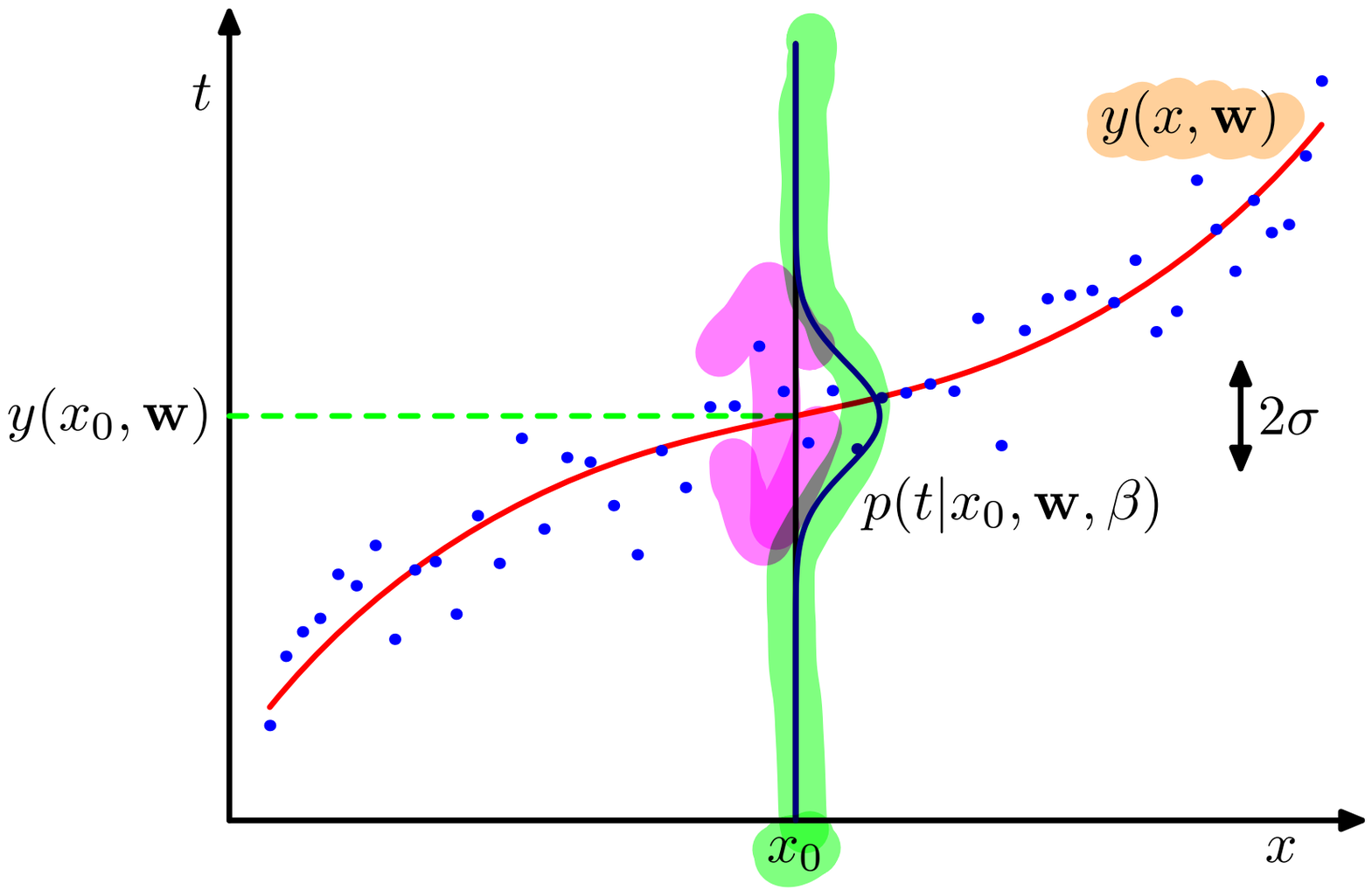
(Bishop 3.3.1 (and 2.3.3)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

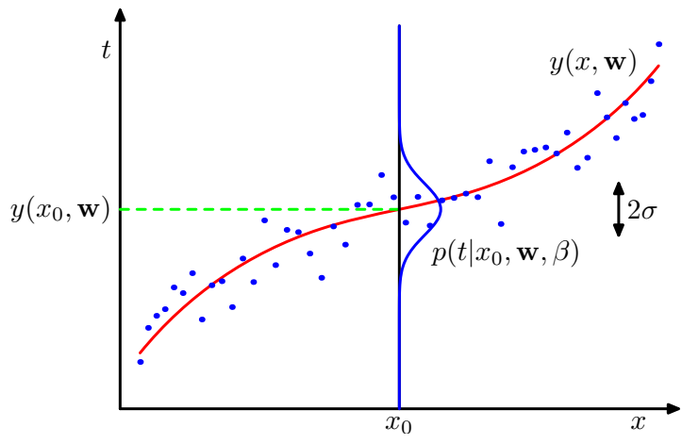
Bayesian Linear Regression

- Regression problem with:
  - Data: $\mathbf{X} = (\mathbf{x}_1, \ldots \mathbf{x}_N)^T$, $\mathbf{t} = (t_1, \ldots, t_N)^T$
  - Predictive distribution $p(t' | \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t' | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}'), \beta^{-1})$



Bayesian Linear Regression

- Regression problem with:
  - Data: $\mathbf{X} = (\mathbf{x}_1, \ldots \mathbf{x}_N)^T$, $\mathbf{t} = (t_1, \ldots, t_N)^T$
  - Predictive distribution $p(t' \,|\, \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t' \,|\, \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}'), \beta^{-1})$



- Probabilistic model with Gaussians:
  - Likelihood: $$p(\mathbf{t} \,|\, \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \,|\, \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) = \mathcal{N}(\mathbf{t} \,|\, \boldsymbol{\Phi}\mathbf{w}, \beta^{-1}\mathbf{I})$$
  - Conjugate prior: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \,|\, \mathbf{m}_0, \mathbf{S}_0)$
  - Posterior: $$p(\mathbf{w} \,|\, \mathbf{t}, \mathbf{X}) = \frac{p(\mathbf{t} \,|\, \mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w})}{p(\mathbf{t} \,|\, \mathbf{X}, \beta)} = \mathcal{N}(\mathbf{w} \,|\, \mathbf{m}_N, \mathbf{S}_N)$$

conjugate here means that the posterior will be of the same family of distributions (Gaussian)

Bishop Ch 2.3, Eq. 2.116

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$
$$\mathbf{m}_N = \mathbf{S}_N (\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \boldsymbol{\Phi}^T \mathbf{t})$$

Bayesian Linear Regression

- Regression problem with:
  - Data: $\mathbf{X} = (\mathbf{x}_1, \ldots \mathbf{x}_N)^T$, $\mathbf{t} = (t_1, \ldots t_N)^T$
  - Predictive distribution: $p(t \mid \mathbf{x}, \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int p(t \mid \mathbf{x}, \mathbf{w}, \beta)\, p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}, \alpha, \beta)\, d\mathbf{w}$
- Probabilistic model with Gaussians:

So, with a **Gaussian likelihood** and **Gaussian prior**
we obtain an
explicit expression for the **posterior**
(which is again Gaussian)

- Likelihood: $$p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) = \mathcal{N}(\mathbf{t} \mid \boldsymbol{\Phi}\mathbf{w}, \beta^{-1}\mathbf{I}))$$
- Conjugate prior: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_0, \mathbf{S}_0)$
- Posterior: $$p(\mathbf{w} \mid \mathbf{t}, \mathbf{X}) = \frac{p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w})}{p(\mathbf{t} \mid \mathbf{X}, \beta)} = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N)$$

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 4.4 - Supervised Learning
Bayesian Linear Regression - Sequential Bayesian Learning

Erik Bekkers

(Bishop 3.3.1)

Recap

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Example: Sequential Bayesian Learning

- Data come in as a sequences of observations of input x, target t
- **Synthetic data generated by**
  - $x \sim \mathcal{U}(x|-1,1)$
  - $t = f(x, \mathbf{a}) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, 0.2^2)$
  - $f(x, \mathbf{a}) = a_0 + a_1 x, \quad a_0 = -0.3, a_1 = 0.5$

$D_{N-1} = \{(x_1, t_1), \ldots, (x_{N-1}, t_{N-1})\}$

size is $N-1$

$D_N = D_{N-1} \cup (x_N, t_N)$

- **Modeling choices**
  - Target distribution: $p(t'|x', \mathbf{w}, \beta) = \mathcal{N}(t'|y(x', \mathbf{w}), \beta^{-1}), \qquad \beta^{-1} =$
  - Linear model: $y(x, \mathbf{w}) = w_0 + w_1 x$
  - Prior: $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}), \qquad \alpha = 2$
- **Sequential Bayesian Learning**: Posterior after $N-1$ observations is prior for arrival of N^{th} datapoint!

$$p(\mathbf{w}|D_N) = \frac{p(\mathbf{x}_N|\mathbf{w})p(\mathbf{w}|D_{N-1})}{p(x_N)}$$

Example: Sequential Bayesian Learning

- Data generated by

$$t = -0.3 + 0.5x + \epsilon$$

$w_0 + w_1 \cdot x + \epsilon$

- Prior

$$p(\mathbf{w}\,|\,\alpha) = \mathcal{N}(\mathbf{w}\,|\,\mathbf{0}, \alpha^{-1}\mathbf{I})$$

- Sample 1 datapoint ★

$$(x_1, t_1)$$

- Likelihood

$$p(t_1\,|\,x_1, \mathbf{w}, \beta) = \mathcal{N}(t_1 \mid w_0 + w_1 x_1, \beta^{-1})$$

- Posterior

$$p(\mathbf{w}\,|\,x_1, t_1, \alpha, \beta) \propto p(t_1 \mid x_1, \mathbf{w}, \beta)\, p(\mathbf{w} \mid \alpha)$$

sample $\underline{w}$ drawn from the posterior

$p(w)$ / $p(w|D)$

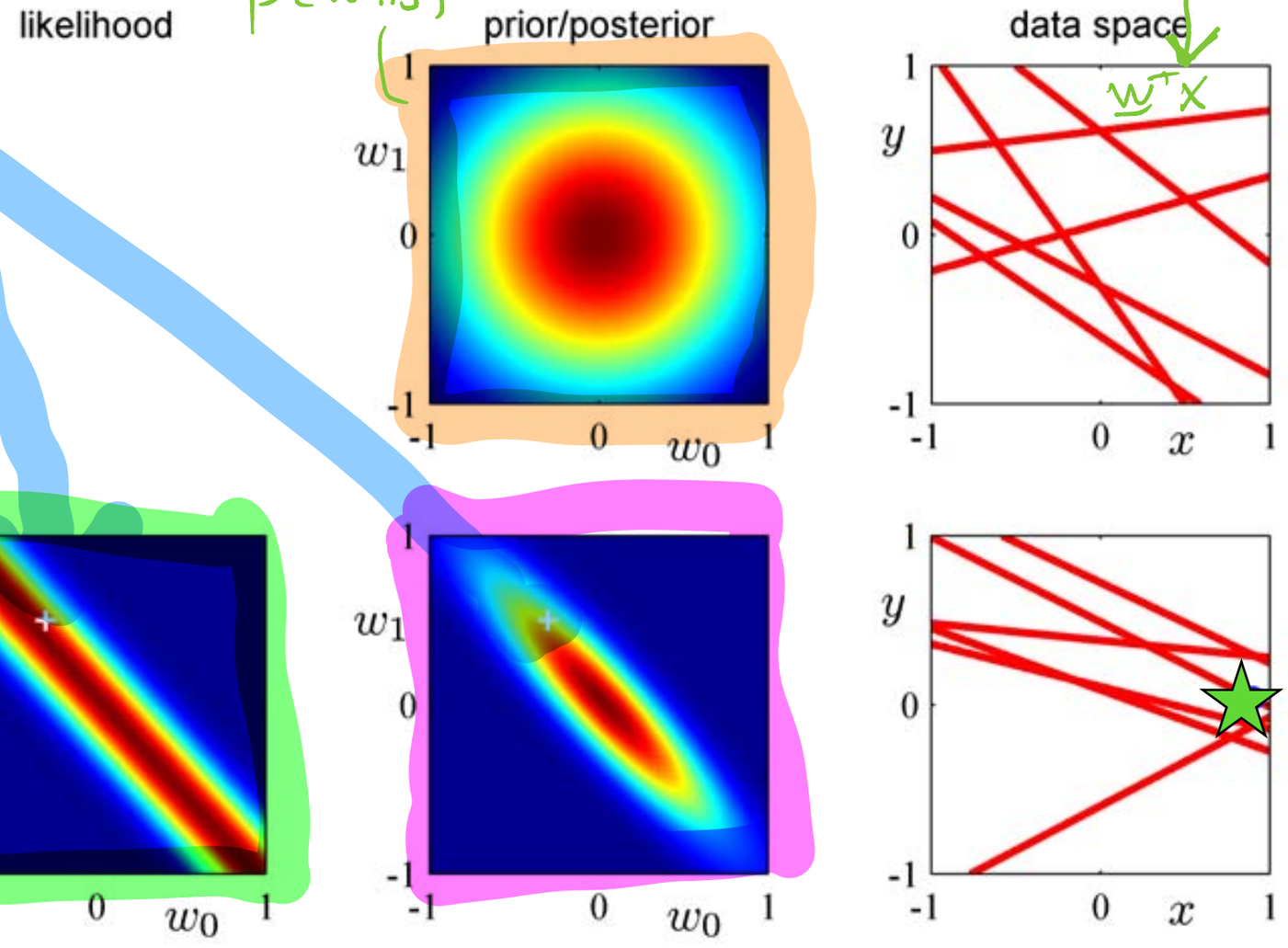


Figure: Sequential Bayesian learning (Bishop 3.7)

proportional to ≡ the same up to a constant

Example: Sequential Bayesian Learning

- Sample 2nd data point ★
- Posterior ⇨ Prior
- Likelihood

$$p(t_2 \mid x_2, \mathbf{w}, \beta)$$

- Posterior

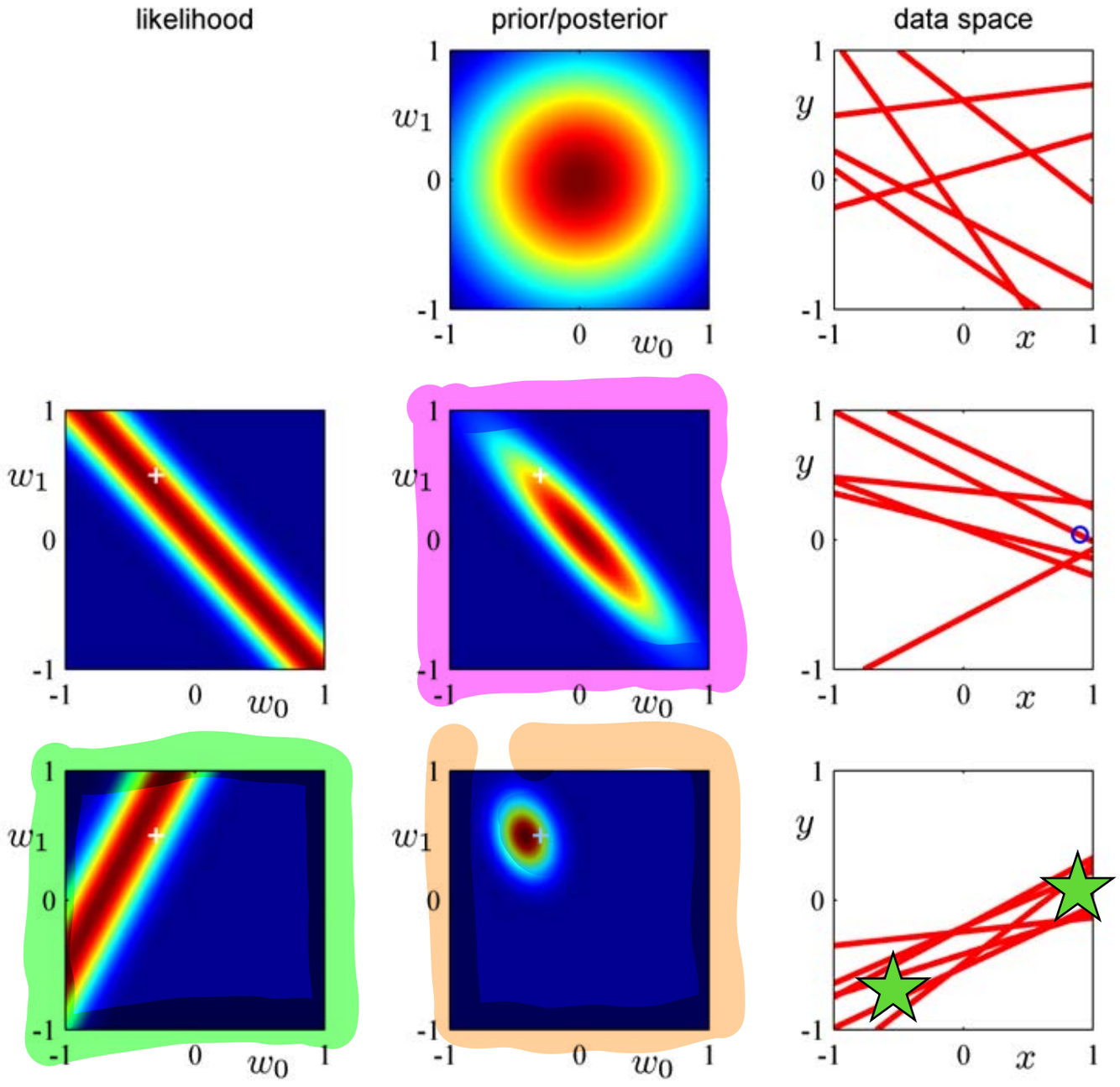


Figure: Sequential Bayesian learning (Bishop 3.7)

$$p(\mathbf{w} \mid (x_1, t_1), (x_2, t_2), \alpha, \beta) \propto p(t_2 \mid x_2, \mathbf{w}, \beta) p(\mathbf{w} \mid (x_1, t_1), \alpha, \beta)$$

Example: Sequential Bayesian Learning

- After 19 data ponts

- Prior

$$p(\mathbf{w} \mid \{(x_n, t_n)\}_{n=1}^{19}, \alpha, \beta)$$

- Likelihood

$$p(t_{20} \mid x_{20}, \mathbf{w}, \beta)$$

- Posterior

$$p(\mathbf{w} \mid \{(x_n, t_n)\}_{n=1}^{20}, \alpha, \beta) \propto p(t_{20} \mid x_{20}, \mathbf{w}, \beta) p(\mathbf{w} \mid \{(x_n, t_n)\}_{n=1}^{19}, \alpha, \beta)$$

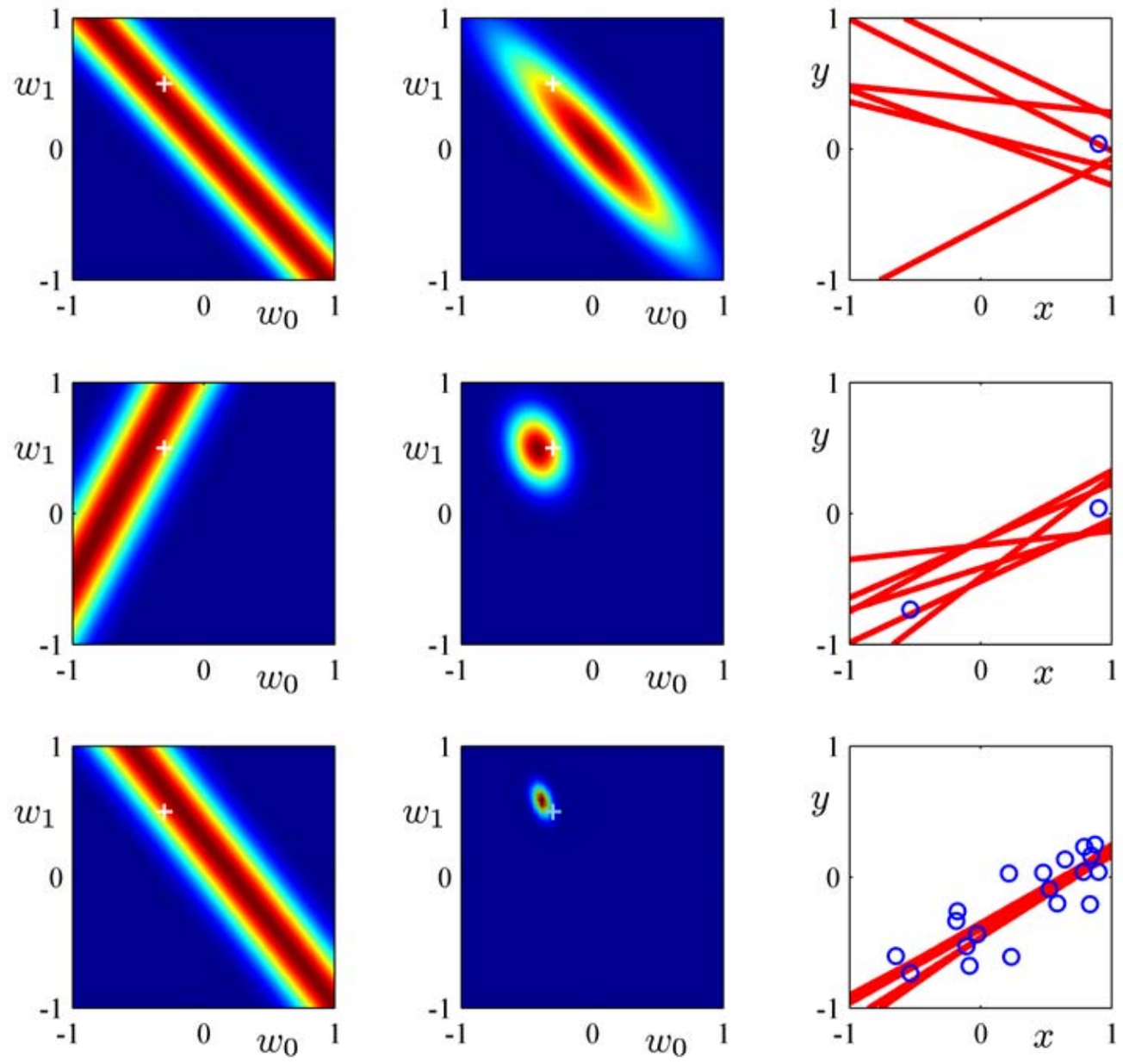


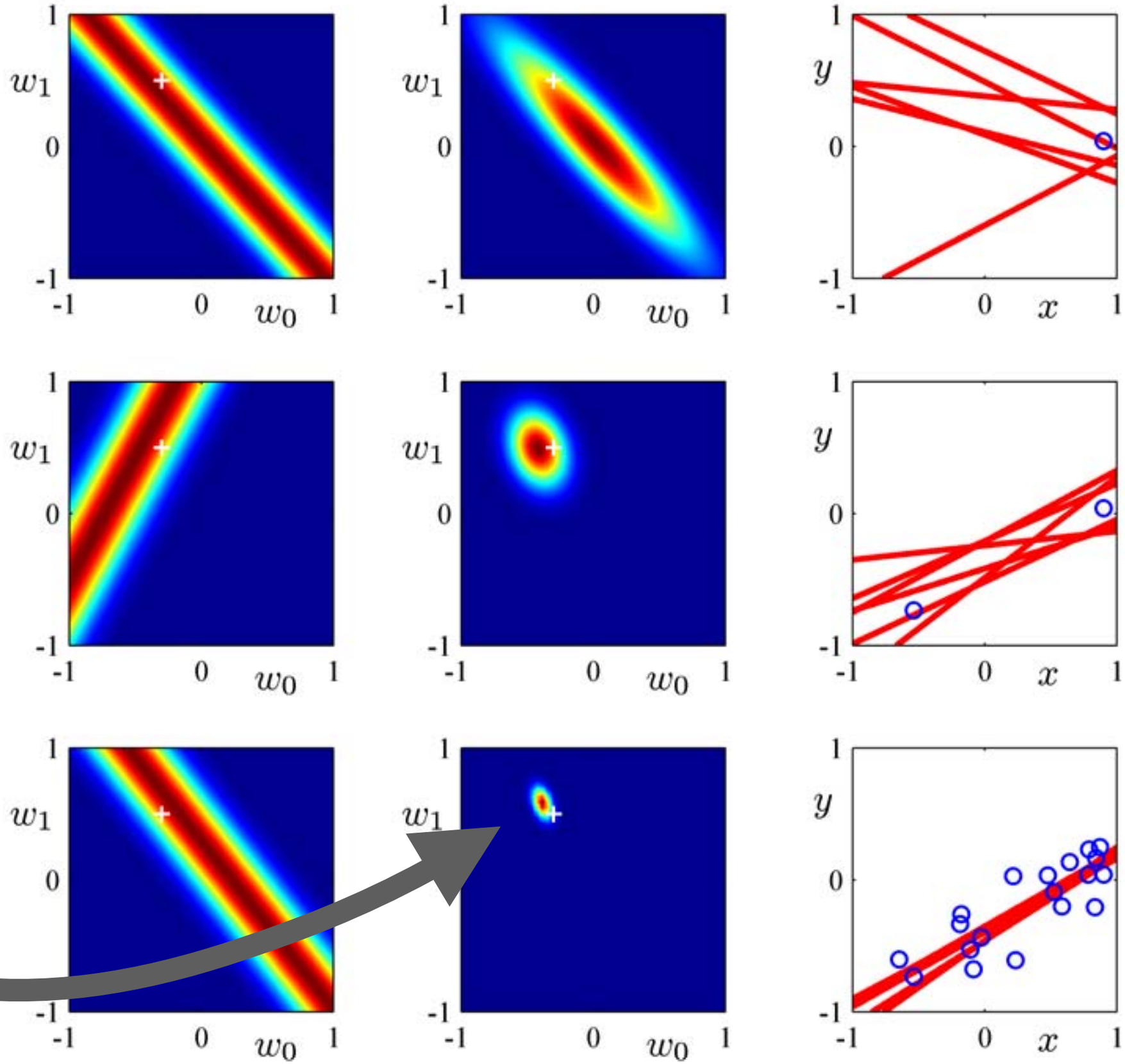
Figure: Sequential Bayesian learning (Bishop 3.7)

Bayesian Linear Regression

- Limiting cases of the posterior
  - $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N)$ with $\mathbf{m}_N = \beta\, \mathbf{S}_N\, \mathbf{\Phi}^T\, \mathbf{t}$

    $\mathbf{S}_N^{-1} = \alpha\, \mathbf{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi}$

- After infinite amount of data ($N \to \infty$):
  - $\lim_{N \to \infty} \mathbf{S}_N = \mathbf{0}$ (zero matrix)
  - $\lim_{N \to \infty} \mathbf{m}_N = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{t} =$ w_{ML}

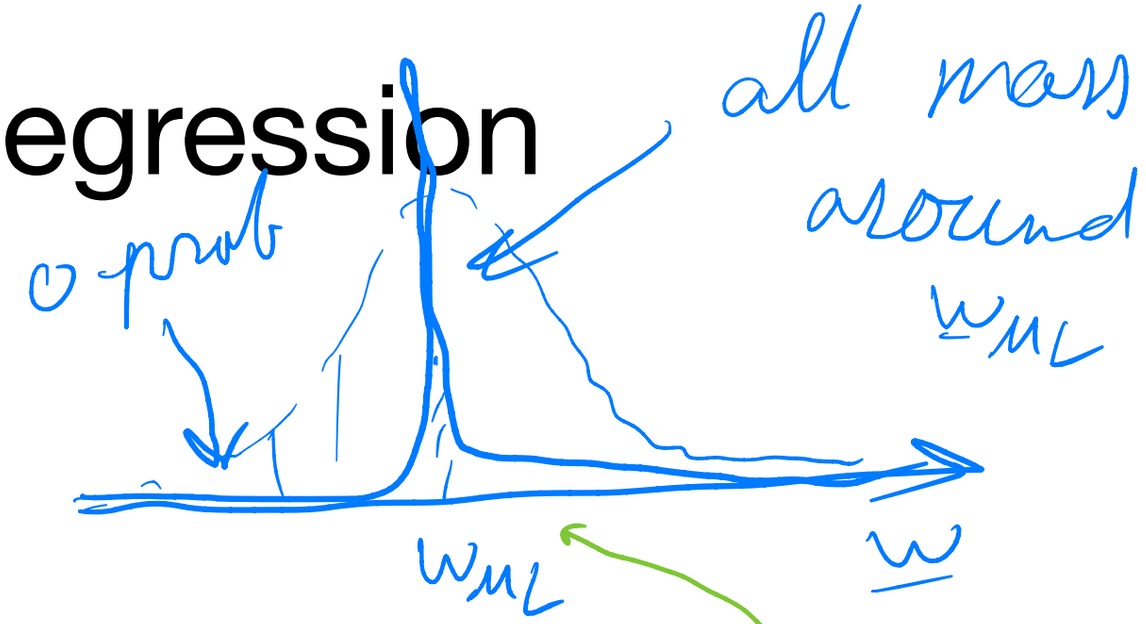


$N \to \infty$

ML solution

Bayesian Linear Regression

- Important limits



- If $N \to \infty$
  - MAP solution convergeces to ML solution!
  - Bayesian predictive distribution

$$p(t' \mid x', D) = \int p(t' \mid x', \mathbf{w}) p(\mathbf{w} \mid D) \mathrm{d}\mathbf{w}$$

converges to?

$$= p(t' \mid x', w_{ML})$$

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 4.5 - Supervised Learning
Bayesian Linear Regression - Predictive Distribution

Erik Bekkers

(Bishop 3.3.2)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Predictive Distribution

- Observed dataset with inputs $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$ and targets $\mathbf{t} = (t_1, \ldots, t_N)^T$
- Gaussian Posterior distribution (from Gaussian prior and Gaussian likelihood)
  - $p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N)$ with $\mathbf{m}_N = \beta\, \mathbf{S}_N\, \mathbf{\Phi}^T\, \mathbf{t}$

    $\mathbf{S}_N^{-1} = \alpha\, \mathbf{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi}$
- Parametrized Gaussian predictive distribution:
  - $p(t' \mid \mathbf{x}', \mathbf{w}, \beta) = \mathcal{N}(t \mid \boldsymbol{\phi}(\mathbf{x}')^T \mathbf{w}, \beta^{-1})$
- Gaussian Bayesian predictive distribution for new input
  - $p(t' \mid x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int \mathcal{N}(t' \mid \boldsymbol{\phi}(\mathbf{x}')^T \mathbf{w}, \beta^{-1})\, \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N)\, d\mathbf{w}$

    $= \mathcal{N}(t' \mid \boldsymbol{\phi}(\mathbf{x}')^T \mathbf{m}_N, \sigma_N^2(\mathbf{x}'))$
- With $\sigma_N^2(\mathbf{x}') = \beta^{-1} + \boldsymbol{\phi}(\mathbf{x}')^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}')$

Bishop Eq. 2.115

mean given by MAP

this integral is analytic and leads to yet another Gaussian

analytic expr. for σ depends on x

Predictive Distribution

active learning
↳ collect more data in uncertain regions

- Datasets:
  - $t = \sin(2\pi x) + \epsilon$
  - $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- Dataset sizes:
  - $N = 1,2,4,25$
- Model:
  - $y(x, \mathbf{w}) = \boldsymbol{\phi}(x)^T \mathbf{w}$
  - $\phi_j(x)$: Gaussian basis functions

Bayesian

- Predictive distribution:
  - $p(t' \mid x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = = \mathcal{N}(t' \mid \boldsymbol{\phi}(\mathbf{x}')^T \mathbf{m}_N, \sigma_N^2(\mathbf{x}'))$
  - $\sigma_N^2(\mathbf{x}') = \beta^{-1} + \boldsymbol{\phi}(\mathbf{x}')^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}'),$ $\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t},$ $\mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$

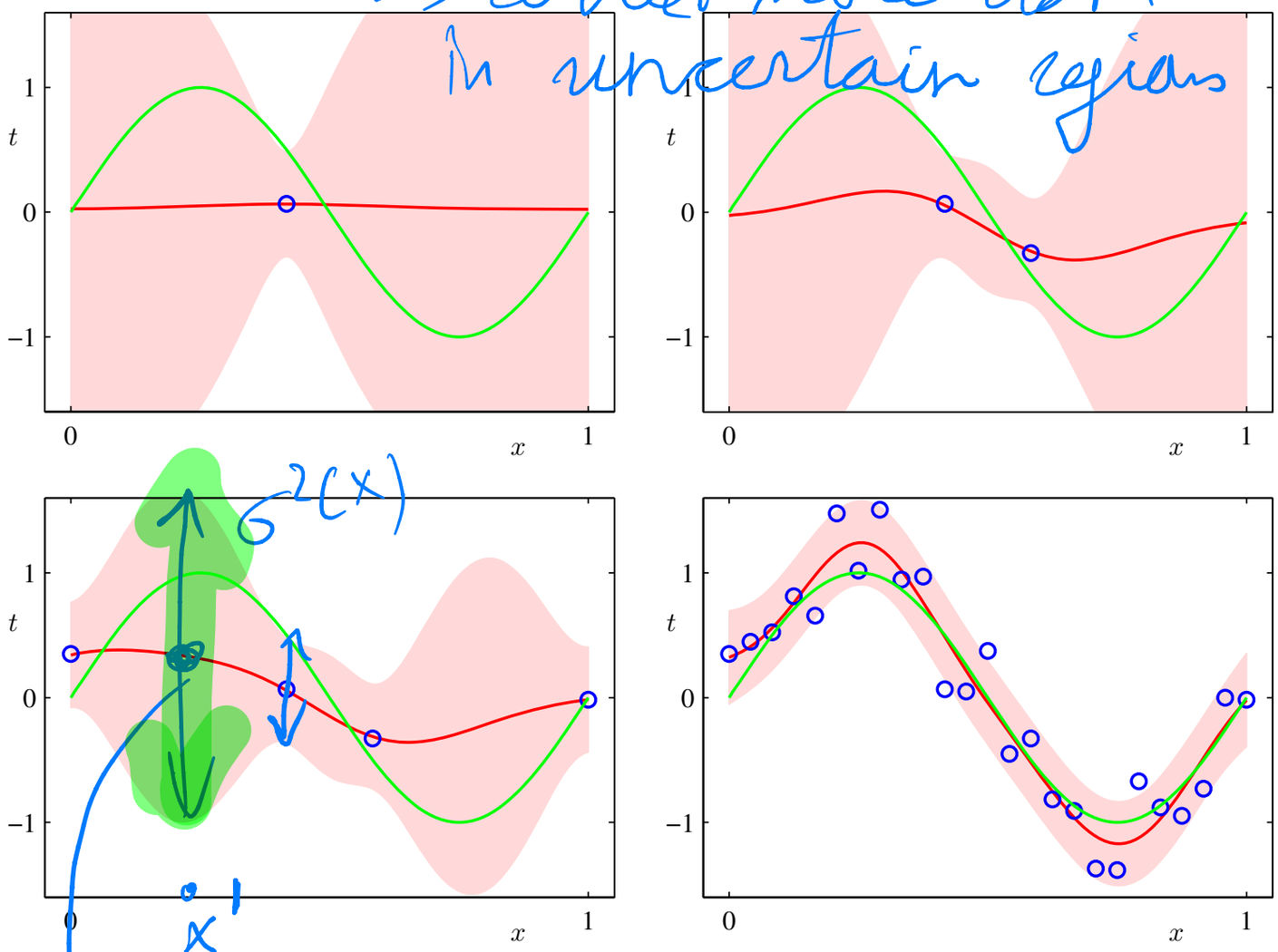
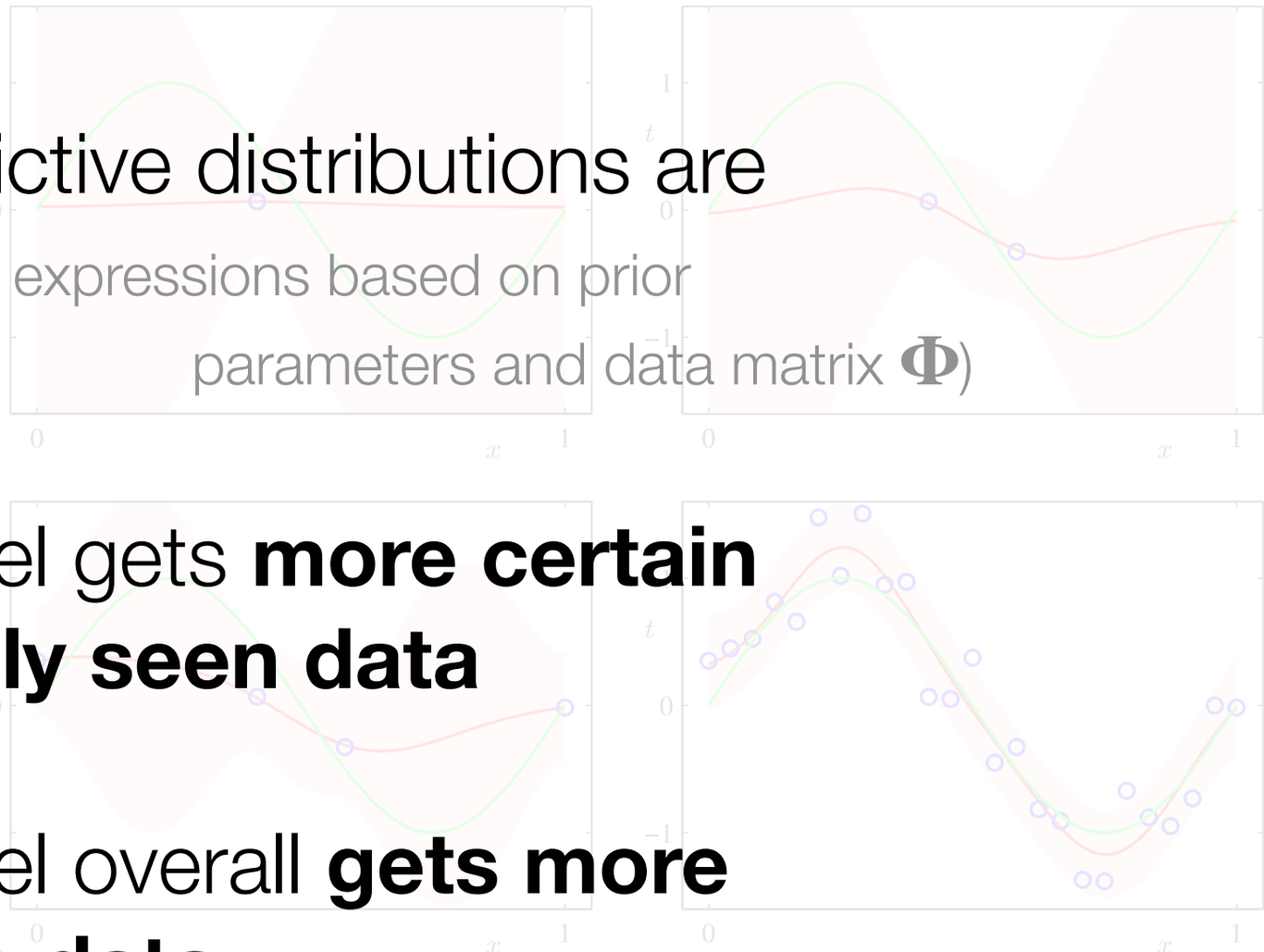


Figure: Predictive distribution (Bishop 3.8)

Predictive Distribution

1. The Bayesian predictive distributions are Gaussian (with analytic expressions based on prior parameters and data matrix Φ)



2. The Bayesian model gets **more certain close to previously seen data**
3. The Bayesian model overall **gets more certain with more data**

Figure: Predictive distribution (Bishop 3.8)

4. The (mean) predictions themselves **get more accurate with more data**

Samples drawn from Bayesian Predictive Distribution

$$p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N),$$

with $\mathbf{m}_N = \beta\, \mathbf{S}_N\, \mathbf{\Phi}^T\, \mathbf{t}$

$$\mathbf{S}_N^{-1} = \alpha\, \mathbf{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi}$$

posterior for w →

Example, linear basis (polynomial degree 1):

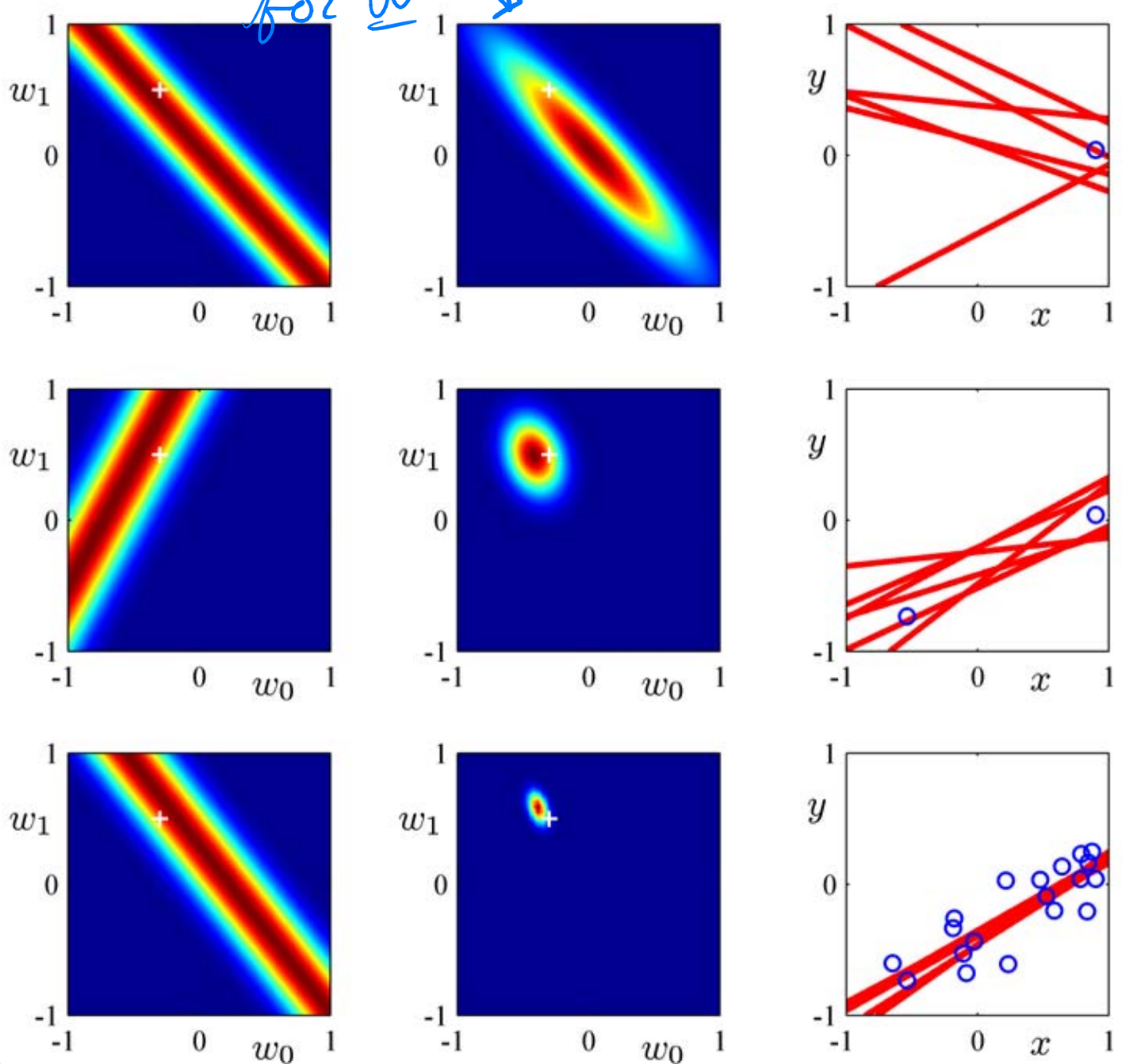


Figure: *Sample functions $y(x, \mathbf{w})$ with $\mathbf{w}$ sampled from posterior distribution (Bishop 3.9)*

Samples drawn from Bayesian Predictive Distribution

$$p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N), \qquad \text{with } \mathbf{m}_N = \beta\, \mathbf{S}_N\, \mathbf{\Phi}^T\, \mathbf{t}$$

$$\mathbf{S}_N^{-1} = \alpha\, \mathbf{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi}$$

$y(x, w)$, $w \sim p(w \mid X, t, \alpha, \beta)$

Example, linear model w basis functions:

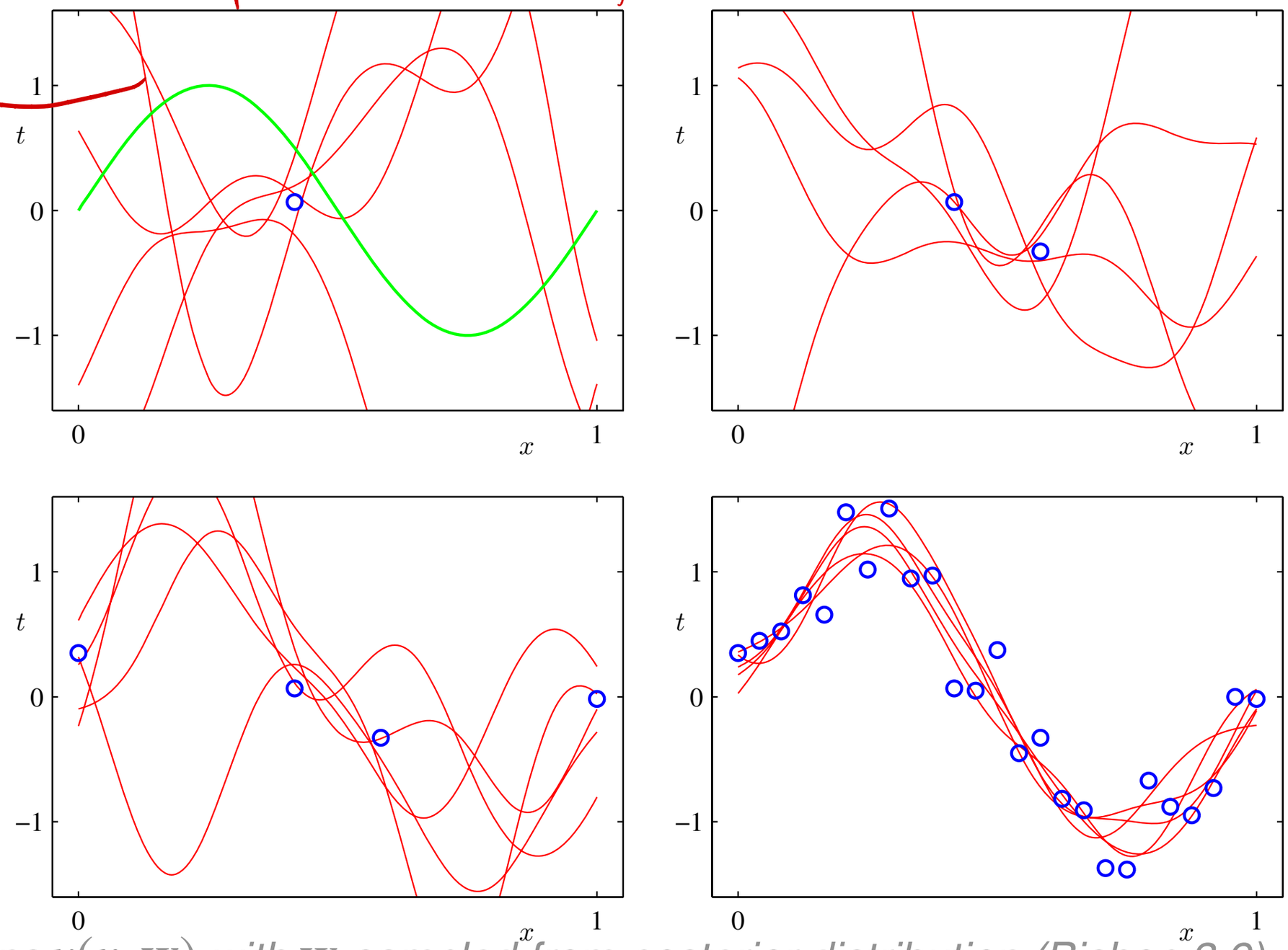


Figure: *Sample functions $y(x, \mathbf{w})$ with $\mathbf{w}$ sampled from posterior distribution (Bishop 3.9)*

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

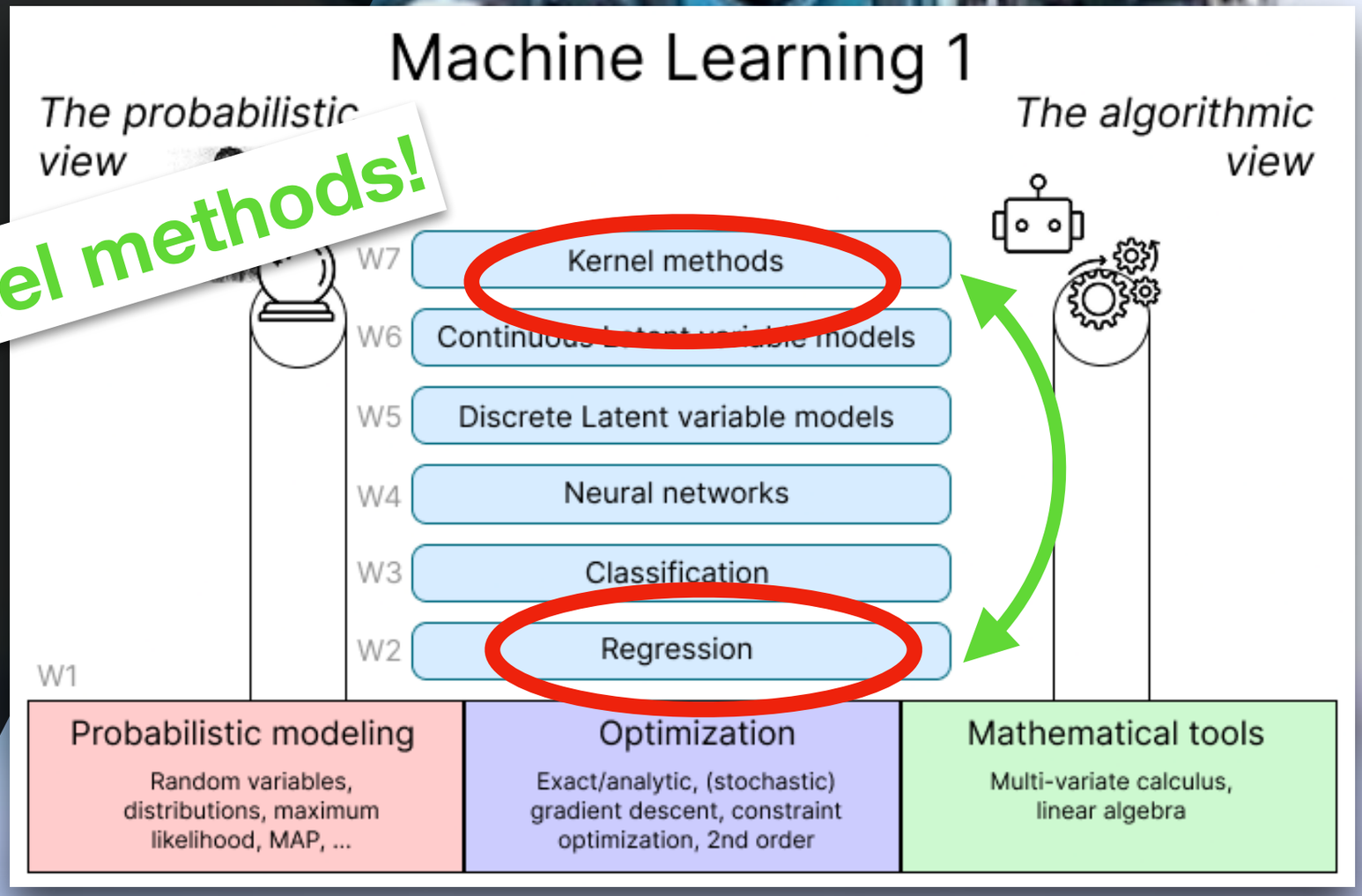
Lecture 5.1 - Supervised Learning
Bayesian Linear Regression - The Equivalent Kernel

Erik Bekkers

(Bishop 3.3.3)

Sneak preview to kernel methods!

for now understand that predictions can be made as a "weighted sum" of the data, without any explicit parameters



Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Equivalent Kernel Formulation

- predictive distribution

$$p(t'|x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int p(t'|x', \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) \mathrm{d}\mathbf{w}$$

$$= \mathcal{N}(t'|\mathbf{m}_N^T \boldsymbol{\phi}(x'), \sigma_N^2(x'))$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \qquad \sigma_N^2(x') = \frac{1}{\beta} + \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x')$$

$$\mathbf{S}_N^{-1} = \alpha \mathbb{1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

Equivalent Kernel Formulation

- predictive distribution

$$p(t'|x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int p(t'|x', \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) \mathrm{d}\mathbf{w}$$

$$= \mathcal{N}(t'|\mathbf{m}_N^T \boldsymbol{\phi}(x'), \sigma_N^2(x'))$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \qquad \sigma_N^2(x') = \frac{1}{\beta} + \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x') \qquad \mathbf{S}_N^{-1} = \alpha \mathbb{1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- predictive mean:

$w = w_{MAP} = m_N$

$$y(x', \mathbf{m}_N) = \boldsymbol{\phi}(x')^T \mathbf{m}_N$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

Equivalent Kernel Formulation

- predictive distribution

$$p(t'|x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int p(t'|x', \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) \mathrm{d}\mathbf{w}$$

$$= \mathcal{N}(t'|\mathbf{m}_N^T \boldsymbol{\phi}(x'), \sigma_N^2(x'))$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \qquad \sigma_N^2(x') = \frac{1}{\beta} + \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x') \qquad \mathbf{S}_N^{-1} = \alpha \mathbb{1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- predictive mean:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

$$y(x', \mathbf{m}_N) = \boldsymbol{\phi}(x')^T \mathbf{m}_N$$

$$= \ldots = \sum_{n=1}^{N} k(x', x_n) t_n$$

A weighted sum of training samples,
no parameters!

kernel provides weight

Equivalent Kernel Formulation

- predictive distribution

$$p(t'|x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int p(t'|x', \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) \mathrm{d}\mathbf{w}$$

$$= \mathcal{N}(t'|\mathbf{m}_N^T \boldsymbol{\phi}(x'), \sigma_N^2(x'))$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \qquad \sigma_N^2(x') = \frac{1}{\beta} + \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x') \qquad \mathbf{S}_N^{-1} = \alpha \mathbb{1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- predictive mean:

$$y(x', \mathbf{m}_N) = \boldsymbol{\phi}(x')^T \mathbf{m}_N = \beta \boldsymbol{\phi}(x') \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t}$$

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

Equivalent Kernel Formulation

- predictive distribution

$$p(t'|x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int p(t'|x', \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) \mathrm{d}\mathbf{w}$$

$$= \mathcal{N}(t'|\mathbf{m}_N^T \boldsymbol{\phi}(x'), \sigma_N^2(x'))$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \qquad \sigma_N^2(x') = \frac{1}{\beta} + \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x') \qquad \mathbf{S}_N^{-1} = \alpha \mathbb{1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- predictive mean:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

$$y(x', \mathbf{m}_N) = \boldsymbol{\phi}(x')^T \mathbf{m}_N = \beta \boldsymbol{\phi}(x') \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_n \sum_{n=1}^{N} \boldsymbol{\Phi}_{n,:}^T t_n$$

Equivalent Kernel Formulation

- predictive distribution

$$p(t'|x', \mathbf{X}, \mathbf{t}, \alpha, \beta) = \int p(t'|x', \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) \mathrm{d}\mathbf{w}$$

$$= \mathcal{N}(t'|\mathbf{m}_N^T \boldsymbol{\phi}(x'), \sigma_N^2(x'))$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \qquad \sigma_N^2(x') = \frac{1}{\beta} + \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x') \qquad \mathbf{S}_N^{-1} = \alpha \mathbb{1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

- predictive mean:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

$$y(x', \mathbf{m}_N) = \boldsymbol{\phi}(x')^T \mathbf{m}_N = \beta \boldsymbol{\phi}(x') \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_n \sum_{n=1}^{N} \boldsymbol{\Phi}_{n,:}^T t_n$$

$$= \sum_{n=1}^{N} \beta \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x_n) t_n$$

kernel

- Equivalent kernel

$$k(x', x) = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x_n)$$

Equivalent kernel for Gaussian Basis Functions

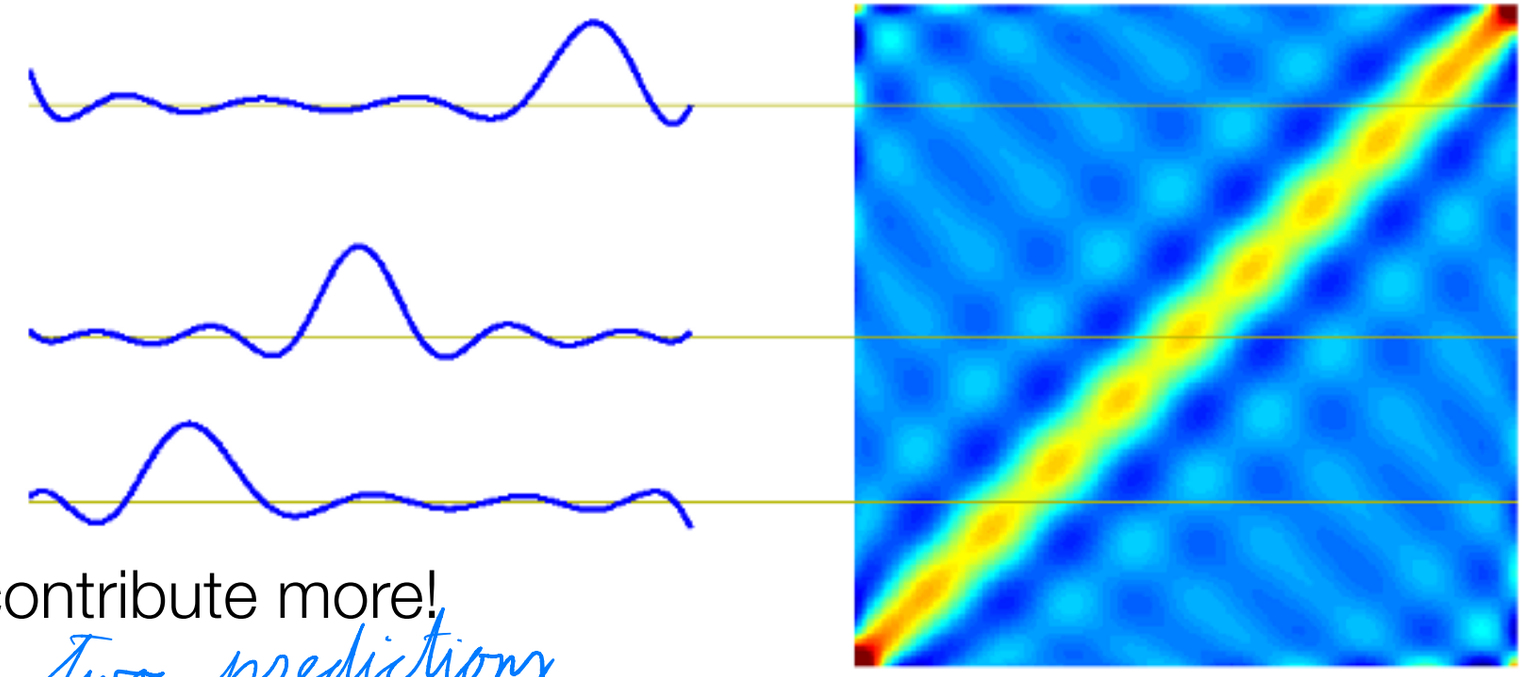
Figure: Equivalent kernel k(x', x) (Bishop 3.10)

- Localized kernel

$$k(x', x) = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x)$$

- predictive mean

$$y(x', \mathbf{m}_N) = \sum_{n=1}^{N} k(x', x_n) t_n$$



- Training points x_n close to x' contribute more!

The kernel describes how two predictions covary

- Covariance of between predictions:

$$\text{cov}[t_1, t_2 \,|\, x_1, x_2] = \text{cov}_{\mathbf{w}}[y(x_1, \mathbf{w}), y(x_2, \mathbf{w})] = \text{cov}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}, \mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

Equivalent kernel for Gaussian Basis Functions

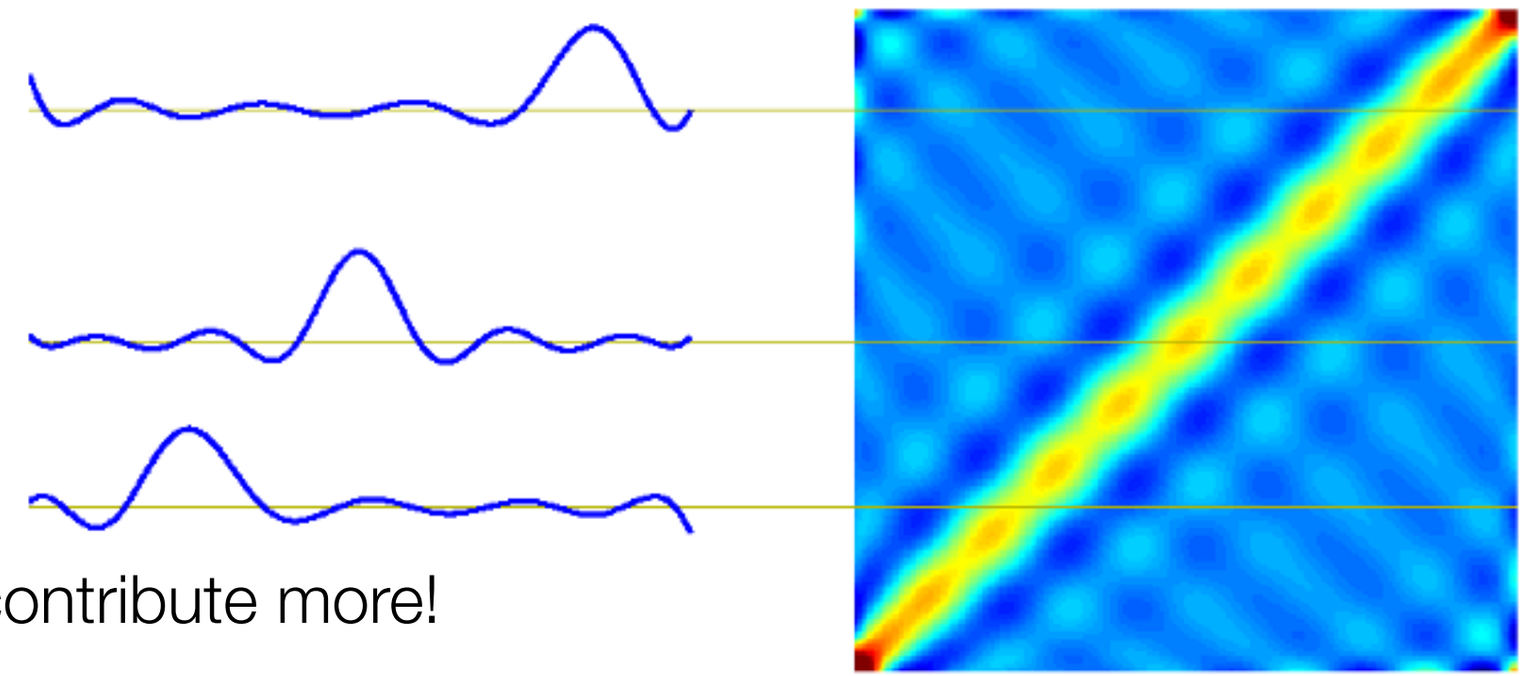
Figure: Equivalent kernel k(x', x) (Bishop 3.10)

- Localized kernel

$$k(x', x) = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x)$$

- predictive mean

$$y(x', \mathbf{m}_N) = \sum_{n=1}^{N} k(x', x_n) t_n$$



- Training points x_n close to x' contribute more!

- Covariance of between predictions:

$$\text{cov}[t_1, t_2 \,|\, x_1, x_2] = \text{cov}_{\mathbf{w}}[y(x_1, \mathbf{w}), y(x_2, \mathbf{w})] = \text{cov}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}, \mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$= \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}\, \mathbf{w}^T \boldsymbol{\phi}(x_2)] - \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}] \mathbb{E}_{\mathbf{w}}[\mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \qquad \mathbb{E}[t'|x', \mathbf{w}] = y(x', \mathbf{w})$$

Equivalent kernel for Gaussian Basis Functions

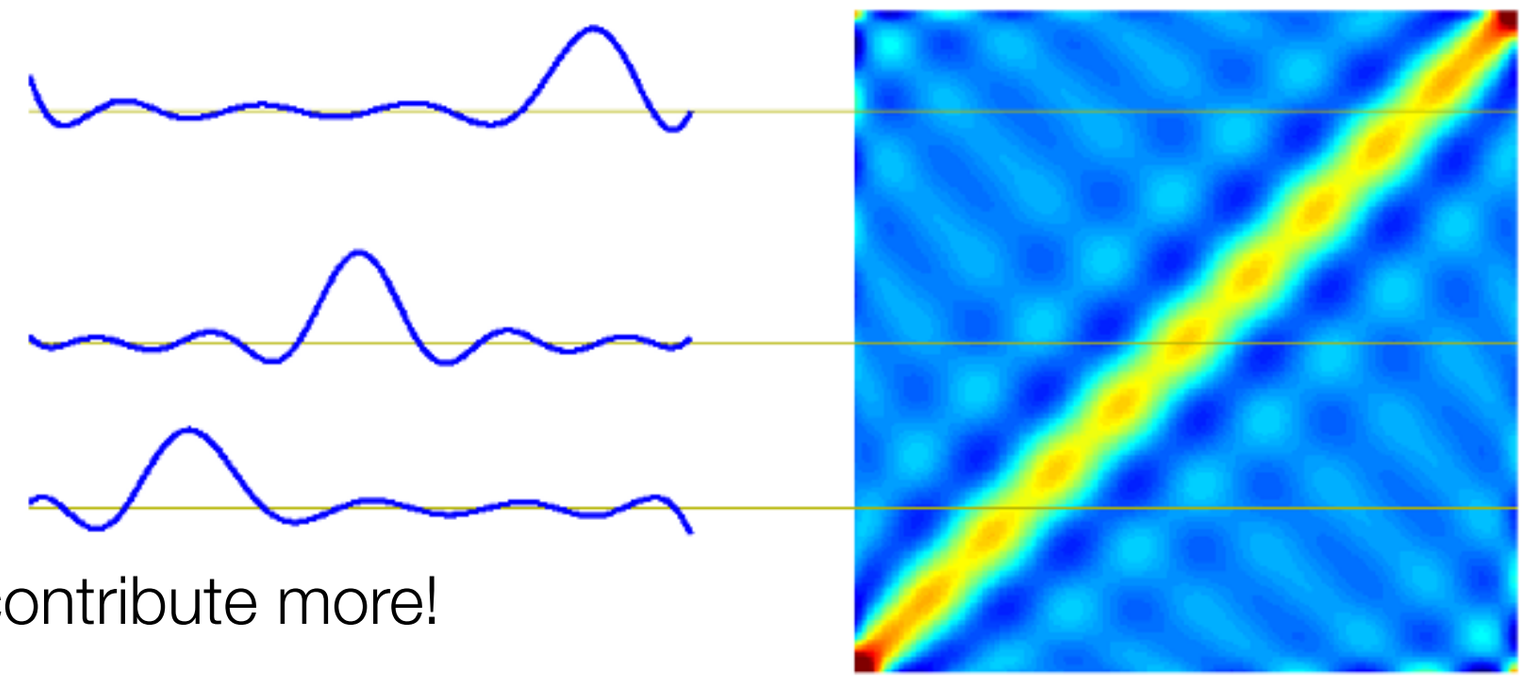
Figure: Equivalent kernel k(x', x) (Bishop 3.10)

- Localized kernel

$$k(x', x) = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x)$$

- predictive mean

$$y(x', \mathbf{m}_N) = \sum_{n=1}^{N} k(x', x_n) t_n$$



- Training points x_n close to x' contribute more!

- Covariance of between predictions:

$$\text{cov}[t_1, t_2 \,|\, x_1, x_2] = \text{cov}_{\mathbf{w}}[y(x_1, \mathbf{w}), y(x_2, \mathbf{w})] = \text{cov}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}, \mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$= \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}\, \mathbf{w}^T \boldsymbol{\phi}(x_2)] - \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}] \mathbb{E}_{\mathbf{w}}[\mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$= \boldsymbol{\phi}(x_1)^T (\mathbb{E}_{\mathbf{w}}[\mathbf{w}\mathbf{w}^T] - \mathbb{E}[\mathbf{w}]\mathbb{E}[\mathbf{w}]^T) \boldsymbol{\phi}(x_2)$$

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \qquad \mathbb{E}[t'|x', \mathbf{w}] = y(x', \mathbf{w})$$

Equivalent kernel for Gaussian Basis Functions

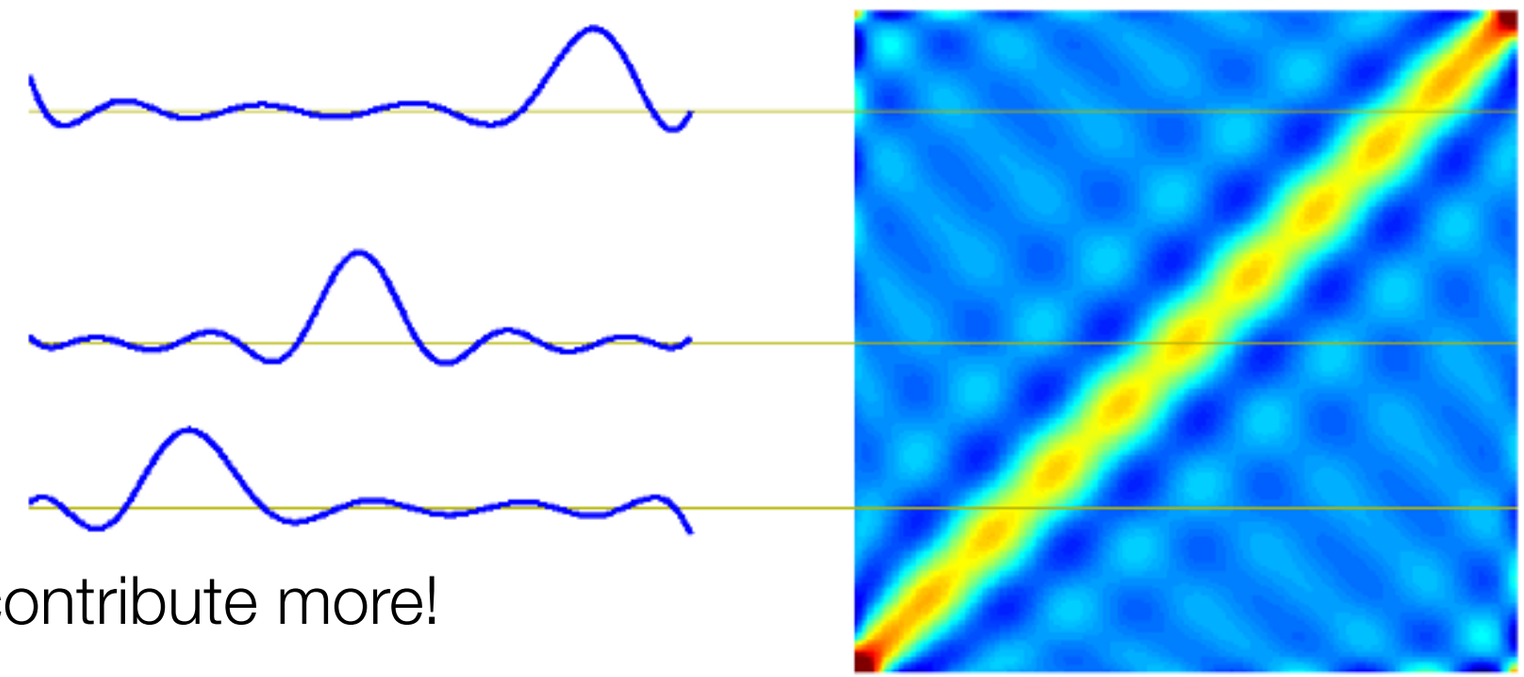
Figure: Equivalent kernel k(x', x) (Bishop 3.10)

- Localized kernel

$$k(x', x) = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x)$$

- predictive mean

$$y(x', \mathbf{m}_N) = \sum_{n=1}^{N} k(x', x_n) t_n$$



- Training points x_n close to x' contribute more!

- Covariance of between predictions:

$$\operatorname{cov}[t_1, t_2 \,|\, x_1, x_2] = \operatorname{cov}_{\mathbf{w}}[y(x_1, \mathbf{w}), y(x_2, \mathbf{w})] = \operatorname{cov}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}, \mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$= \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}\, \mathbf{w}^T \boldsymbol{\phi}(x_2)] - \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}] \mathbb{E}_{\mathbf{w}}[\mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$= \boldsymbol{\phi}(x_1)^T (\mathbb{E}_{\mathbf{w}}[\mathbf{w}\mathbf{w}^T] - \mathbb{E}[\mathbf{w}]\mathbb{E}[\mathbf{w}]^T) \boldsymbol{\phi}(x_2) = \boldsymbol{\phi}(x_1)^T \operatorname{cov}[\mathbf{w}, \mathbf{w}] \boldsymbol{\phi}(x_2)$$

Equivalent kernel for Gaussian Basis Functions

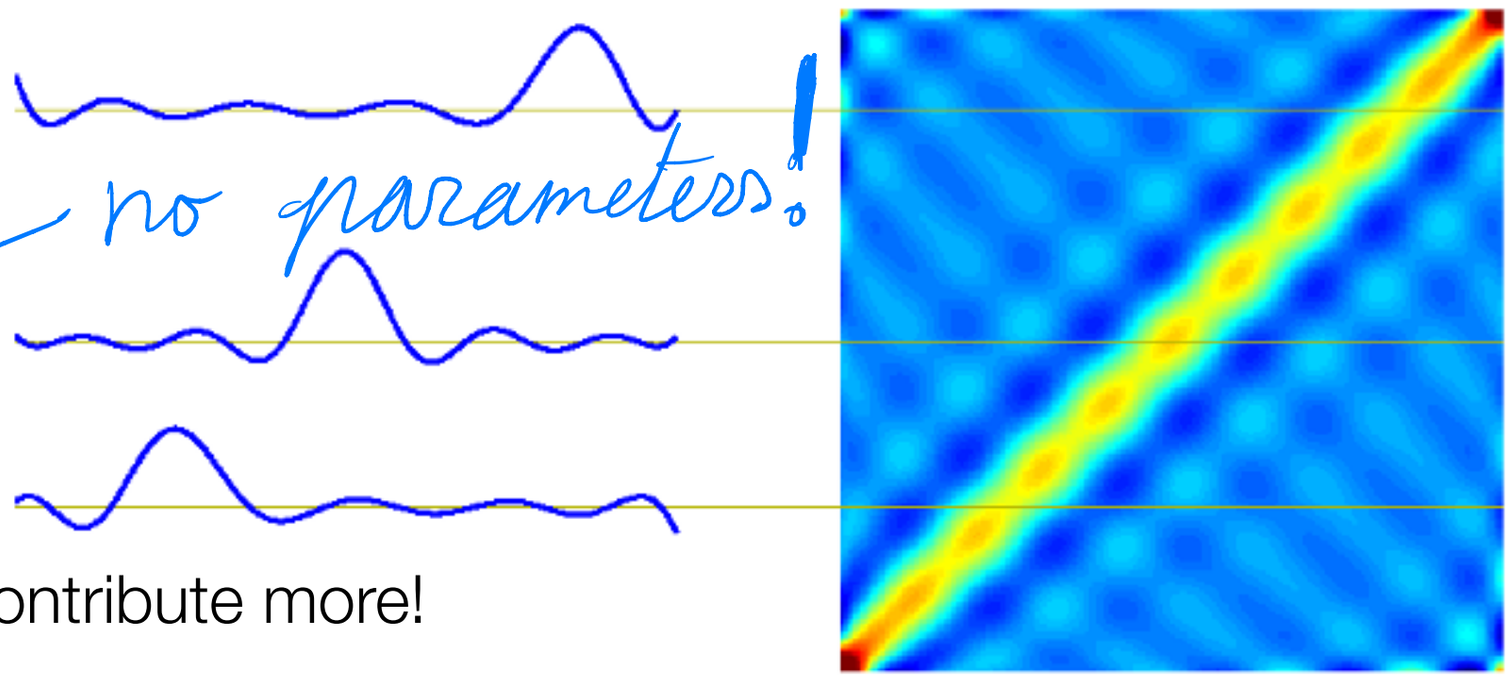
Figure: Equivalent kernel k(x', x) (Bishop 3.10)

- Localized kernel

$$k(x', x) = \beta \boldsymbol{\phi}(x')^T \mathbf{S}_N \boldsymbol{\phi}(x)$$

- predictive mean

$$y(x', \mathbf{m}_N) = \sum_{n=1}^{N} k(x', x_n) t_n$$



- Training points x_n close to x' contribute more!

- Covariance of between predictions:

$$\operatorname{cov}[t_1, t_2 \,|\, x_1, x_2] = \operatorname{cov}_{\mathbf{w}}[y(x_1, \mathbf{w}), y(x_2, \mathbf{w})] = \operatorname{cov}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}, \mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$= \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}\, \mathbf{w}^T \boldsymbol{\phi}(x_2)] - \mathbb{E}_{\mathbf{w}}[\boldsymbol{\phi}(x_1)^T \mathbf{w}] \mathbb{E}_{\mathbf{w}}[\mathbf{w}^T \boldsymbol{\phi}(x_2)]$$

$$= \boldsymbol{\phi}(x_1)^T (\mathbb{E}_{\mathbf{w}}[\mathbf{w}\mathbf{w}^T] - \mathbb{E}[\mathbf{w}]\mathbb{E}[\mathbf{w}]^T) \boldsymbol{\phi}(x_2) = \boldsymbol{\phi}(x_1)^T \operatorname{cov}[\mathbf{w}, \mathbf{w}] \boldsymbol{\phi}(x_2) = \boldsymbol{\phi}(x_1)^T \mathbf{S}_N \boldsymbol{\phi}(x_2)$$

e.g. covariance matrix of posterior $p(w|D)$ that was derived before in the Gaussian case

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \qquad \mathbb{E}[t'|x', \mathbf{w}] = y(x', \mathbf{w})$$

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 5.2 - Supervised Learning
Bayesian Linear Regression - Bayesian Model Comparison

Erik Bekkers

(Bishop 3.4)

Image credit: Kirillm | Getty Images

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 5.3 - Supervised Learning
Bayesian Linear Regression - Approximating the Model Evidence

Erik Bekkers

(Bishop 3.5.0)

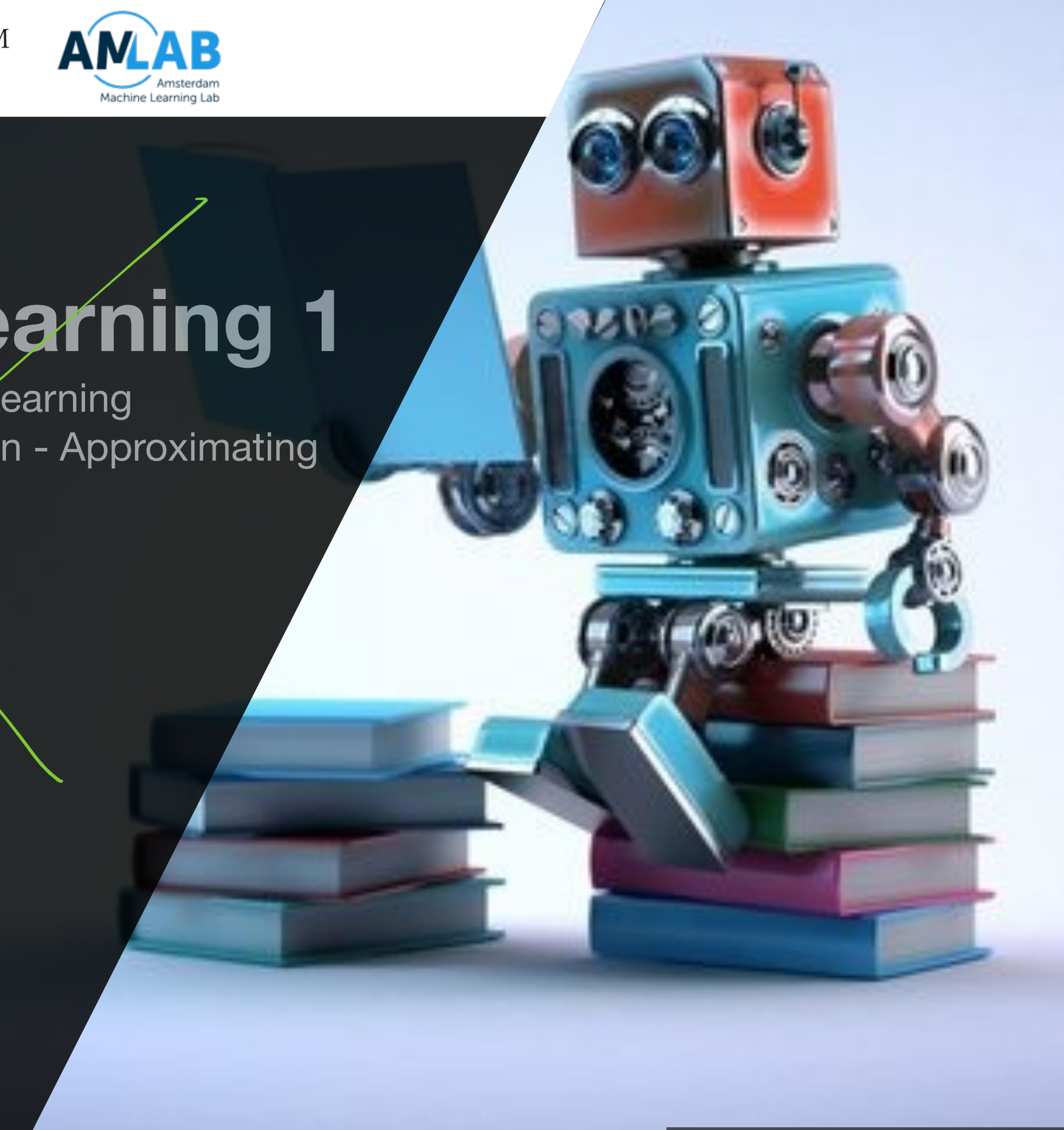


Image credit: Kirillm | Getty Images

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 5.4 - Supervised Learning
Classification - Decision Regions

Erik Bekkers

(Bishop 1.5, 4.1)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Classification through decision regions

- Input: $\mathbf{x} = (x_1, \ldots, x_D)^T$ $\in \mathbb{R}^D$

$\underline{t} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$

- Target:
  - 2-class targets: $t \in \{C_1, C_2\} = \{\text{"cat"}, \text{"dog"}\}$
  - Multi-class targets $t \in \{C_1, C_2, \ldots, C_K\}$ $\underline{t} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \in \mathbb{R}^K$

Strategy: One-hot encoding

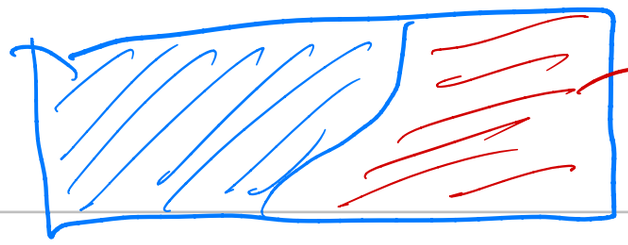
a vector with only a 1 at the class-index but all other elements zero

$\underline{t} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$

- Divide input space $\mathbb{R}^D$ into K *decision regions* R_k
- Assign each decision region to a class C_k
- Boundaries of decision regions are called *decision boundaries/surfaces*.

if $\underline{x}$ falls in R_2 ($\underline{x} \in R_2$) then classify as C_2 ($\underline{x} \mapsto C_2$)

"maps to"

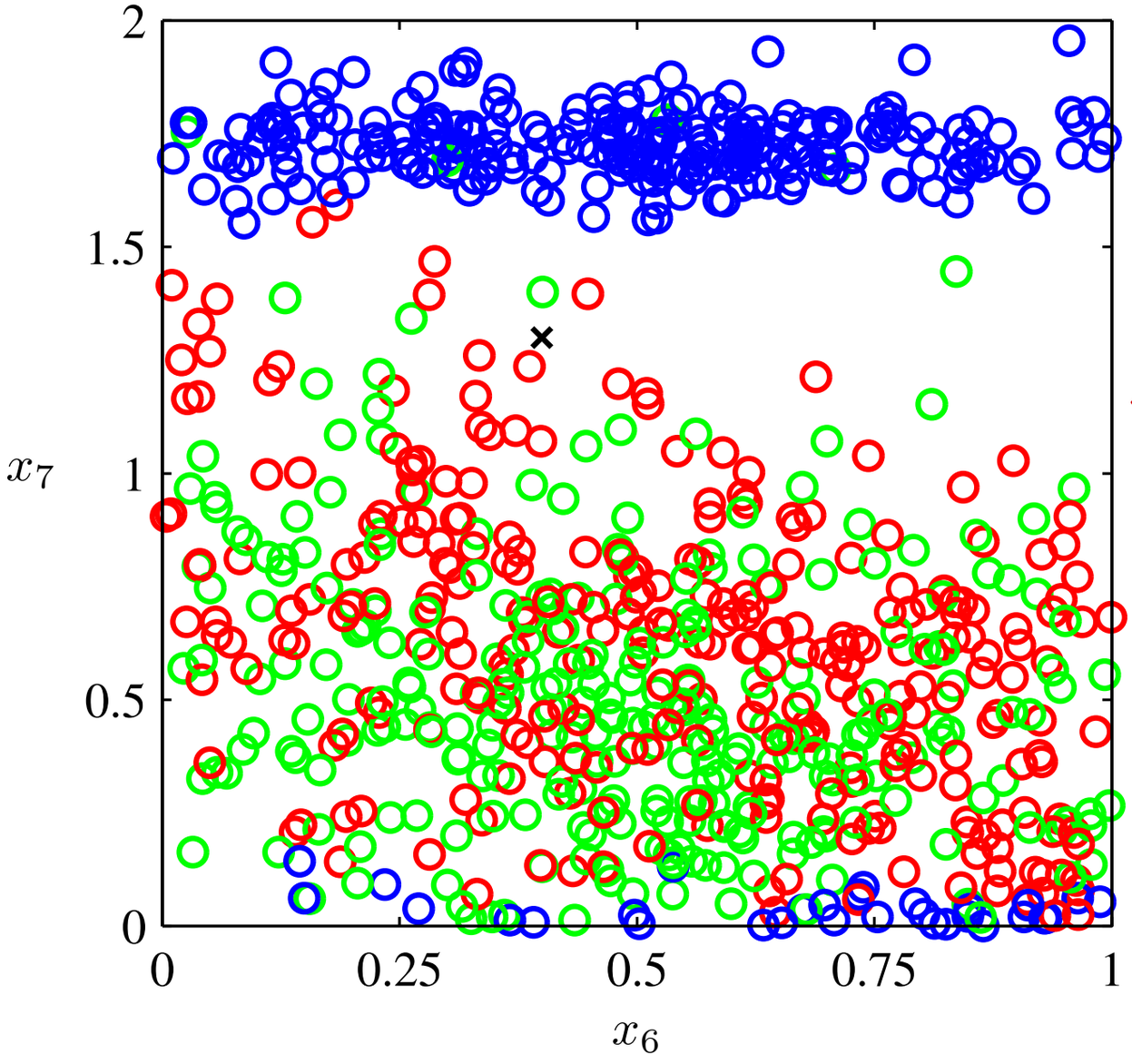


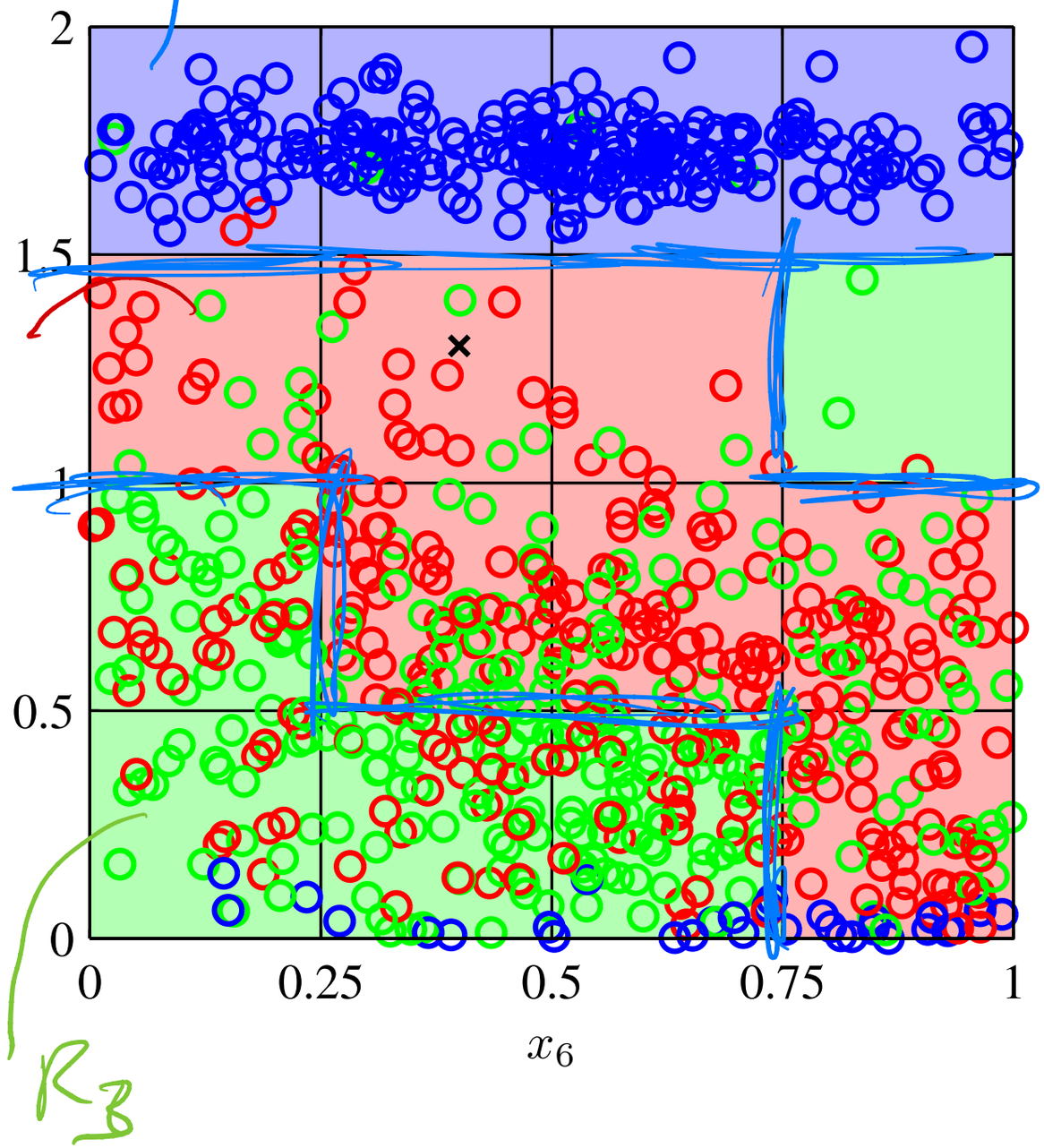
$\underline{x} \in R_1 \mapsto C_1$

$$\underline{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_K \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \vdots \end{pmatrix} \in \{0,1\}^K$$

is a one-hot encoding for class C_k if $t_i = 1$ if $i = k$ and zero otherwise. In the above $\underline{t}$ is a one-hot encoding for class C_2 since $t_2 = 1$ and all other entries are zero. Such a one-hot encoding can be thought of as a categorical distribution assigning probability 1 to the corresponding class and zero to others $p(\text{class} = C_k) = 1$.

Classification through Decision Regions





Figures: 3 class problem with decision boundaries. (Bishop 1.19 & 1.20)

Linear Classification

- Linear Classification: consider only *linear* decision boundaries
- For D-dimensional input space: $\mathbf{x} \in \mathbb{R}^D$
  decision surface is a $D-1$ dimensional hyperplane
- Datasets whose classes can be separated *exactly* by linear decision surfaces are called **linearly separable**

D=2

D=1

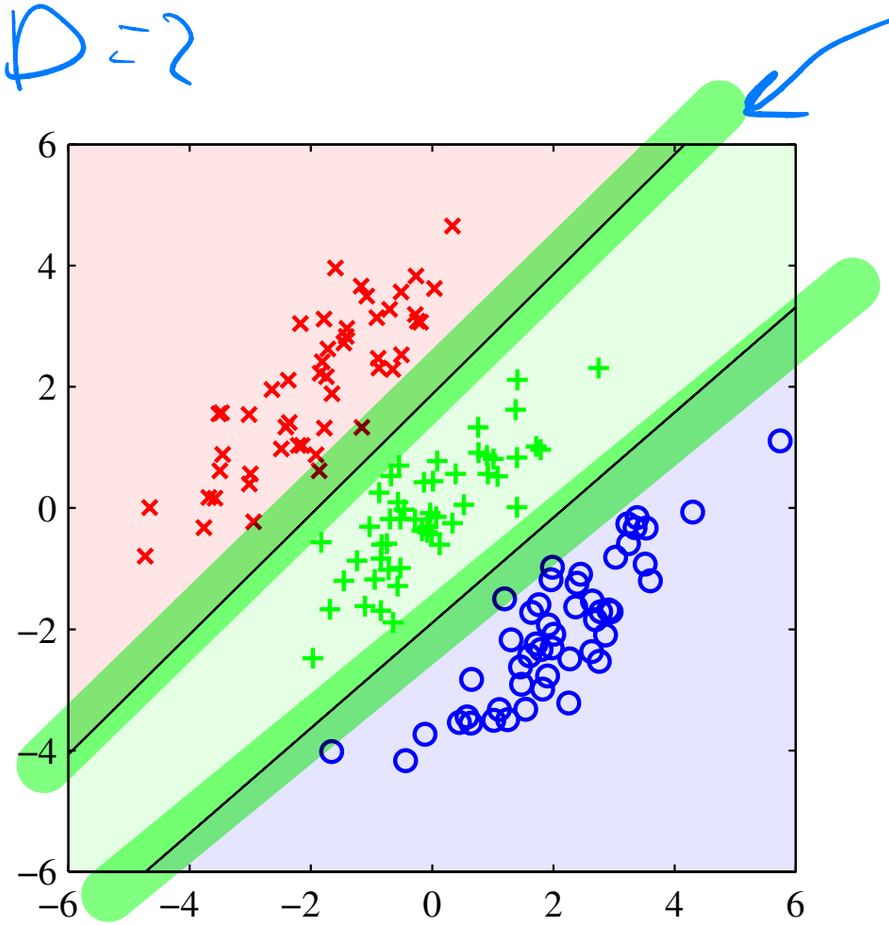


Figure: Linearly separable dataset (Bishop 4.5)

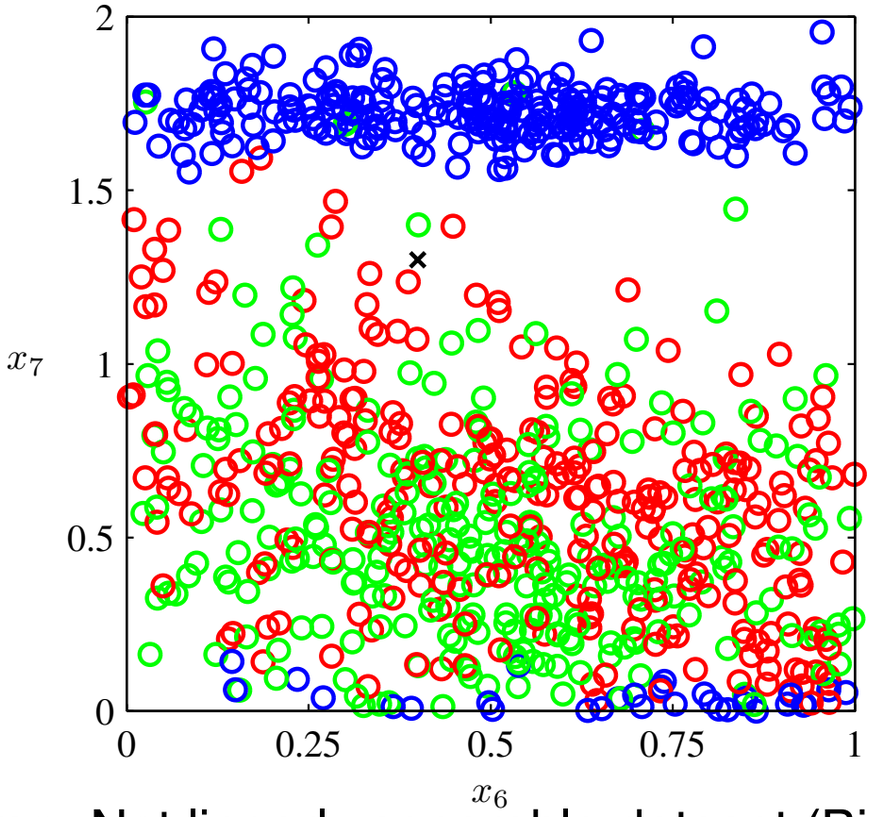


Figure: Not linearly separable dataset (Bishop 1.19)

Multiple Classes ($K > 2$)

- $K = 2$ classes:
  - 1 classifier determines
- Multiple classes: $K > 2$
  - $K - 1$ classifiers:
- One-versus-the-rest

using multiple binary classifiers

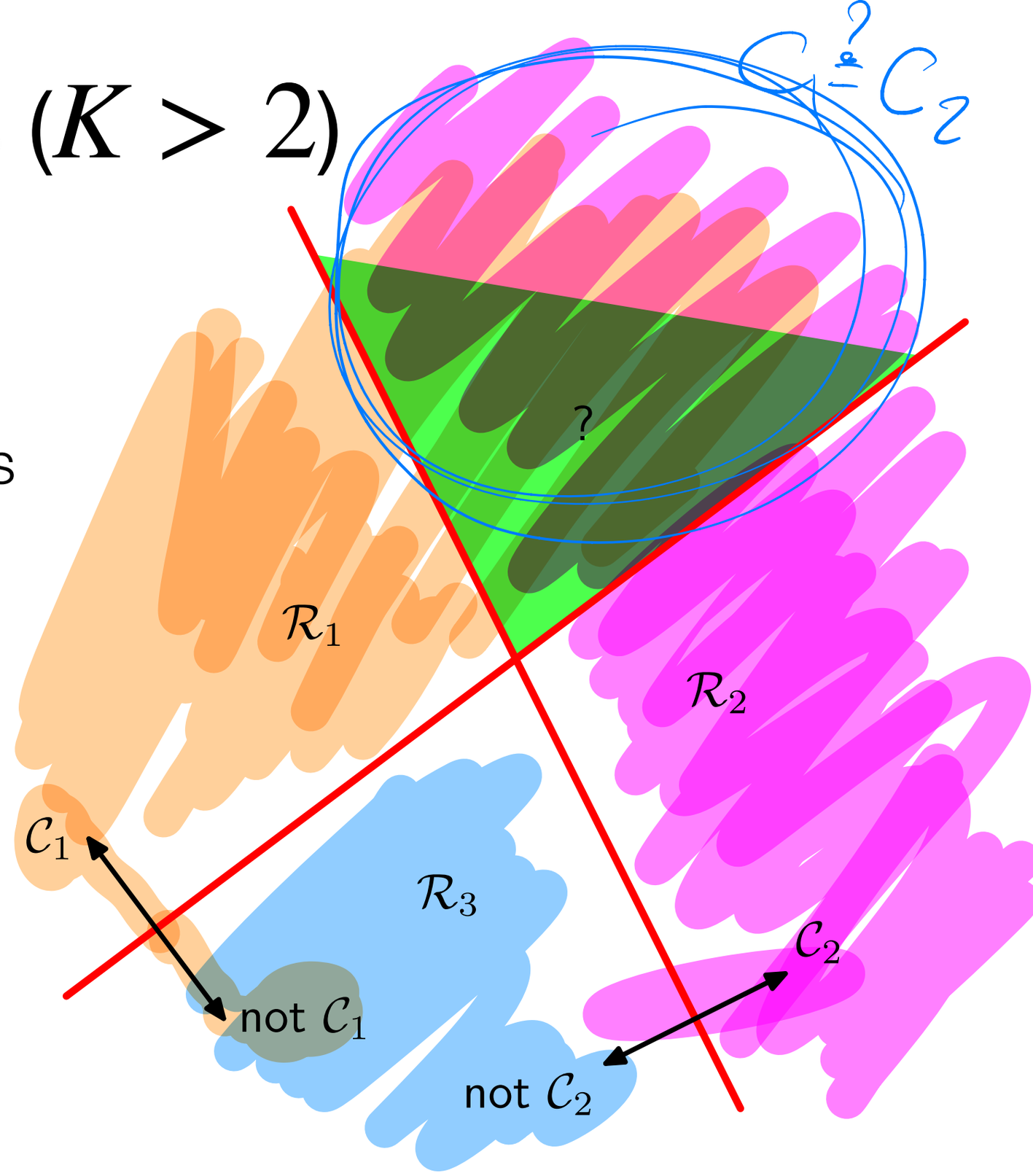
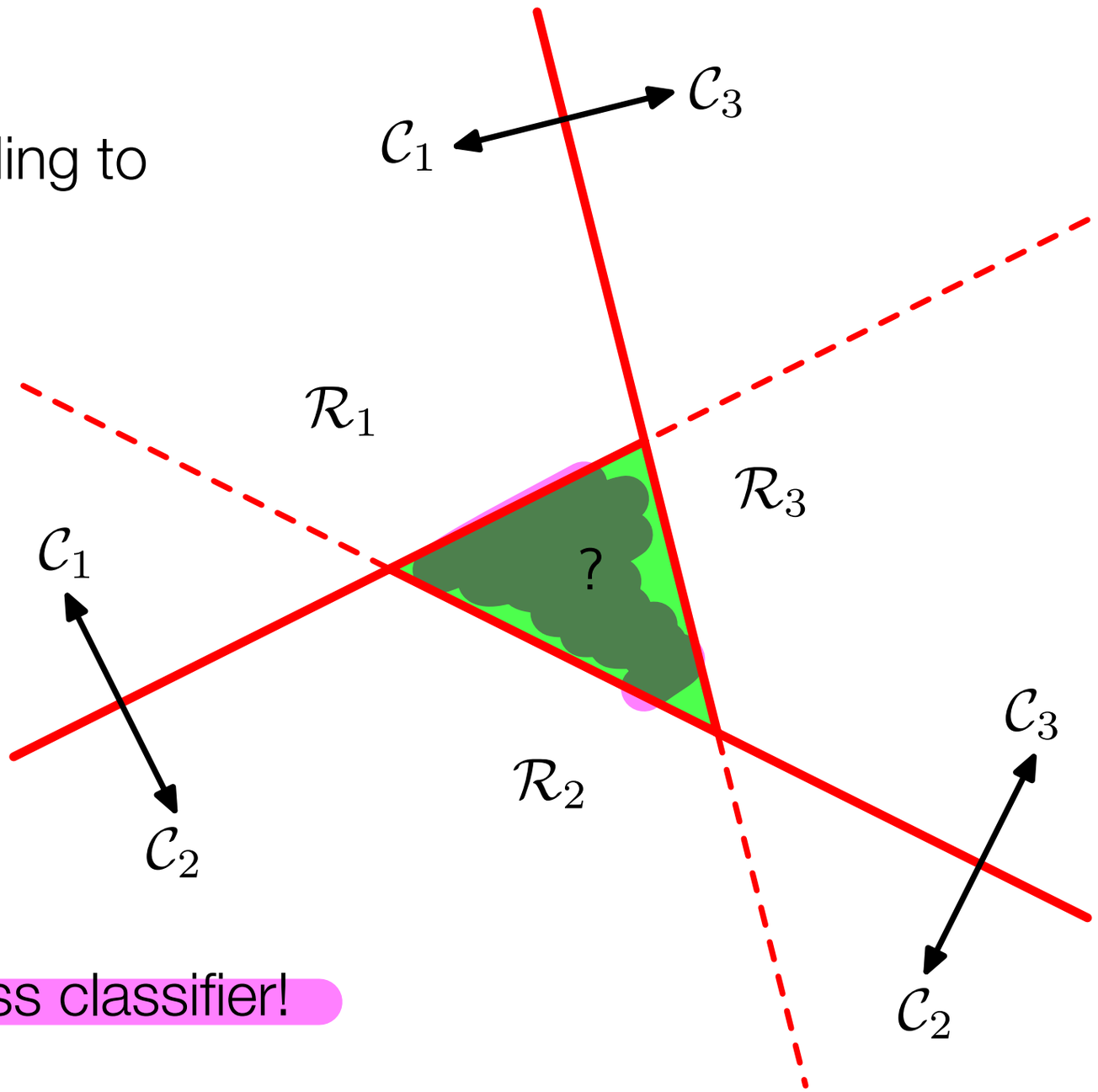


Figure: one-versus-the-rest classifiers (Bishop 4.2)

Multiple Classes ($K > 2$)

- $K(K-1)/2$ classifiers:
- Points are classified according to majority vote of classifiers
- one-versus-one



- **Solution**: Make one K-class classifier!
  (See later)

Figure: one-versus-one classifiers (Bishop 4.2)

University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 5.5 - Supervised Learning
Classification - Decision Theory

Erik Bekkers

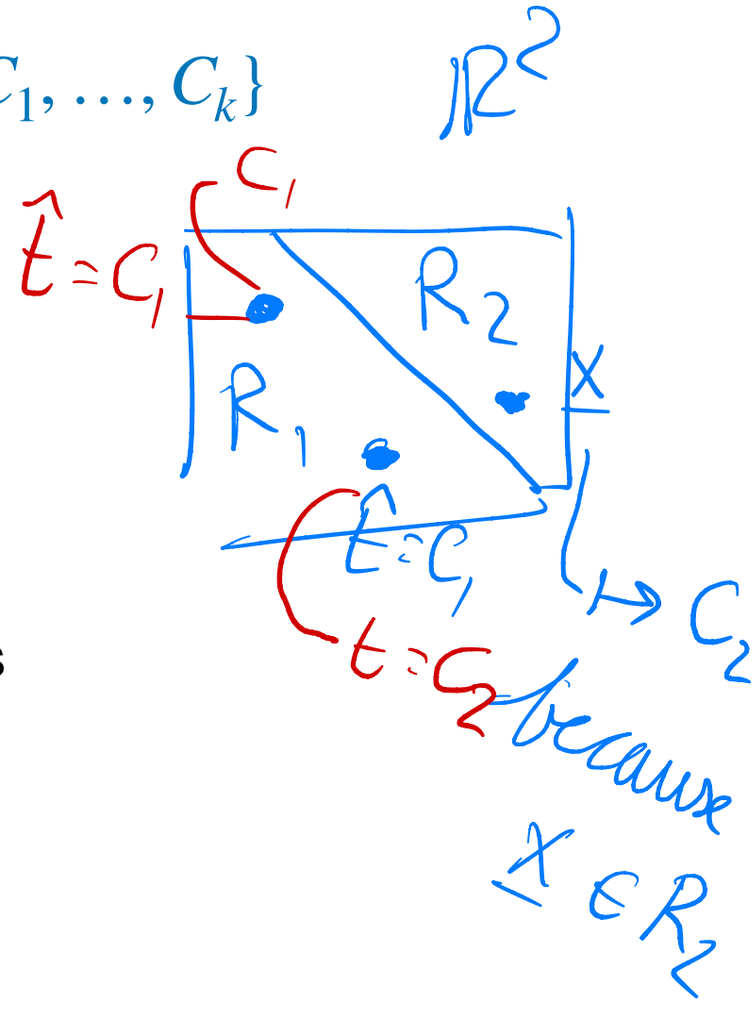
(Bishop 1.5)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Decision theory

- Dataset: Input vectors $\mathbf{x} \in \mathbb{R}^D$, ground truth targets $t \in \{C_1, \ldots, C_k\}$
- Divide input space $\mathbb{R}^D$ into K decision regions R_k
- Every observed datapoint: { ground truth / prediction
- Confusion matrix: ground truth classes vs. predicted classes



matrix of counts or frequencies for each type of prediction

ground truth

pred C_1 pred C_2 pred C_K

	$\mathcal{R}_1$	$\mathcal{R}_2$	$\ldots$	$\mathcal{R}_K$
C_1	6	1	$\ldots$	0
C_2	5	3	$\ldots$	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
C_K	2	0	$\ldots$	8

- Diagonal elements: correctly classified
- Off-diagonal elements: misclassified

Decision theory: Misclassification Rate

- Classification goal: Minimize the misclassification rate
- Assume observations are drawn from joint distribution $(\mathbf{x}, t) \sim p(\mathbf{x}, t)$
- Probability of a misclassification:

Note: For every one of K classes, we can make $K-1$ mistakes (so $K(K-1)$ mistakes)

(Only when predicted class C_i coincides with true class C_k)

$$p(\text{mistake}) = \sum_{i=1}^{K} \sum_{k \neq i} p(\mathbf{x} \in R_i, t = C_k)$$

assign to class i ← should belong to class k

$$= 1 - p(\text{correct}) = 1 - \sum_{k=1}^{K} p(\mathbf{x} \in R_k, C_k)$$

Minimizing misclassification rate

- Assign $\mathbf{x}$ to class C_k if $p(\mathbf{x}, t = C_k) > p(\mathbf{x}, t = C_j)$
- Note: $p(\mathbf{x}, C_k) = p(C_k \mid \mathbf{x}) p(\mathbf{x})$

→ instead maximize:

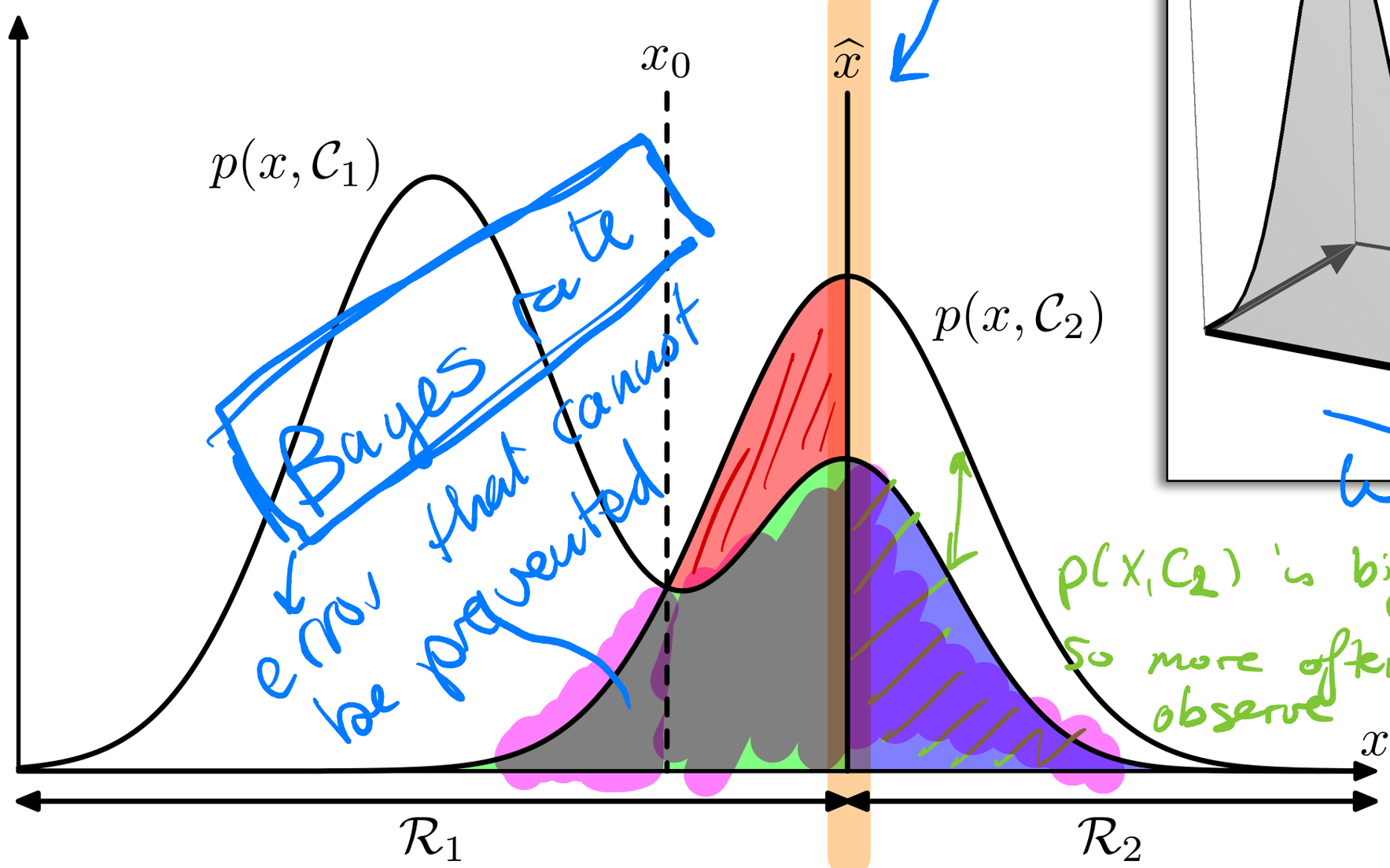
$$p(C_k \mid \mathbf{x}) > p(C_j \mid \mathbf{x})$$

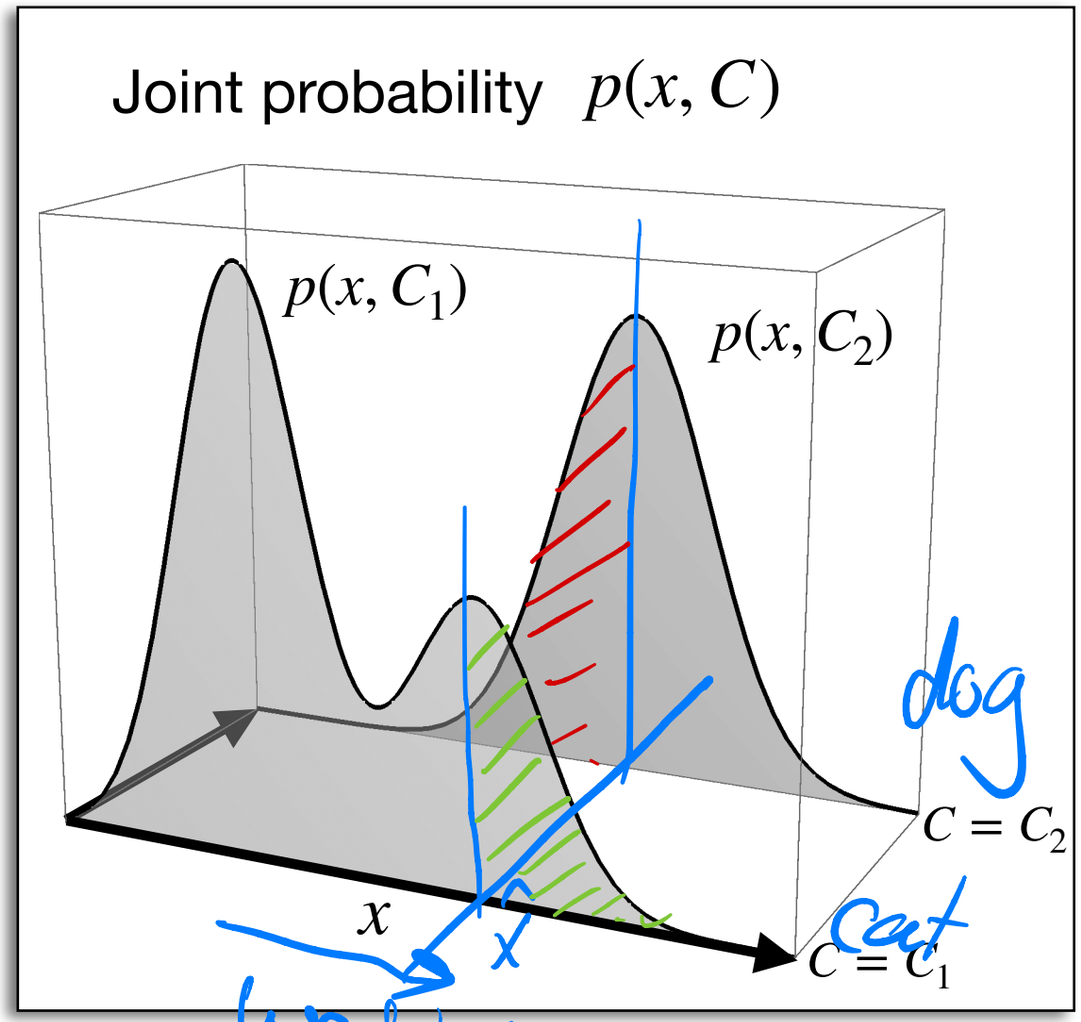
Decision theory: Misclassification Rate

$\hat{x}$: Decision boundary

x_0 : Optimal Decision boundary

some chosen threshold





$p(x, C_2)$ is bigger than $p(x, C_1)$ so more often than not we will observe a dog in this region.

Figure: joint probability distributions and decision boundary (Bishop 1.24)

Minimizing the Misclassification Rate: Problems

- Not all errors have the same impact!

Example: Medical diagnosis of cancer

- Error 1: Label a healthy person as having cancer.
- Error 2: Label a sick person as healthy. Lack of treatment!

important

severity of error

- If cancer only occurs in 1% of all patients, a classifier which labels everyone as healthy has a misclassification rate of 1%!

class imbalance

should also be corrected for

Expected Loss

- Possible solution: use different weights for different error types

$$L = \begin{pmatrix} & \text{label cancer} & \text{label healthy} & \\ & 0 & 1000 & \text{true cancer} \\ & 1 & 0 & \text{true healthy} \end{pmatrix}$$

- Expected loss:

$$\mathbb{E}_{(\mathbf{x},t)\sim p(\mathbf{x},t)}[L] = \sum_{k,j} L_{kj}\; p(x \in R_j, t = C_k)$$

- Objective: Minimize the expected loss:

$$\text{Assign } \mathbf{x} \text{ to } C_k \text{ if } \sum_{j=1}^{K} L_{jk}\; p(\mathbf{x}, C_j) \text{ is minimal}$$

Classification Strategies

- Discriminant functions

Direct mapping of input to target:

$$\hat{t} = y(\underline{x}, \underline{w})$$

- Probabilistic discriminative models

Posterior class probabilities:

$$p(C_k \mid \underline{x})$$

- Probabilistic generative models

Class-conditional densities:

$$p(\underline{x} \mid C_k)$$

Prior class probabilities:

$$p(C_k)$$

$$\Big\} \; p(\underline{x}, C_k)$$

$$p(C_k \mid \underline{x})$$

University of Amsterdam
Informatics Institute

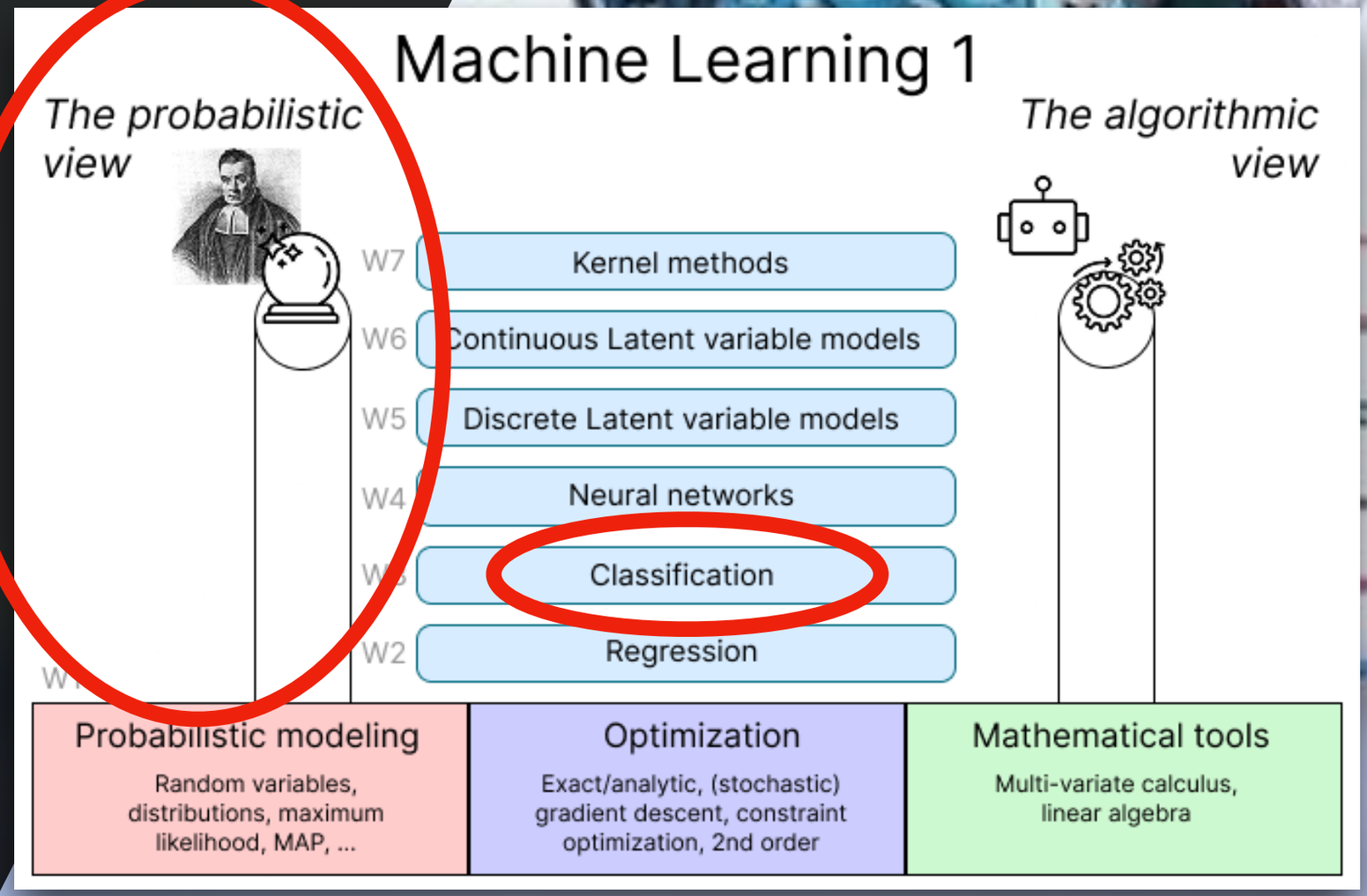
AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 5.6 - Supervised Learning
Classification - Probabilistic Generative Models

Erik Bekkers

(Bishop 1.5)

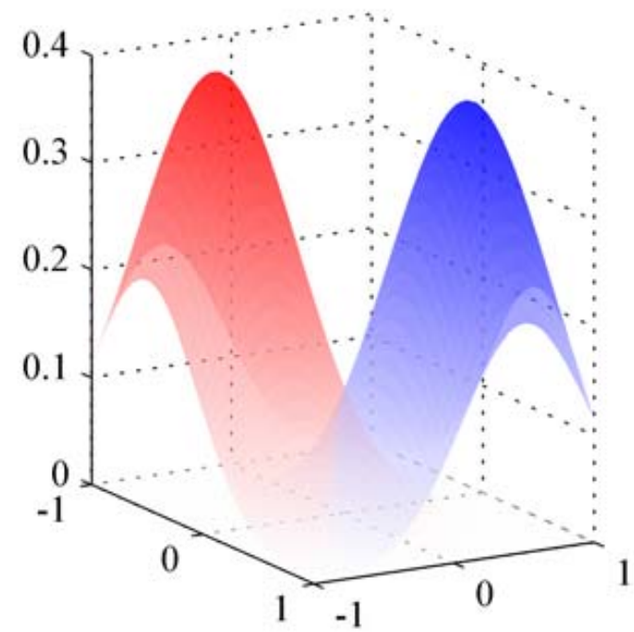


Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Probabilistic Generative Models: general K

- Model:
  - Class-conditional densities: $p(\mathbf{x} \mid C_k)$
  - Prior class probabilities: $p(C_k)$
- This gives access to joint and posterior:
  - $p(\mathbf{x}, C_k)$ and $p(C_k \mid \mathbf{x})$



- Why are we interested in joint or posterior?
  - Decision theory tells us that the best prediction for input $\mathbf{x}$, is to choose the class with highest joint $p(\mathbf{x}, C_k)$
  - Or equivalently: choose class with highest posterior $p(C_k \mid \mathbf{x})$
  - Decision boundary between C_k and C_j are at $p(C_k \mid \mathbf{x}) = p(C_j \mid \mathbf{x})$

Probabilistic Generative Models: $K = 2$

- Class-conditional densities:
- Prior class probabilities:
- Joint distribution:
- Posterior distribution: $K = 2$

$$p(C_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_1)p(C_1) + p(\mathbf{x} \mid C_2)p(C_2)} =$$

- Log odds $a = \ln \frac{\sigma}{1 - \sigma} =$

Logistic Sigmoid Function

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

$$\sigma(-a) = 1 - \sigma(a)$$

$$\sigma'(a) = \sigma(a)(1 - \sigma(a))$$

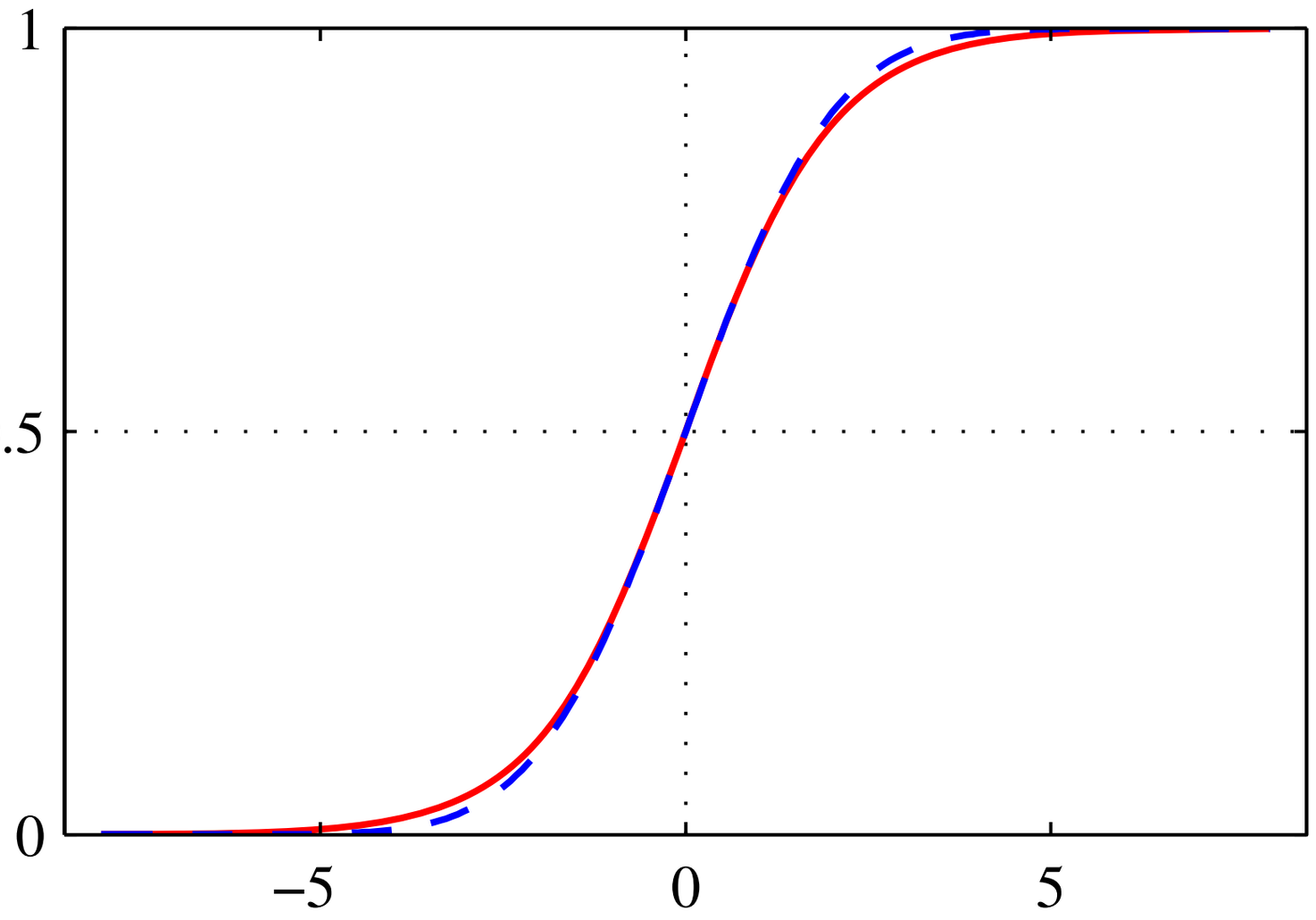


Figure: Logistic Sigmoid function (red) (Bishop 4.9)

Probabilistic Generative Models: general K

- For multiple classes (general K):

$$p(C_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_k)p(C_k)}{\sum_{j=1}^{K} p(\mathbf{x} \mid C_j)p(C_j)} =$$

$$a_k = \ln(p(\mathbf{x} \mid C_k)p(C_k))$$

- Softmax: if $a_k \gg a_j$ for all $j \neq k$:

- Note: for $K = 2$:

$$p(C_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_1)p(C_1) + p(\mathbf{x} \mid C_2)p(C_2)} = \frac{1}{1 + \frac{p(\mathbf{x} \mid C_2)p(C_2)}{p(\mathbf{x} \mid C_1)p(C_1)}}$$

$$=$$

$$a = \frac{p(\mathbf{x} \mid C_2)p(C_2)}{p(\mathbf{x} \mid C_1)p(C_1)}$$

Class Conditional Densities: Continuous Inputs

- Gaussian Class-conditional densities:

$$p(\mathbf{x} \mid C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_k|^{1/2}} \exp\{\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\}$$

- Class posteriors simplify with the assumption of shared covariance matrix: $\Sigma_k = \Sigma$

- For $K = 2$ classes they are given as $p(C_1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-a)} = \sigma(a)$ with

$$a = \ln \frac{p(\mathbf{x} \mid C_1) p(C_1)}{p(\mathbf{x} \mid C_2) p(C_2)} =$$

Class Conditional Densities: Continuous Inputs

- Gaussian Class-conditional densities:

$$p(\mathbf{x}\,|\,C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\,\Sigma_k\,|^{1/2}} \exp\{\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\}$$

- Class posteriors simplify with the assumption of shared covariance matrix: $\Sigma_k = \Sigma$

- For $K = 2$ classes they are given as $p(C_1\,|\,\mathbf{x}) = \dfrac{1}{1+\exp(-a)} = \sigma(a)$ with

$$a = \ln \frac{p(\mathbf{x}\,|\,C_1)p(C_1)}{p(\mathbf{x}\,|\,C_2)p(C_2)} = \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_1, \Sigma) \;-\; \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_2, \Sigma) \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

Class Conditional Densities: Continuous Inputs

- Gaussian Class-conditional densities:

$$p(\mathbf{x}\,|\,C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\,\Sigma_k\,|^{1/2}} \exp\{\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\}$$

- Class posteriors simplify with the assumption of shared covariance matrix: $\Sigma_k = \Sigma$

- For $K = 2$ classes they are given as $p(C_1\,|\,\mathbf{x}) = \dfrac{1}{1+\exp(-a)} = \sigma(a)$ with

$$a = \ln \frac{p(\mathbf{x}\,|\,C_1)p(C_1)}{p(\mathbf{x}\,|\,C_2)p(C_2)} = \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_1, \Sigma) \;-\; \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_2, \Sigma) \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

$$= -\frac{1}{2}\ln|\,\Sigma\,| \;-\; \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \;+\; \frac{1}{2}\ln|\,\Sigma\,| + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

Class Conditional Densities: Continuous Inputs

- Gaussian Class-conditional densities:

$$p(\mathbf{x}\,|\,C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\,\Sigma_k\,|^{1/2}} \exp\{\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\}$$

- Class posteriors simplify with the assumption of shared covariance matrix: $\Sigma_k = \Sigma$

- For $K = 2$ classes they are given as $p(C_1\,|\,\mathbf{x}) = \dfrac{1}{1+\exp(-a)} = \sigma(a)$ with

$$a = \ln \frac{p(\mathbf{x}\,|\,C_1)p(C_1)}{p(\mathbf{x}\,|\,C_2)p(C_2)} = \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_1, \Sigma) \;-\; \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_2, \Sigma) \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

$$= -\frac{1}{2}\ln|\,\Sigma\,| \;-\; \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \;+\; \frac{1}{2}\ln|\,\Sigma\,| + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

$$= (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T\Sigma^{-1}\mathbf{x} \;-\; \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 \;+\; \frac{1}{2}\boldsymbol{\mu}_2\Sigma^{-1}\boldsymbol{\mu}_2 \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

Class Conditional Densities: Continuous Inputs

- Gaussian Class-conditional densities:

$$p(\mathbf{x}\,|\,C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\,\Sigma_k\,|^{1/2}} \exp\{\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\}$$

- Class posteriors simplify with the assumption of shared covariance matrix: $\Sigma_k = \Sigma$

- For $K = 2$ classes they are given as $p(C_1\,|\,\mathbf{x}) = \dfrac{1}{1+\exp(-a)} = \sigma(a)$ with

$$a = \ln \frac{p(\mathbf{x}\,|\,C_1)p(C_1)}{p(\mathbf{x}\,|\,C_2)p(C_2)} = \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_1,\Sigma) \;-\; \ln \mathcal{N}(\mathbf{x}\,|\boldsymbol{\mu}_2,\Sigma) \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

$$= -\frac{1}{2}\ln|\,\Sigma\,| \;-\; \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \;+\; \frac{1}{2}\ln|\,\Sigma\,| + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

$$= (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T\Sigma^{-1}\mathbf{x} \;-\; \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 \;+\; \frac{1}{2}\boldsymbol{\mu}_2\Sigma^{-1}\boldsymbol{\mu}_2 \;+\; \ln \frac{p(C_1)}{p(C_2)}$$

- Generalized Linear Model: $p(C_1\,|\,\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}+w_0)$

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(C_1)}{p(C_2)}$$

Example: Linear Discriminant Analysis for K=2

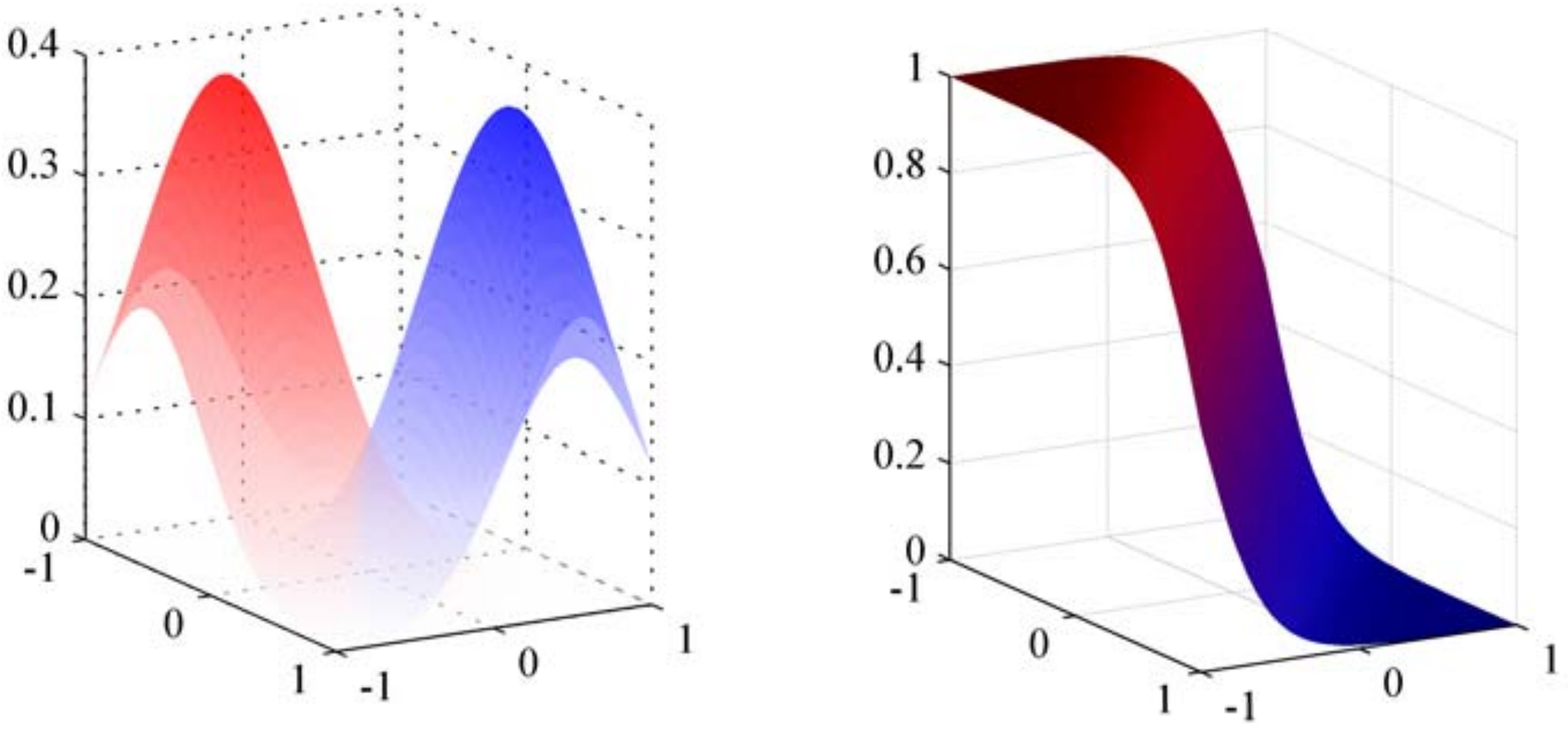


Figure: Left: class conditional densities $p(\mathbf{x} \mid C_k)$. Right: posterior $p(C_1 \mid \mathbf{x})$ as sigmoid of linear function of $\mathbf{x}$. (Bishop 4.9)

Linear Discriminant Analysis: General K

- Gaussian Class-conditional densities & fixed covariance:

$$p(\mathbf{x}\,|\,C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_k|^{1/2}} \exp\{\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\}$$

- Posterior distributions:

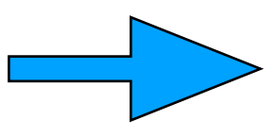
$$p(C_k\,|\,\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_{j=1}^{K}\exp(a_j(\mathbf{x}))}$$

- $a_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x} + w_{k0}$

$$\mathbf{w}_k = \Sigma^{-1}\boldsymbol{\mu}_k$$
$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k\Sigma^{-1}\boldsymbol{\mu}_k + \ln p(C_k)$$

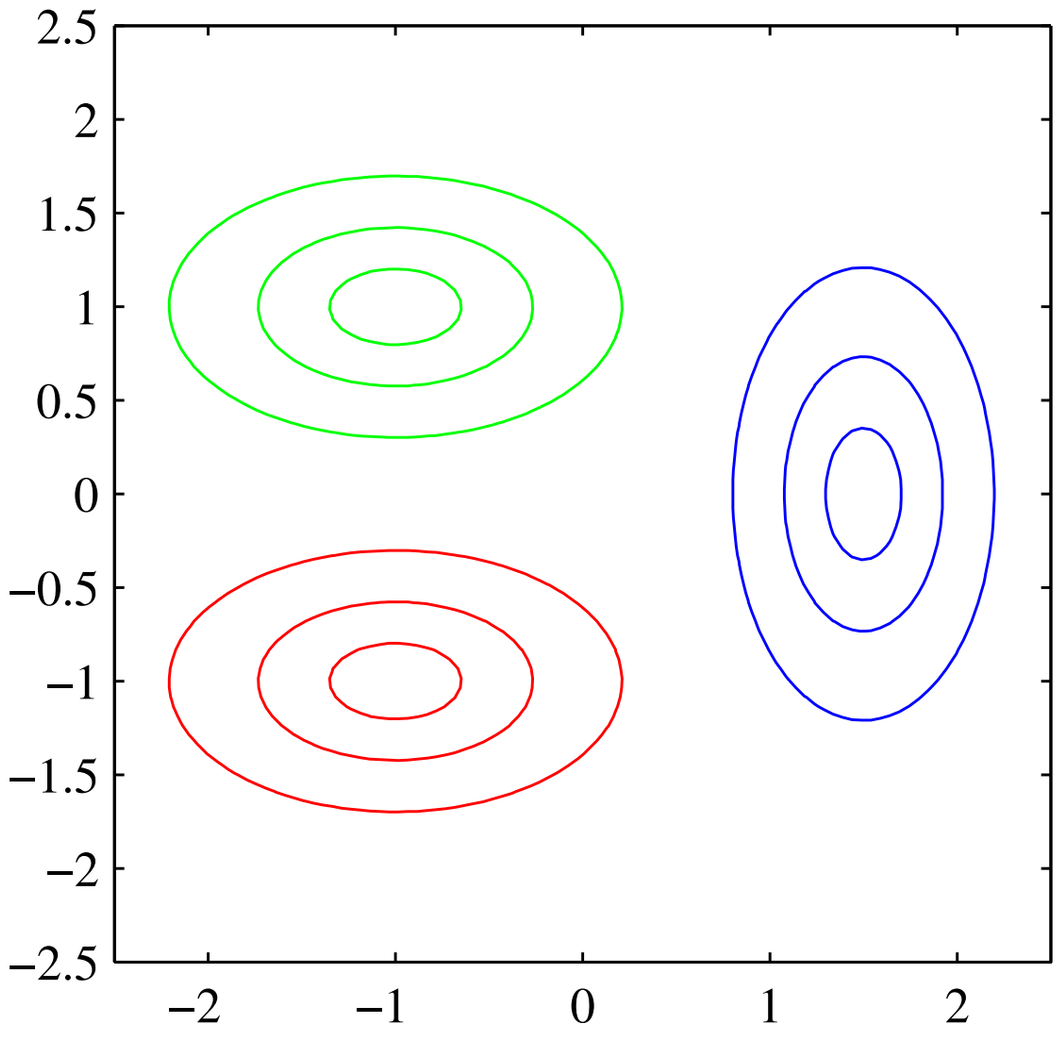
- Decision boundary:

$$p(C_k\,|\,\mathbf{x}) = p(C_j\,|\,\mathbf{x})$$



- If all covariance matrices are different $\Sigma_k \neq \Sigma_j$ then $a_k(\mathbf{x})$ also contains quadratic terms in $\mathbf{x}$

Example: LDA and QDA



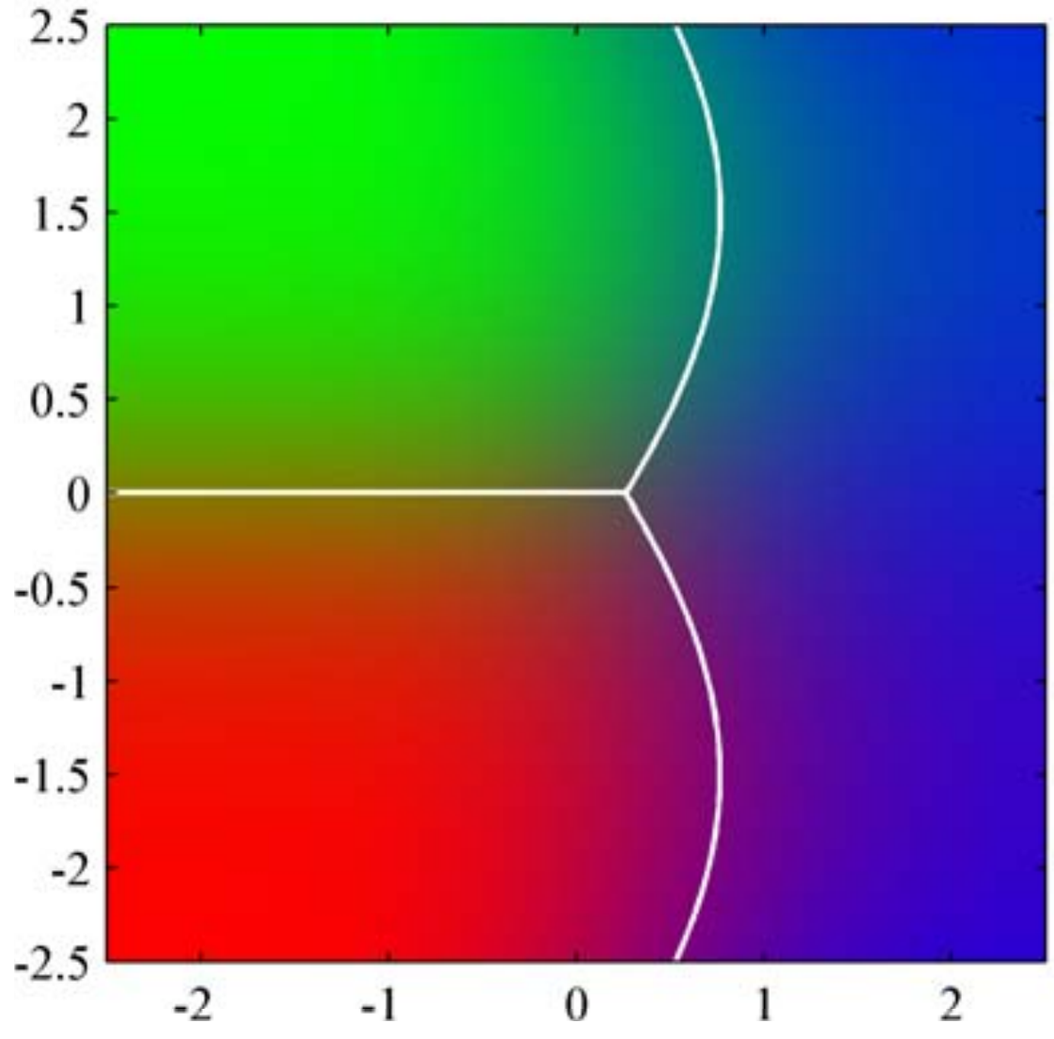


Figure: Left: Gaussian class conditional densities $p(\mathbf{x} \mid C_k)$, red and green have same covariance matrix. Right: posterior $p(C_k \mid \mathbf{x})$ distributions (RGB vectors) and decision boundaries. (Bishop 4.9)