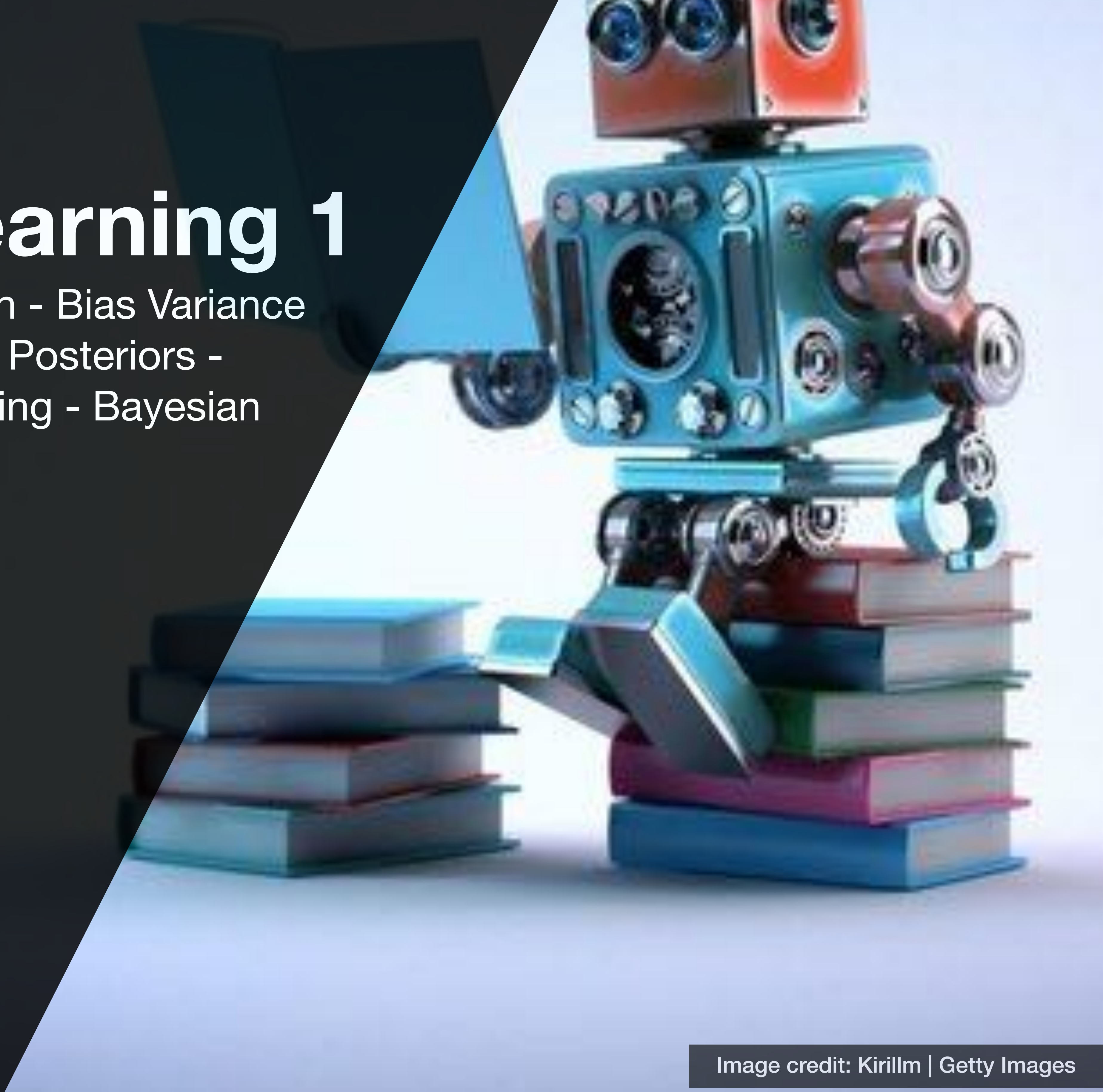
UNIVERSITY OF AMSTERDAM
Informatics Institute



Machine Learning 1

Lecture 4 - Model Selection - Bias Variance Decomposition - Gaussian Posteriors - Sequential Bayesian Learning - Bayesian Predictive Distributions

Erik Bekkers

Image credit: Kirillm | Getty Images

UNIVERSITY OF AMSTERDAM
Informatics Institute

AMLAB Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 3.5 - Supervised Learning

Regularized Least Squares

Erik Bekkers

(Bishop 3.1.4)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Parameter estimates with Gaussians

- Given **Likelihood**/Data model:

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

- The ***ML parameter estimate*** is obtained via **least squares**:

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

- Additionally, given Gaussian weight **Prior**:

$$p(\mathbf{w} \mid \alpha) = \prod_{i=1}^{M} \mathcal{N}(w_i \mid 0, \alpha^{-1})$$

Ridge regr.

$\lambda = \alpha / \beta$

- The **MAP parameter estimate** is obtained via **regularized least squares**:

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

Example: Regularized Polynomial Regression

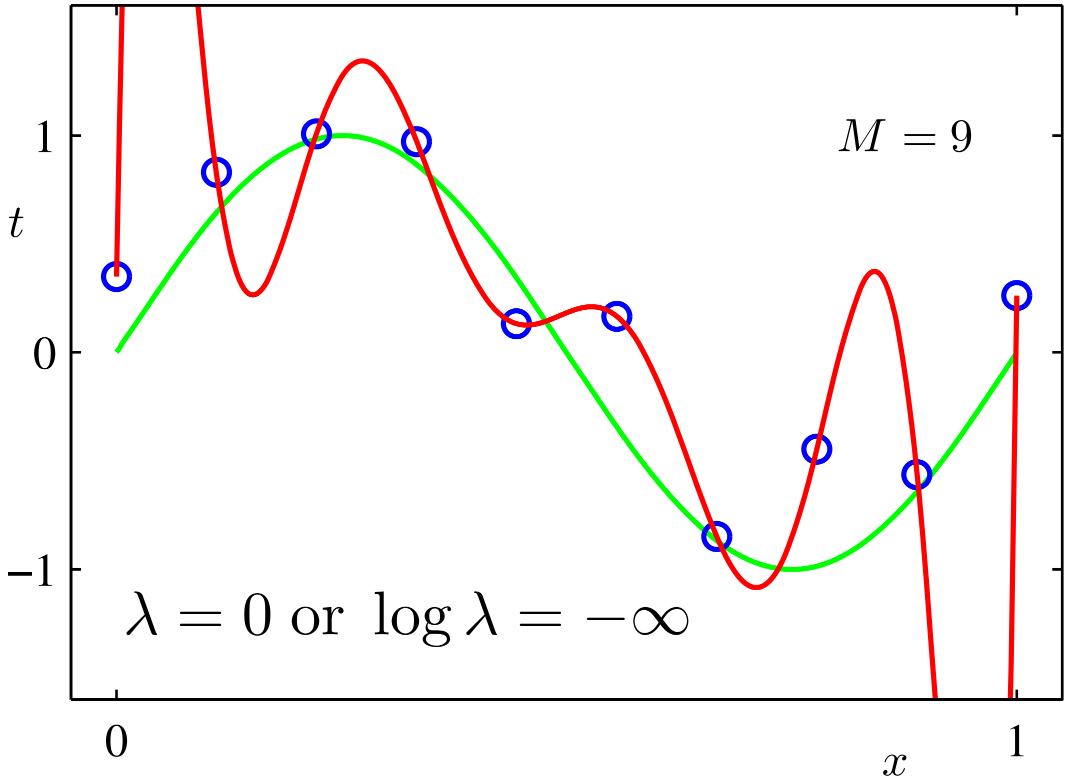


Figure: polynomial regression (Bishop 1.4)

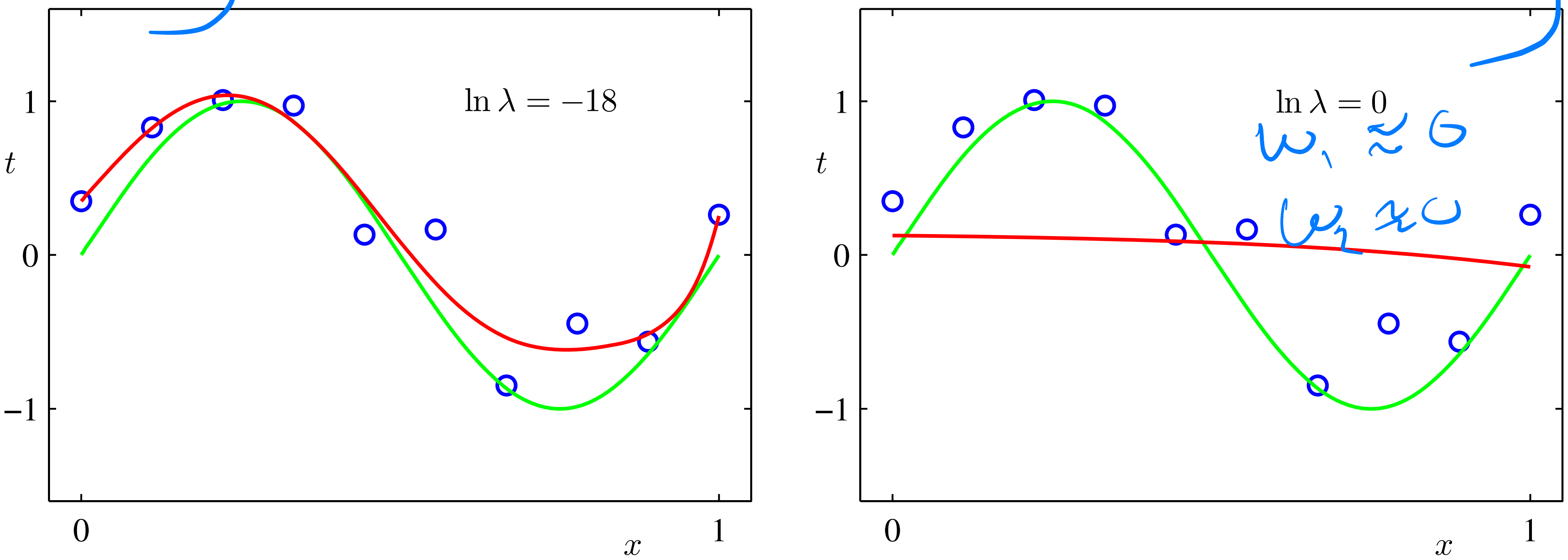


Figure: Regularized polynomial regression (Bishop 1.7)

Example: Regularized Polynomial Regression

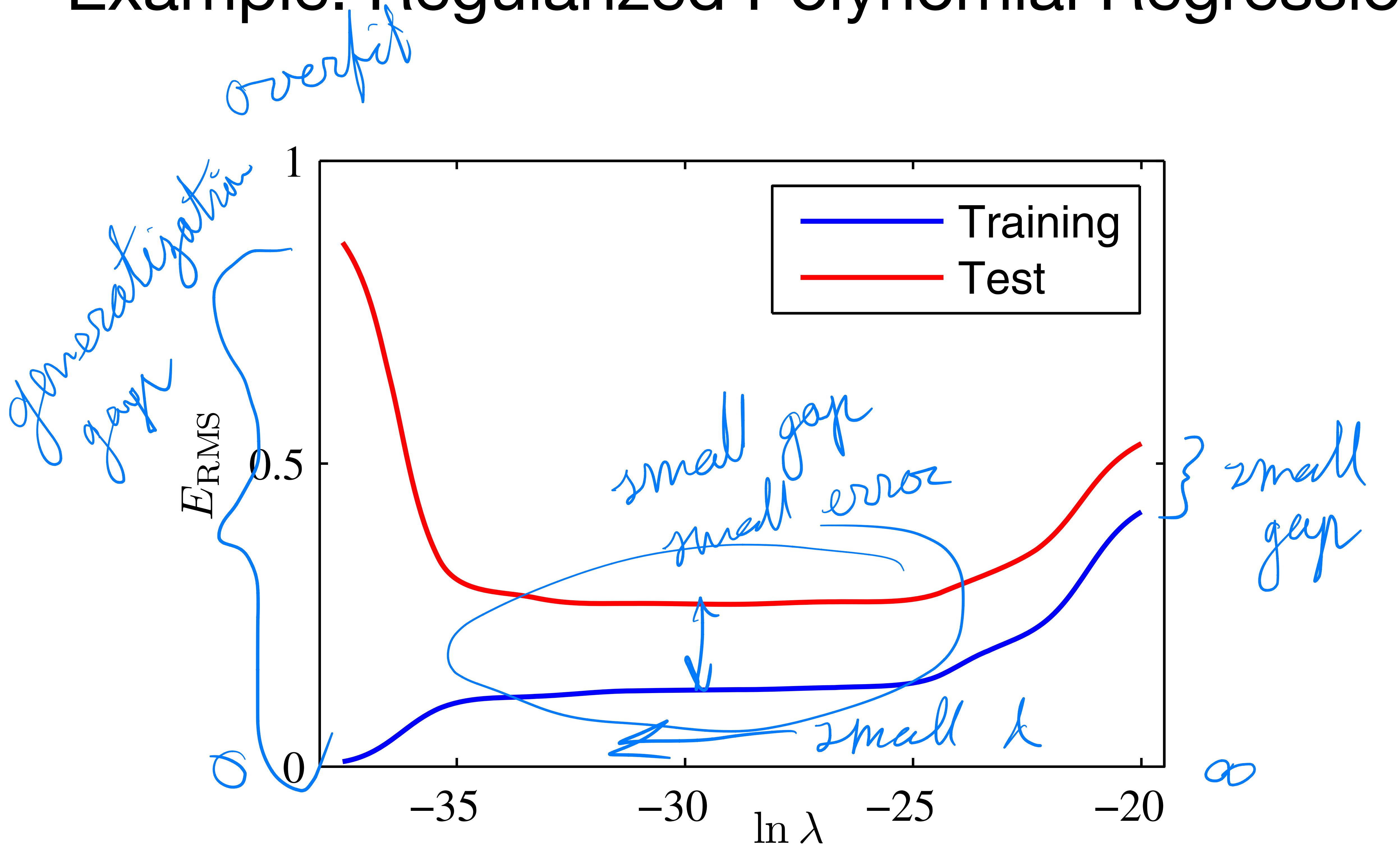


Figure: train and test errors for regularized M=9 polynomial regression (Bishop 1.8)

Regularized Least Squares: sparse weights

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{N}\{t_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)\}^2 \quad \text{subject to constraint} \quad \sum_{i=1}^{M}|w_i|^q \leq \eta$$

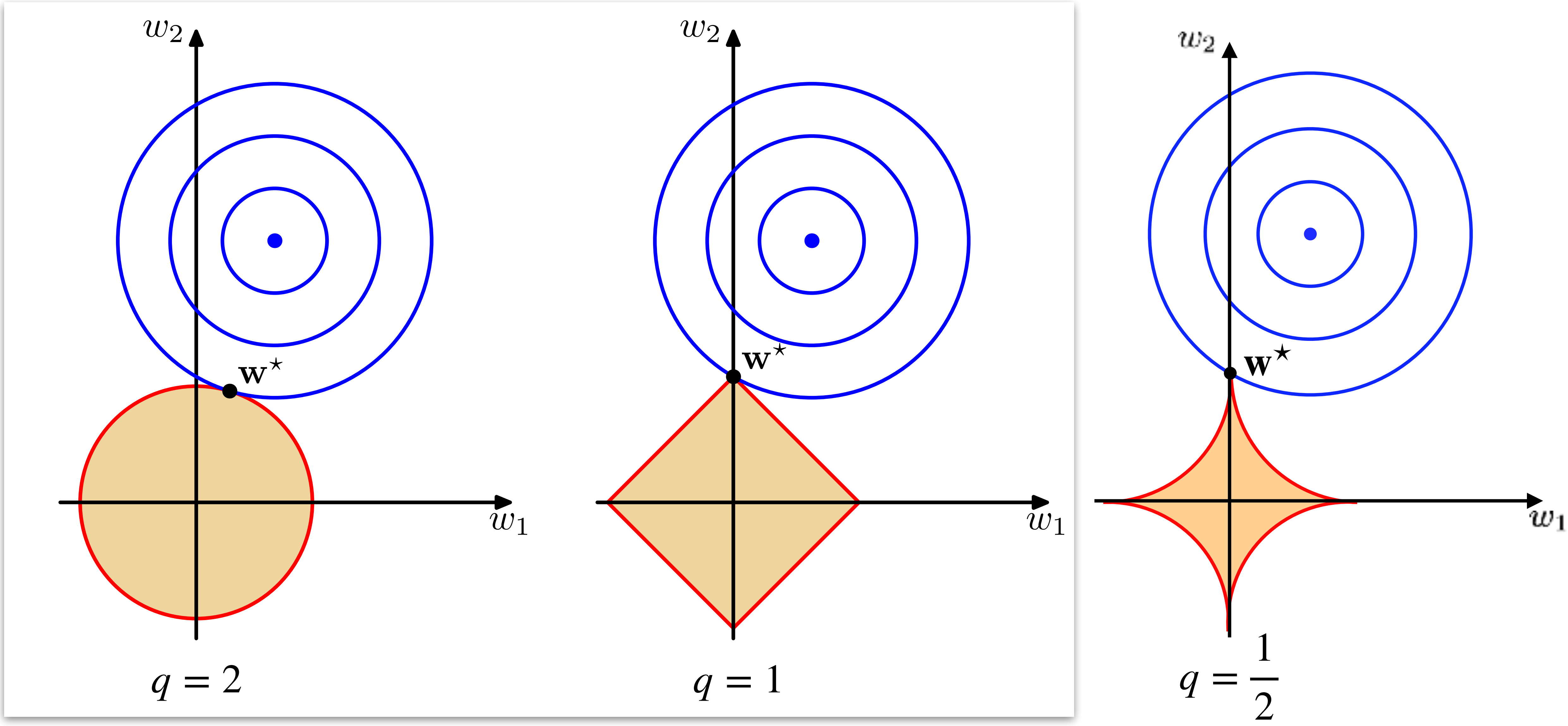


Figure: regularization as constrained optimization (Bishop 3.4)

University of Amsterdam
Informatics Institute

AMLAB Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 4.1 - Supervised Learning

Model Selection

Erik Bekkers

(Bishop 1.3)

λ is a hyperparameter

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

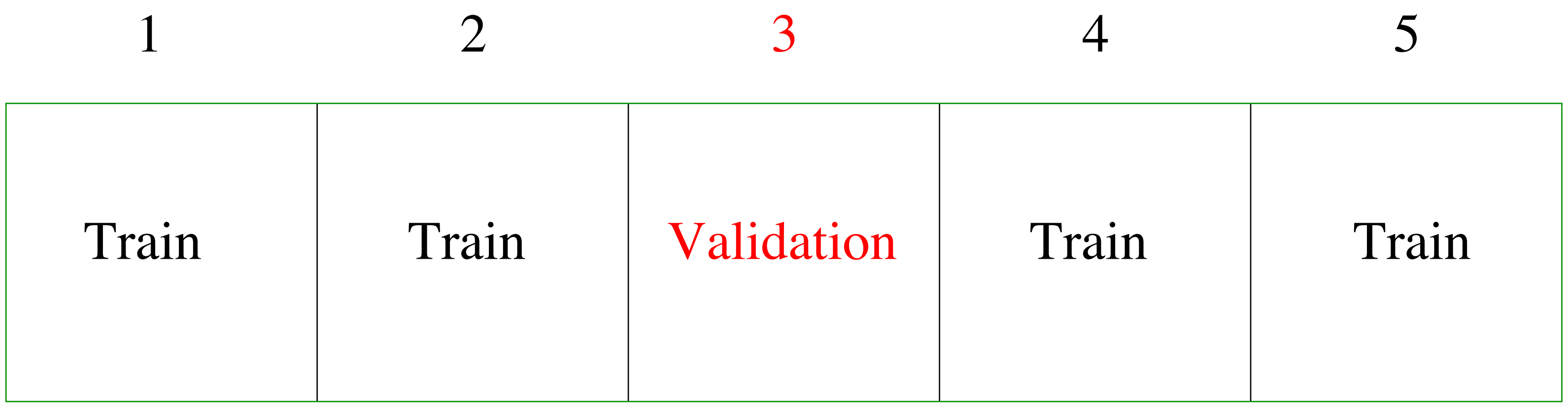
Supervised Learning: Evaluating Errors

Q: How can we reliably estimate the model performance properly for unknown data?

Q: How can we choose the optimal hyperparameters?

Supervised Learning: Small Datasets

- Small dataset → small validation and test set
- Noisy model selection and estimate of generalization error
- Cross-validation:
  - Split data $D = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$ in to K folds
  - Train model y on $K - 1$ folds (fold k left out) → $\hat{y}^{-k}$

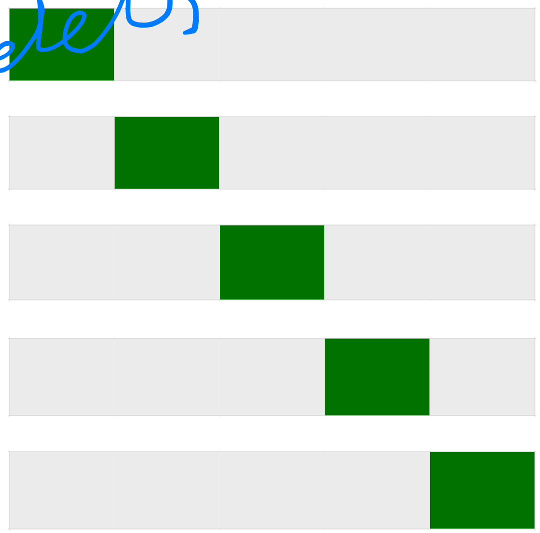


1	2	3	4	5
Train	Train	Validation	Train	Train

Figure: K-fold splitting of dataset (ESL 7.10)

  - Leave-one-out cross validation: $K = N$

Cross-Validation: Model Selection



K models

- Hyperparameter selection: α
  - $$CV(\hat{y}_\alpha) = \frac{1}{N}\sum_{i=1}^{N} E(\hat{y}_\alpha^{-\kappa(i)}(\mathbf{x}_i), t)$$
  - $\alpha^* = \underset{\alpha}{\operatorname{argmax}}\ CV(\hat{y}_\alpha)$
- Multiple hyperparameters: $\alpha \in \{\alpha_1, \alpha_2\}, \beta \in \{\beta_1, \beta_2, \beta_3\}$
  - How many times should CV be performed?

    $\overset{\alpha}{2} \times \overset{\beta}{3} = 6$
  - Total number of training runs?

    $6 \times K$

Nested Cross-Validation

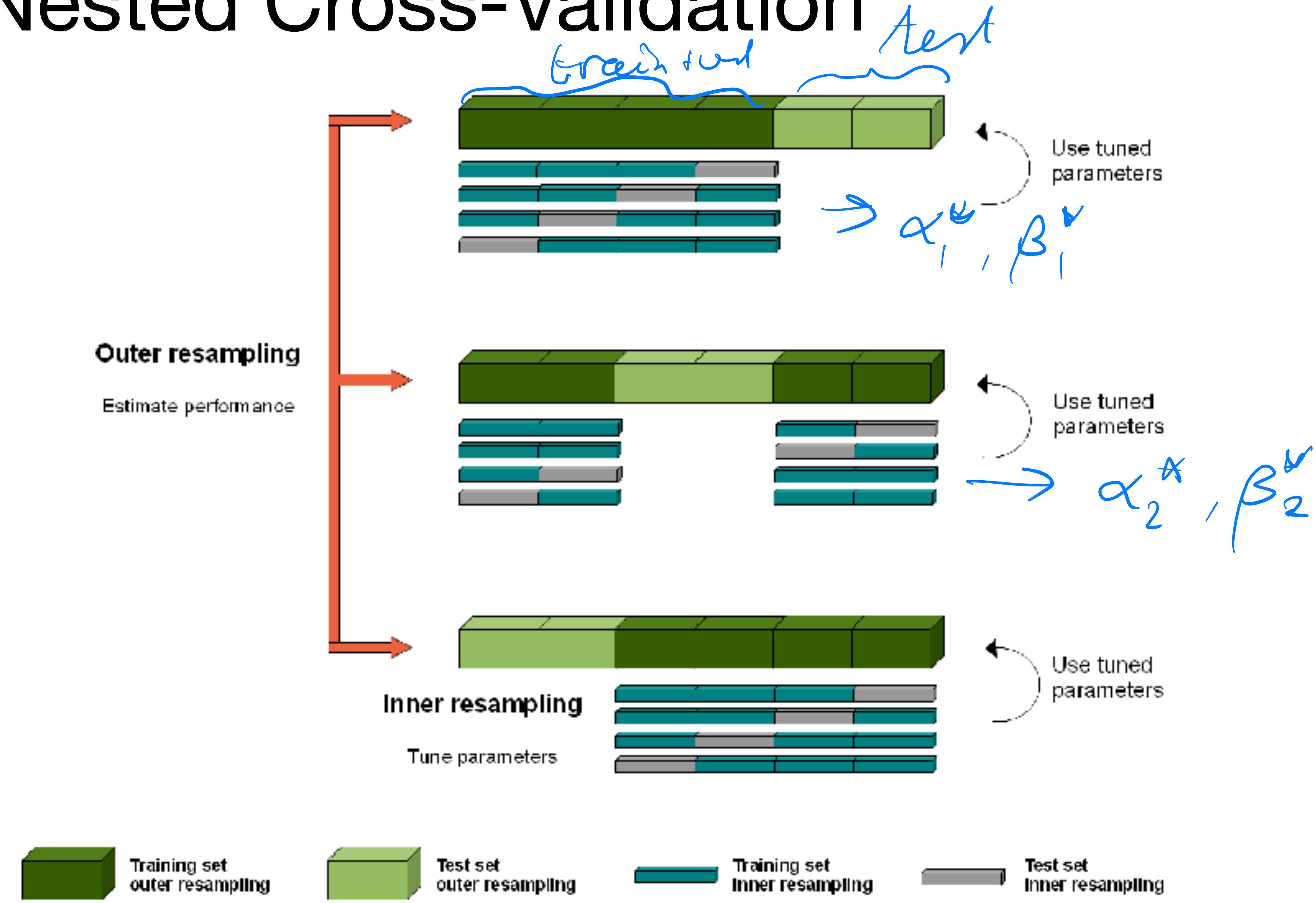


Figure: Nested cross-validation
https://mlr-org.github.io/mlr-tutorial/devel/html/nested_resampling/index.html
(site is offline unfortunately)

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)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

UNIVERSITY OF AMSTERDAM
Informatics Institute

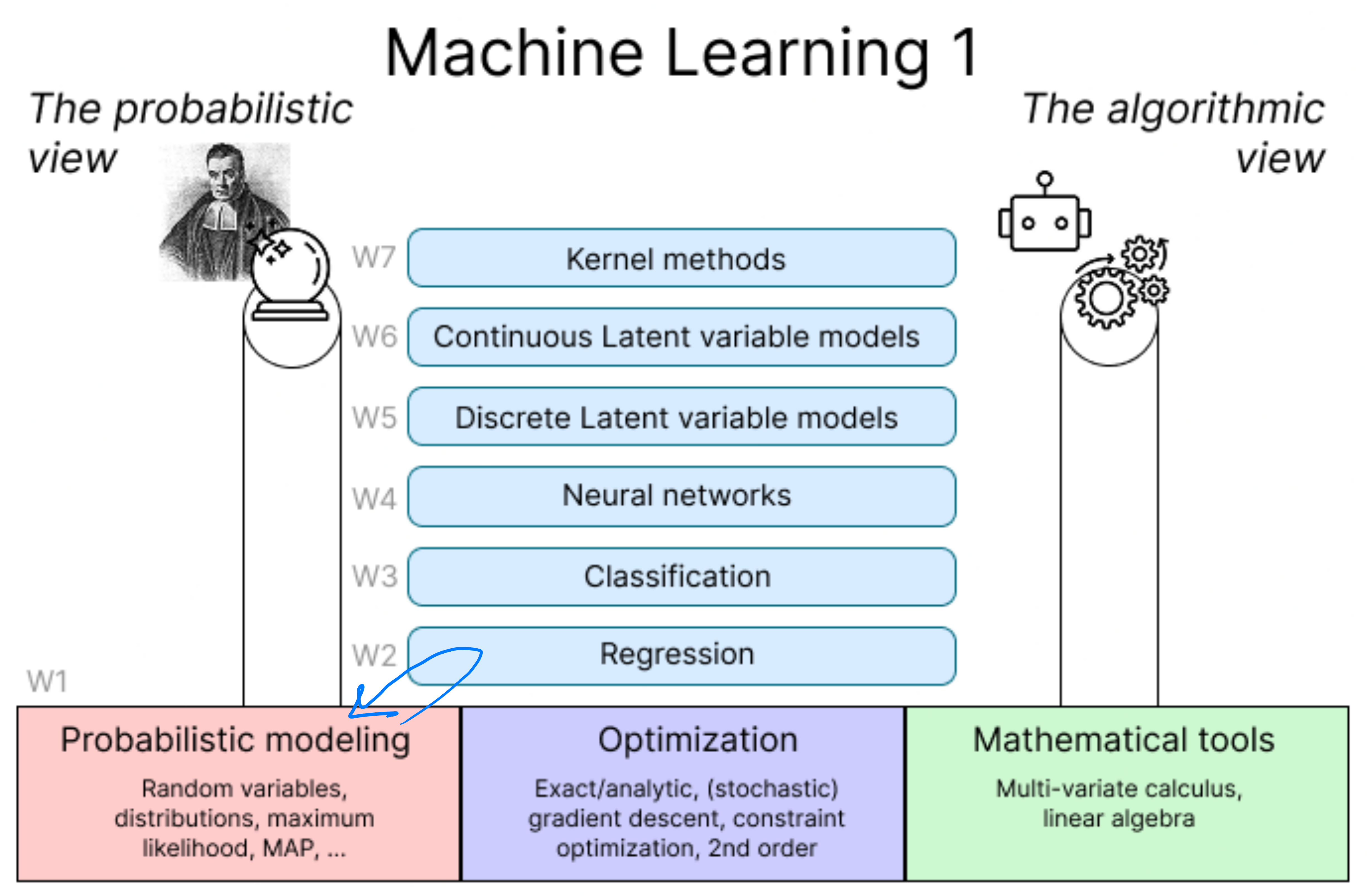
M

Le

Bia

Eri

(Bi



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
Random variables, distributions, maximum likelihood, MAP, ...

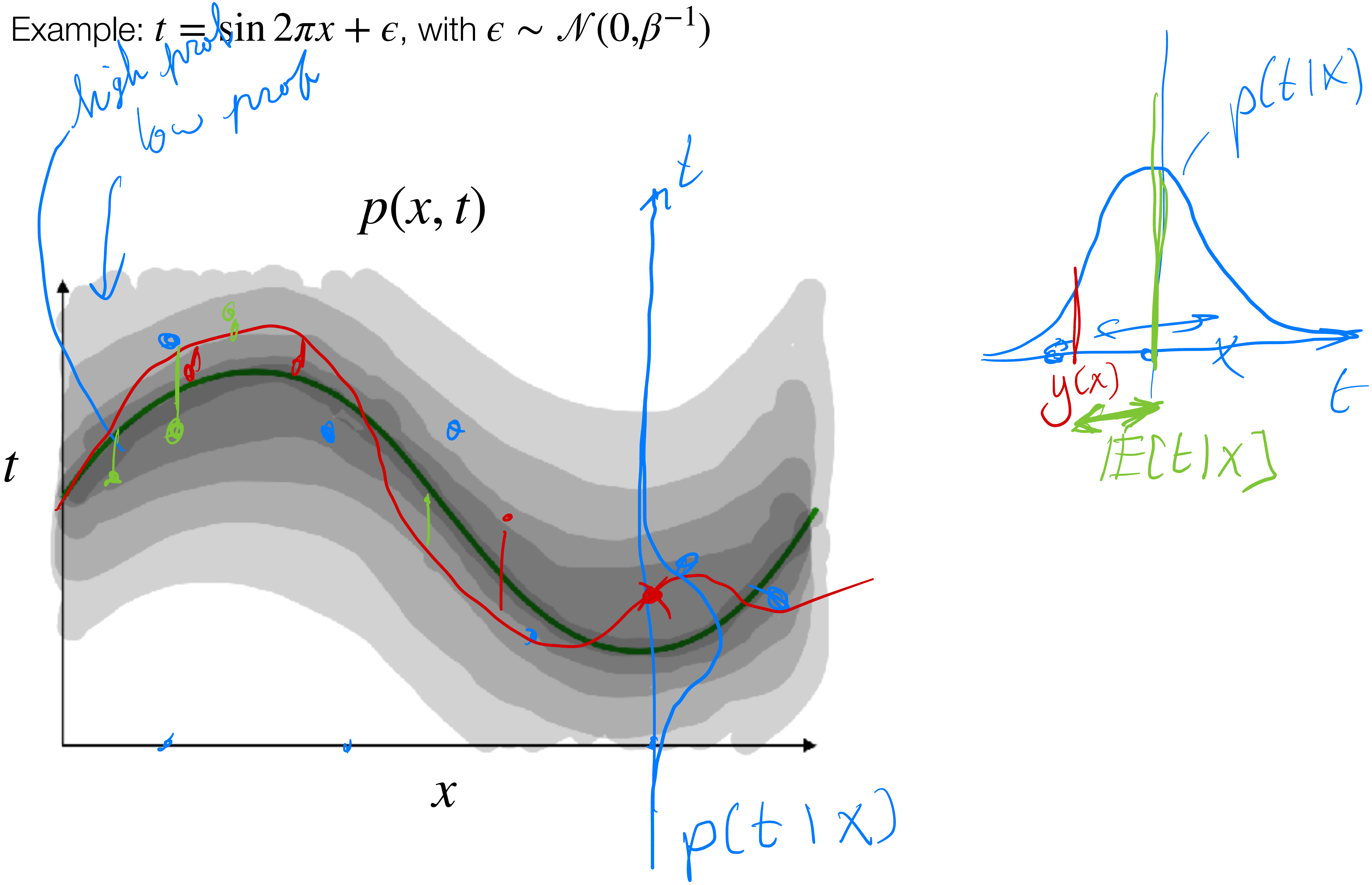
Optimization
Exact/analytic, (stochastic) gradient descent, constraint optimization, 2nd order

Mathematical tools
Multi-variate calculus, linear algebra

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

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



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

- Consider the expected loss:

$$\mathbb{E}[L] = \int\int (y(\mathbf{x}) - t)^2 p(\mathbf{x}, t)\mathrm{d}t\mathrm{d}\mathbf{x}$$

- Let's analyze it relative to the regression function $\mathbb{E}[t \,|\, \mathbf{x}] := \mathbb{E}_{t\sim p(t|\mathbf{x})}[t \,|\, \mathbf{x}]$:

$$\mathbb{E}[L] = \int\int (\underbrace{y(\mathbf{x}) - \mathbb{E}[t \,|\, \mathbf{x}]}_{a} + \underbrace{\mathbb{E}[t \,|\, \mathbf{x}] - t}_{b})^2 p(\mathbf{x}, t)\mathrm{d}t\mathrm{d}\mathbf{x}$$

Summary so far…

- Any model y will make errors, this expected loss decomposes into

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

- The best possible model is the regression function $y(\mathbf{x}) = \mathbb{E}[t \mid \mathbf{x}]$

$$D_1 = \{(x_1, t_1), \ldots, (x_N, t_N)\} \rightarrow y_{D_1}$$

$$D_2 = \{(x_1, t_1), \ldots, (x_N, t_N)\} \rightarrow y_{D_2}$$

- In practice we approximate it with fits $y_D = \underset{y}{\mathrm{argmin}} \sum_{(\mathbf{x},t)\in D} L(t, y(\mathbf{x}))$

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

The Average Expected Loss

- Let's analyze performance of a learning algorithm by averaging the expected loss over learned y_D for different datasets D

$$\mathbb{E}_D[\mathbb{E}[L]] = \int \mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}[t\,|\,\mathbf{x}])^2]\; p(\mathbf{x})\mathrm{d}\mathbf{x} + \int \mathrm{var}[t\,|\,\mathbf{x}]\; p(\mathbf{x})\mathrm{d}\mathbf{x}$$

The Average Expected Loss

- Let's analyze performance of a learning algorithm by averaging the expected loss over learned y_D for different datasets D

$$\mathbb{E}_D[\mathbb{E}[L]] = \int \mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}[t\,|\,\mathbf{x}])^2]\; p(\mathbf{x})\mathrm{d}\mathbf{x} + \int \mathrm{var}[t\,|\,\mathbf{x}]\; p(\mathbf{x})\mathrm{d}\mathbf{x}$$

- Analyze it relative to the average model $\mathbb{E}_D[y_D(\mathbf{x})]$

$$\mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}[t\,|\,\mathbf{x}])^2] = \mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}_D[y_D(\mathbf{x})] + \mathbb{E}_D[y_D(\mathbf{x})] - \mathbb{E}[t\,|\,\mathbf{x}])^2]$$

The Average Expected Loss

- Let's analyze performance of a learning algorithm by averaging the expected loss over learned y_D for different datasets D

$$\mathbb{E}_D[\mathbb{E}[L]] = \int \mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}[t\,|\,\mathbf{x}])^2]\; p(\mathbf{x})\mathrm{d}\mathbf{x} + \int \mathrm{var}[t\,|\,\mathbf{x}]\; p(\mathbf{x})\mathrm{d}\mathbf{x}$$

- Analyze it relative to the average model $\mathbb{E}_D[y_D(\mathbf{x})]$

$$\mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}[t\,|\,\mathbf{x}])^2] = \mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}_D[y_D(\mathbf{x})] + \mathbb{E}_D[y_D(\mathbf{x})] - \mathbb{E}[t\,|\,\mathbf{x}])^2]$$

(Expand square $(a+b)^2 = a^2 + 2ab + b^2$)

$$= \mathbb{E}_D[(y_D(\mathbf{x}) - \mathbb{E}_D[y_D(\mathbf{x})])^2] + \mathbb{E}_D[(\mathbb{E}_D[y_D(\mathbf{x})] - \mathbb{E}[t\,|\,\mathbf{x}])^2]$$

$$+2\; \mathbb{E}_D[\cancel{(y_D(\mathbf{x}) - \mathbb{E}_D[y_D(\mathbf{x})])(\mathbb{E}_D}[y_D(\mathbf{x})] - \mathbb{E}[t\,|\,\mathbf{x}])]$$

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:

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

Variance

$$+ \int (\mathbb{E}_D[y_D(x)] - \mathbb{E}[t \mid x])^2\, p(x)\mathrm{d}x$$

Bias ^{2}

$$+ \int \mathrm{var}[t \mid x]\, p(x)\mathrm{d}x$$

Noise

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:

$$\mathbb{E}_D[\mathbb{E}[L]] \approx \frac{1}{N}\sum_{\underline{n=1}}^{N}\frac{1}{L}\sum_{\underline{l=1}}^{L}(y^{(l)}(x_n)-\bar{y}(\mathbf{x}_n))^2] \qquad \text{Variance}$$

$$+\frac{1}{N}\sum_{i=1}^{N}\bar{y}(x_n)-\sin 2\pi x_n)^2 \qquad \text{Bias}^{\,2}$$

see lecture 1
for finite dataset
approximation

Bias-Variance Decomposition: Example

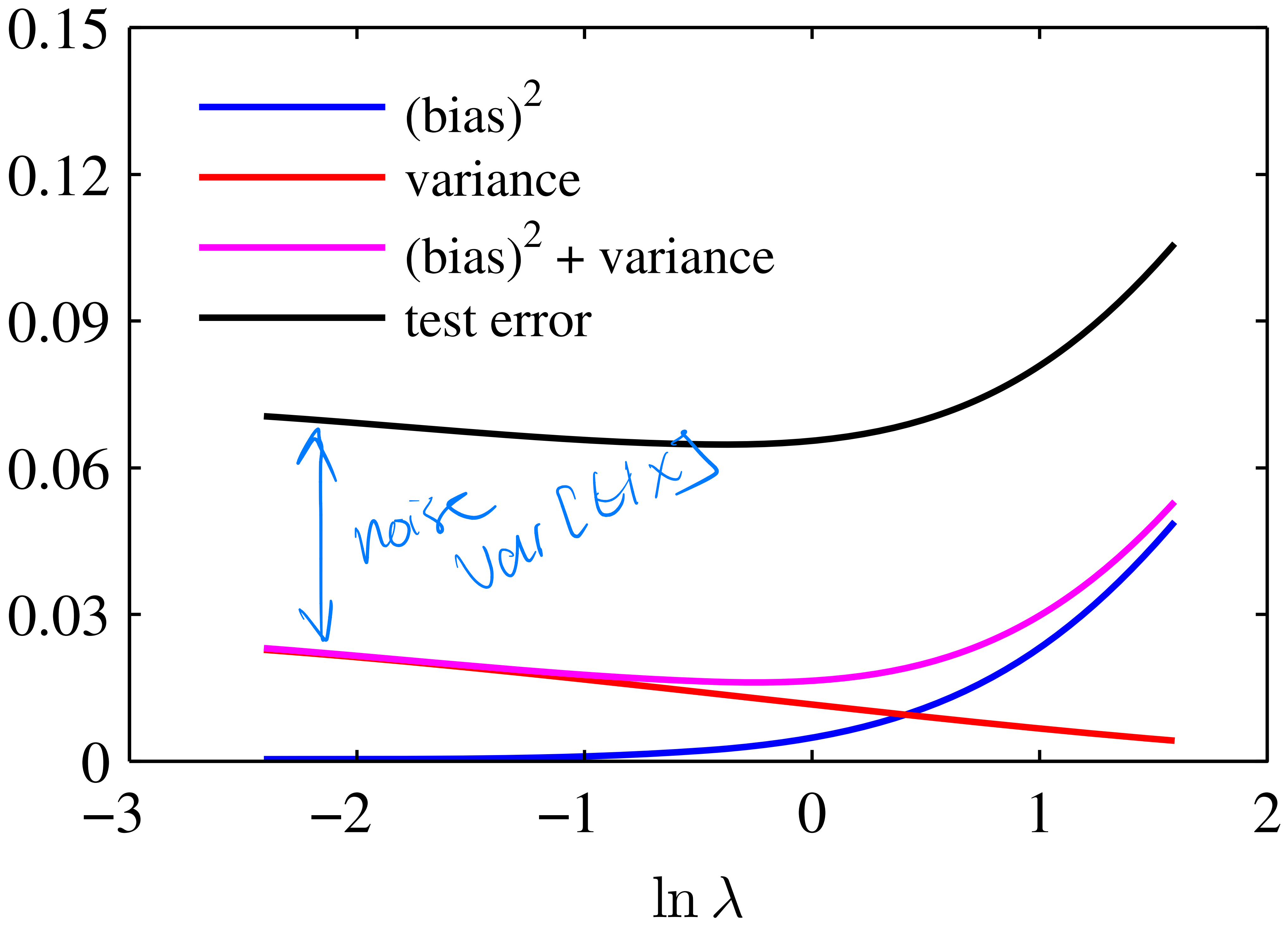


Figure: bias-variance decomposition (Bishop 3.6)

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

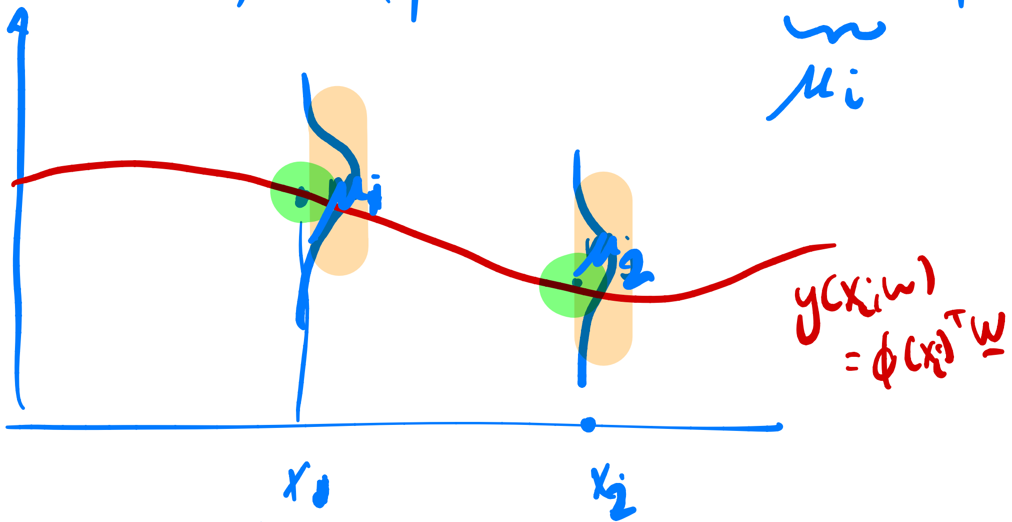
(Bishop 3.3.1 (and 2.3.3)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

(*)

$$p(t_i \mid x_i, \underline{w}, \beta^{-1}) = \mathcal{N}(t_i \mid \underbrace{\phi_i^T \underline{w}}_{\mu_i}, \beta^{-1})$$



$$\underline{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \underline{\phi}(x_1)^T w \\ \underline{\phi}(x_2)^T w \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots \\ \vdots & & \\ \phi_1(x_N) & \phi_2(x_N) & \cdots \end{pmatrix}}_{N \times M} \underline{w}$$

$$= \Phi \underline{w}$$

predict all means at once

$$\Sigma = \beta^{-1} I_N = \overbrace{\begin{pmatrix} \beta^{-1} & 0 & \\ & \beta^{-1} & \\ 0 & & \ddots \end{pmatrix}}^{N \times N}$$

Since independent and shared β

(*) $$\prod_{i=1}^{N} e^{-\frac{\beta}{2}(t_i - \underline{\phi}_i^T \underline{w})^2}$$

$$= e^{-\frac{\beta}{2}\sum_{i=1}^{N}(t_i - \underline{\phi}_i^T \underline{w})^2}$$

$$= e^{-\frac{\beta}{2}(\underline{t} - \Phi \underline{w})^T(\underline{t} - \Phi \underline{w})}$$

$$= e^{-\frac{\beta}{2}(\underline{t} - \Phi w) I (\underline{t} - \Phi w)}$$

so $\underline{\mu} = \Phi w \quad \in \mathbb{R}^N$

$\Sigma = \beta^{-1} I_N \quad \in \mathbb{R}^{N \times N}$

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)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

UNIVERSITY OF AMSTERDAM
Informatics Institute

Machine Learning 1

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

Erik Bekkers

(Bishop 3.3.1)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

University of Amsterdam
Informatics Institute

AMLAB Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 4.5 - Supervised Learning Bayesian Linear Regression - Predictive Distribution

Erik Bekkers

(Bishop 3.3.2)

Image credit: Kirillm | Getty Images

Slide credits: Patrick Forré and Rianne van den Berg

Predictive Distribution

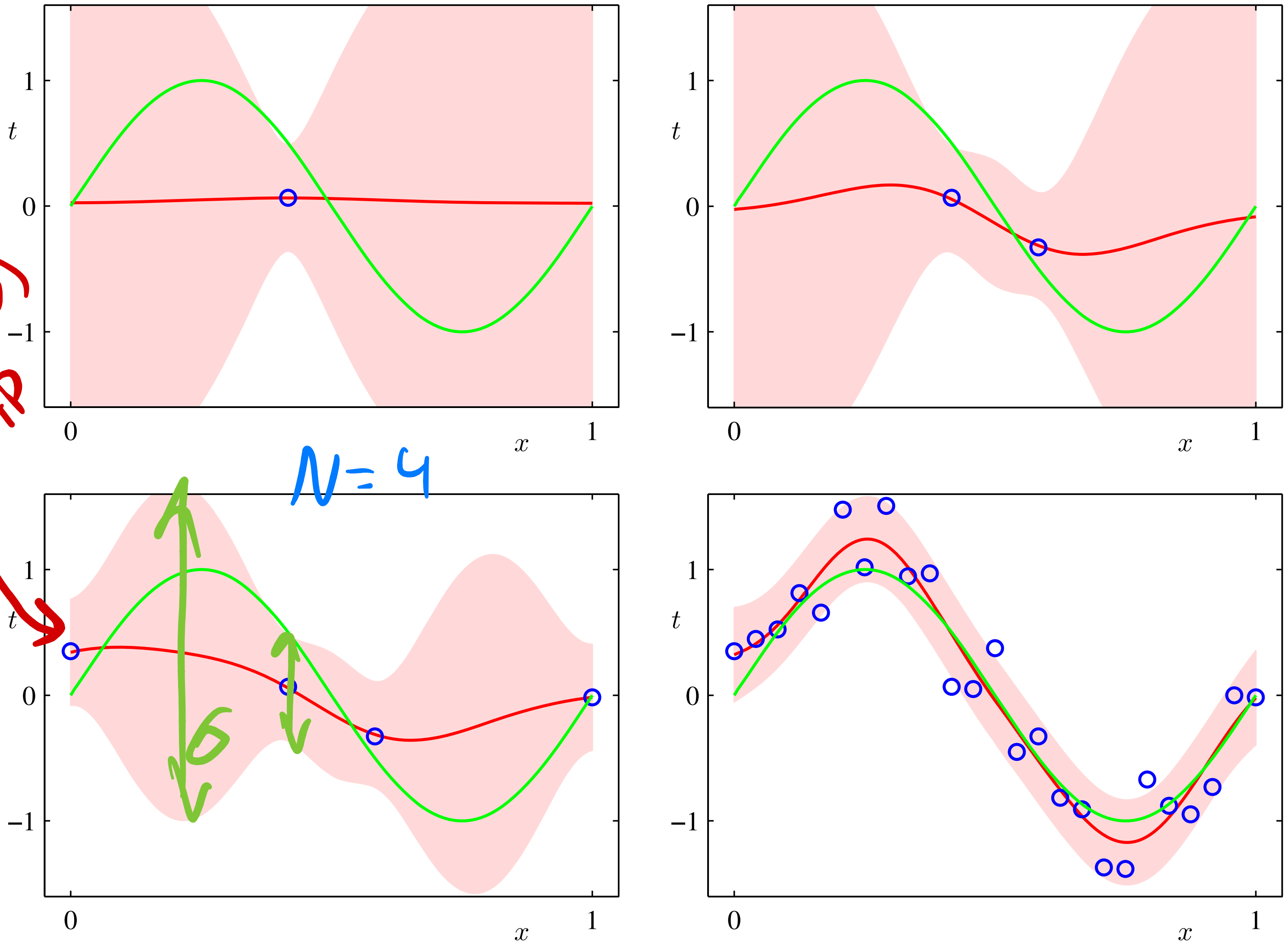
active learning?

- 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

mean $y(x, w)$ given by MAP

$N=1$ $N=2$

$N=4$



σ

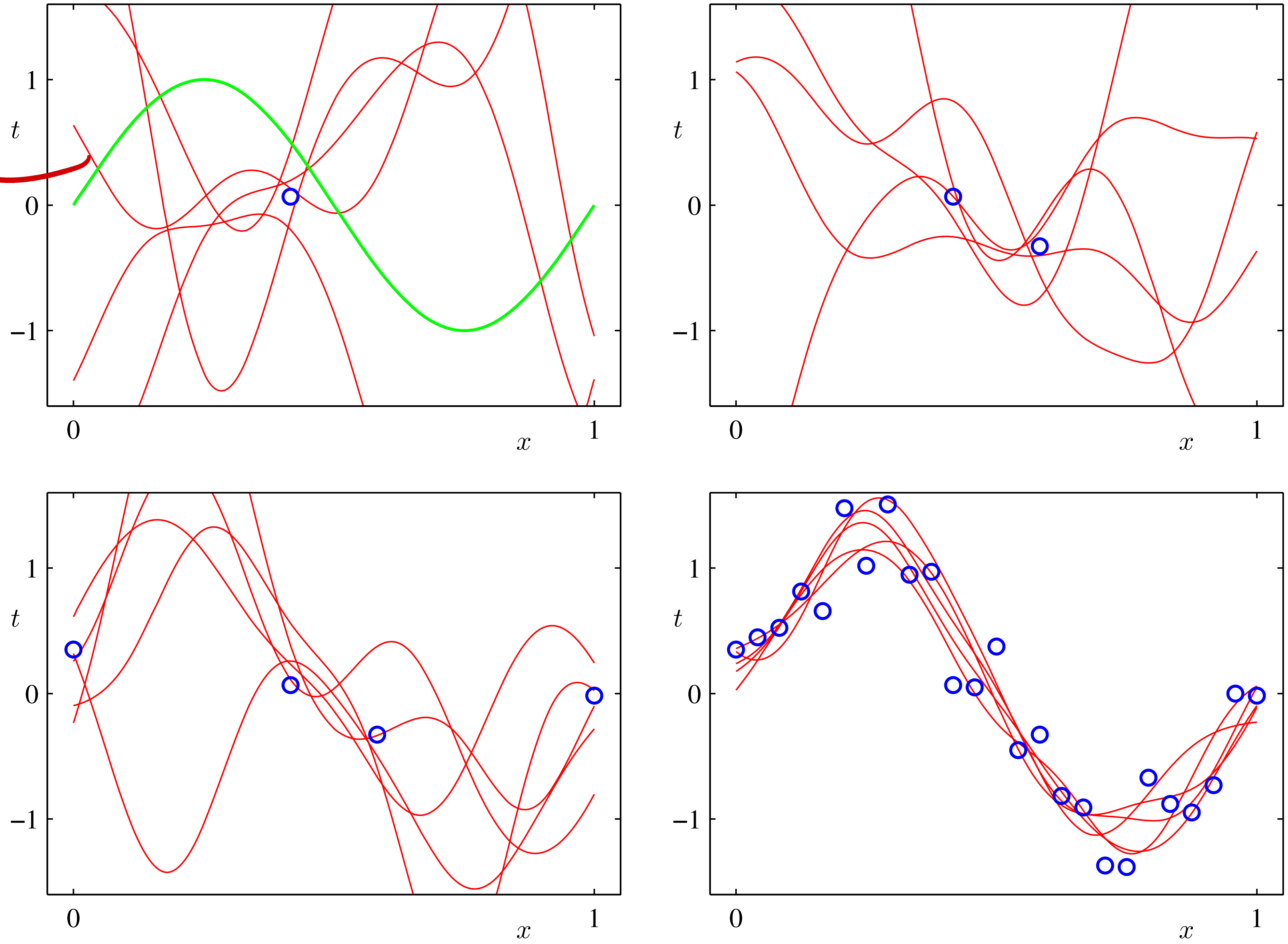
Figure: Predictive distribution (Bishop 3.8)

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

Samples drawn from Bayesian Predictive Distribution

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

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



$y(x, \underline{w})$

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