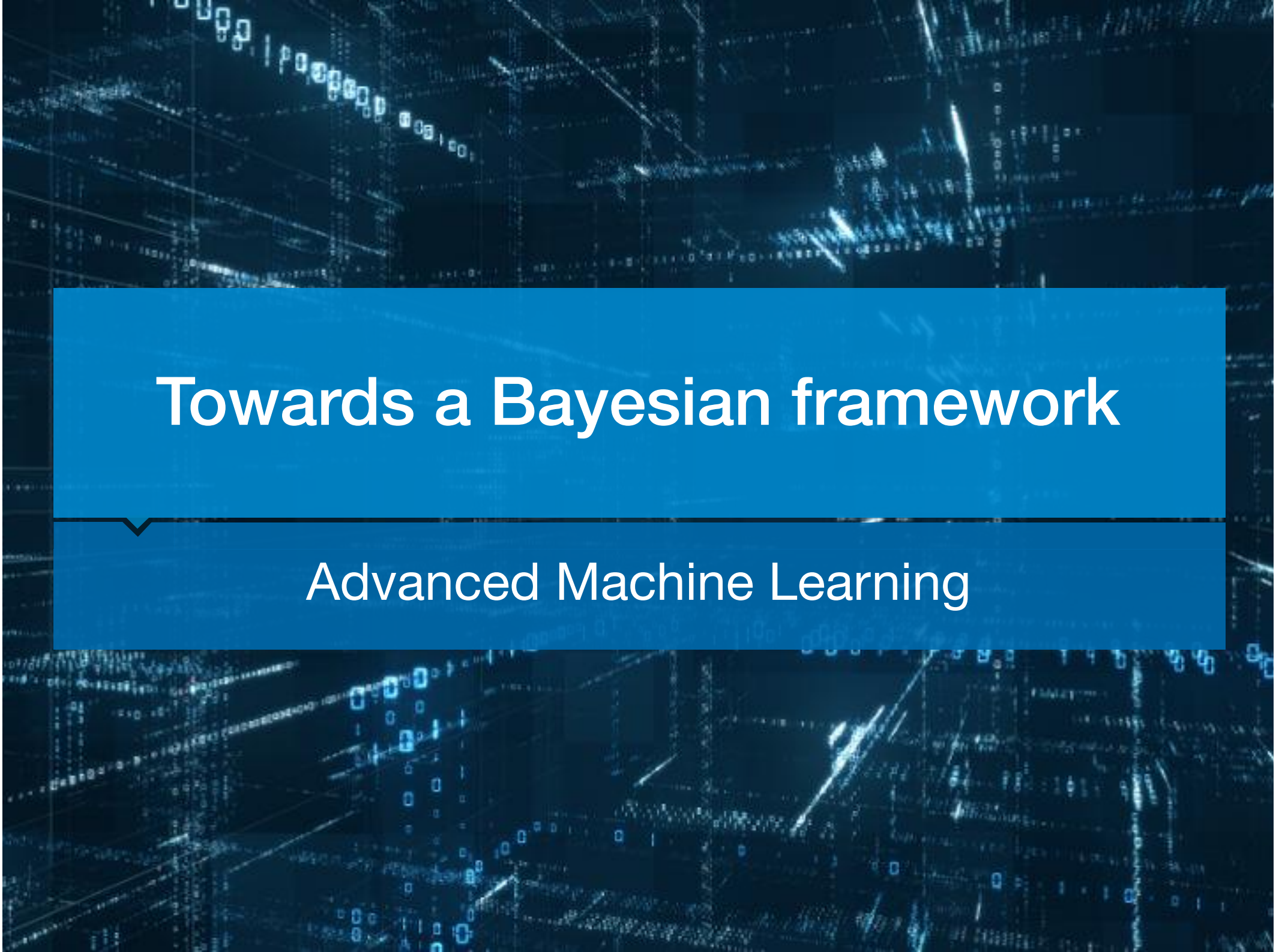
Advanced Machine Learning
Lecture 3: Linear models

Sandjai Bhulai
Vrije Universiteit Amsterdam

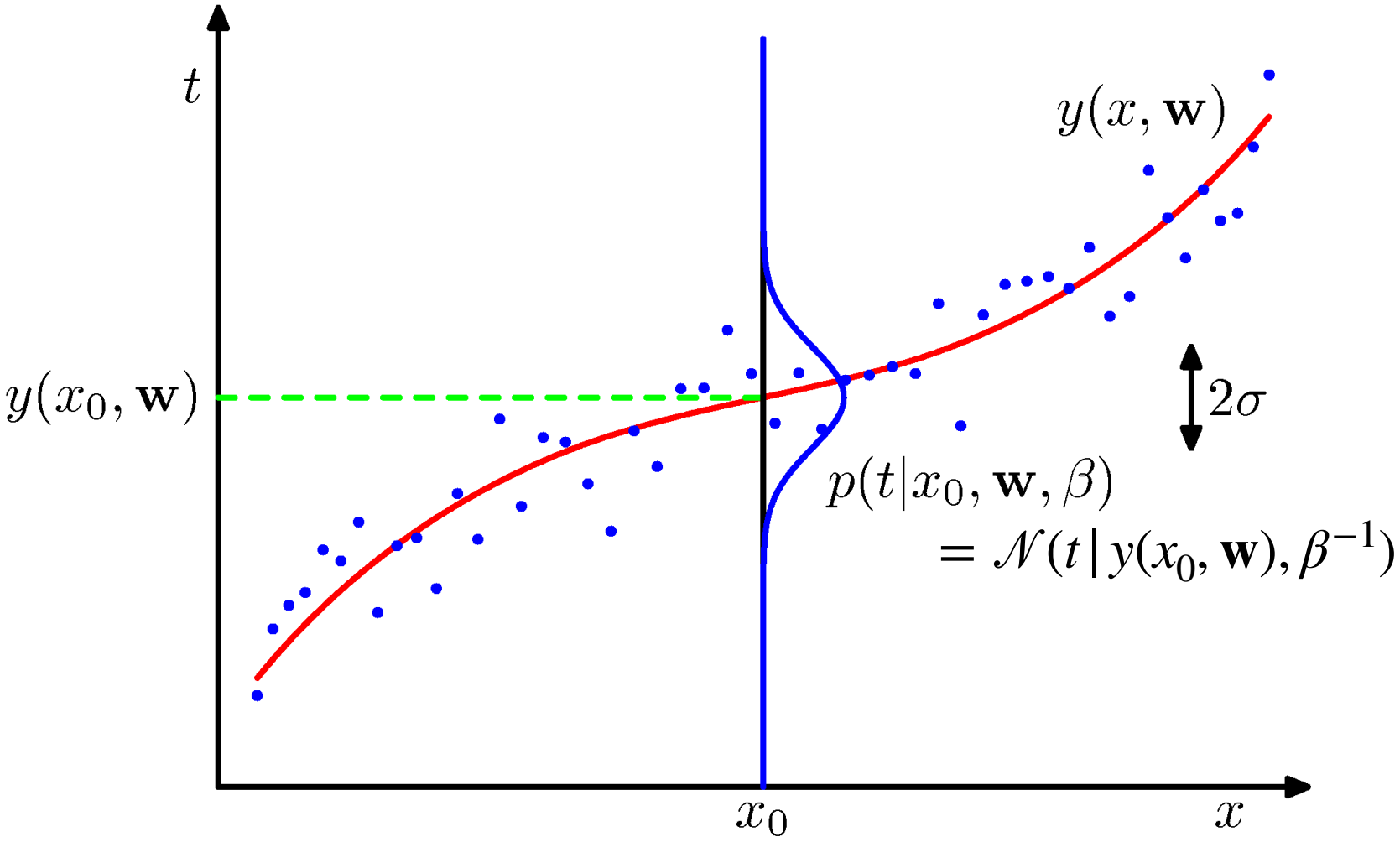
s.bhulai@vu.nl
12 September 2023



Towards a Bayesian framework

Advanced Machine Learning

The frequentist pitfall



The frequentist pitfall

- Given $p(t \mid \mathbf{x})$ or $p(t, \mathbf{x})$ directly minimize expected loss function

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

- Natural choice:

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

The frequentist pitfall

- Given a point x, the expected loss at that point is given by

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

- Taking the derivative w.r.t. $y(\mathbf{x})$ yields

$$2 \int \{y(\mathbf{x}) - t\} p(t \mid \mathbf{x}) \, \mathrm{d}t$$

- Setting this expression to 0, yields

$$y(\mathbf{x}) = \int y(\mathbf{x}) p(t \mid \mathbf{x}) \, \mathrm{d}t = \int t p(t \mid \mathbf{x}) \, \mathrm{d}t = \mathbb{E}_t[t \mid \mathbf{x}]$$

The frequentist pitfall

- The expected loss function

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

- Rewrite integrand as

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

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

- Taking the expected value yields

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

The frequentist pitfall

- Recall the expected square loss,

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

where $h(\mathbf{x}) = \mathbb{E}[t \,|\, x] = \int t p(t \,|\, \mathbf{x})\, \mathrm{d}t$

- The second term corresponds to the noise inherent in the random variable t
- What about the first term?

The frequentist pitfall

- Suppose we were given multiple datasets, each of size N. Any particular dataset $\mathcal{D}$, will give a particular function $y(\mathbf{x}; \mathcal{D})$

- We then have

$$\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2$$

$$= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2$$

$$= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2$$

$$+2\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}$$

The frequentist pitfall

- Taking the expectation over $\mathscr{D}$ yields:

$$\mathbb{E}_{\mathscr{D}}[\{y(\mathbf{x};\mathscr{D}) - h(\mathbf{x})\}^2]$$

$$= \underbrace{\{\mathbb{E}_{\mathscr{D}}[y(\mathbf{x};\mathscr{D})] - h(\mathbf{x})\}^2}_{\text{(bias)}^2} + \underbrace{\mathbb{E}_{\mathscr{D}}[\{y(\mathbf{x};\mathscr{D}) - \mathbb{E}_{\mathscr{D}}[y(\mathbf{x};\mathscr{D})]\}^2]}_{\text{variance}}$$

The frequentist pitfall

- In conclusion:

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

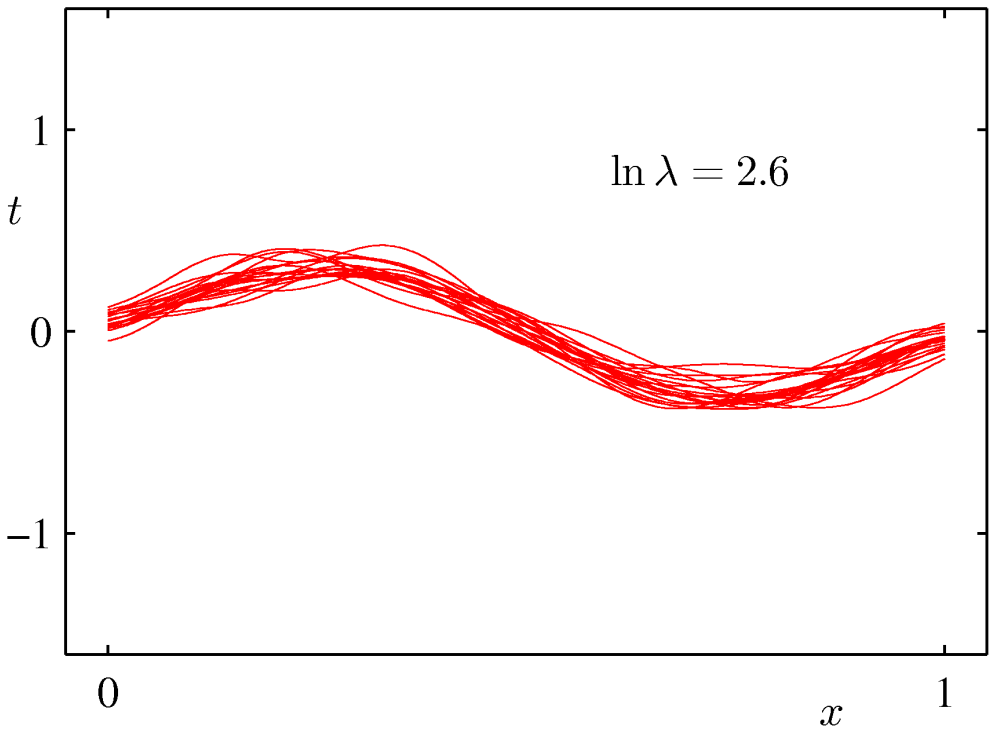
$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\, \mathrm{d}x$$

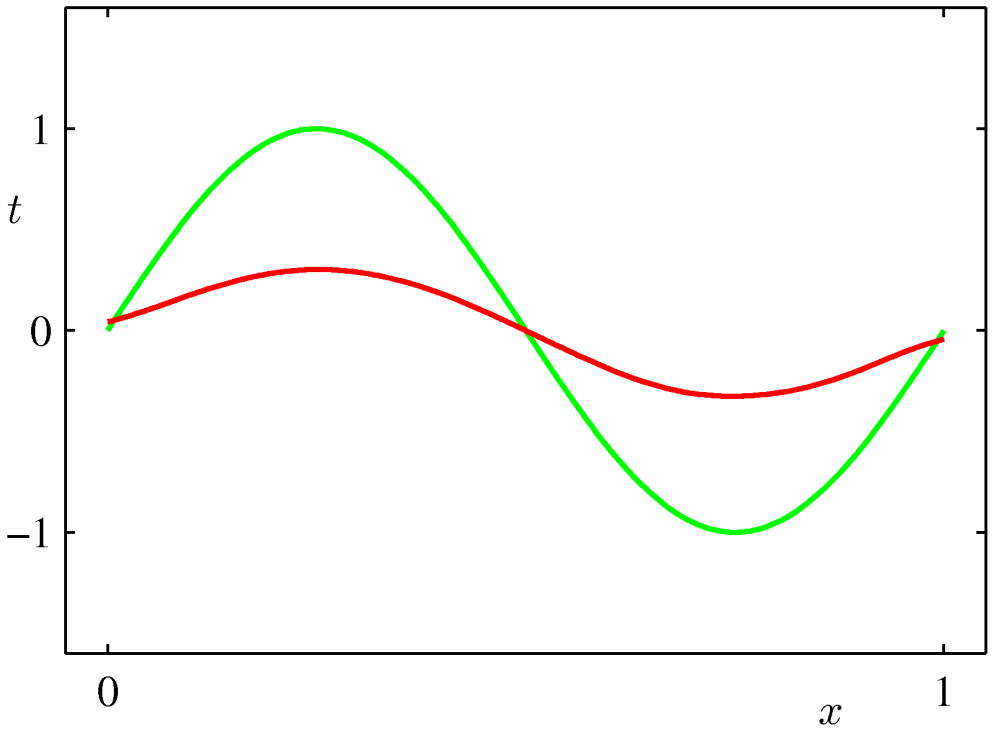
$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x};\mathcal{D})] - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2]\, p(\mathbf{x})\, \mathrm{d}x$$

$$\text{noise} = \int\int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t$$

Bias-variance decomposition

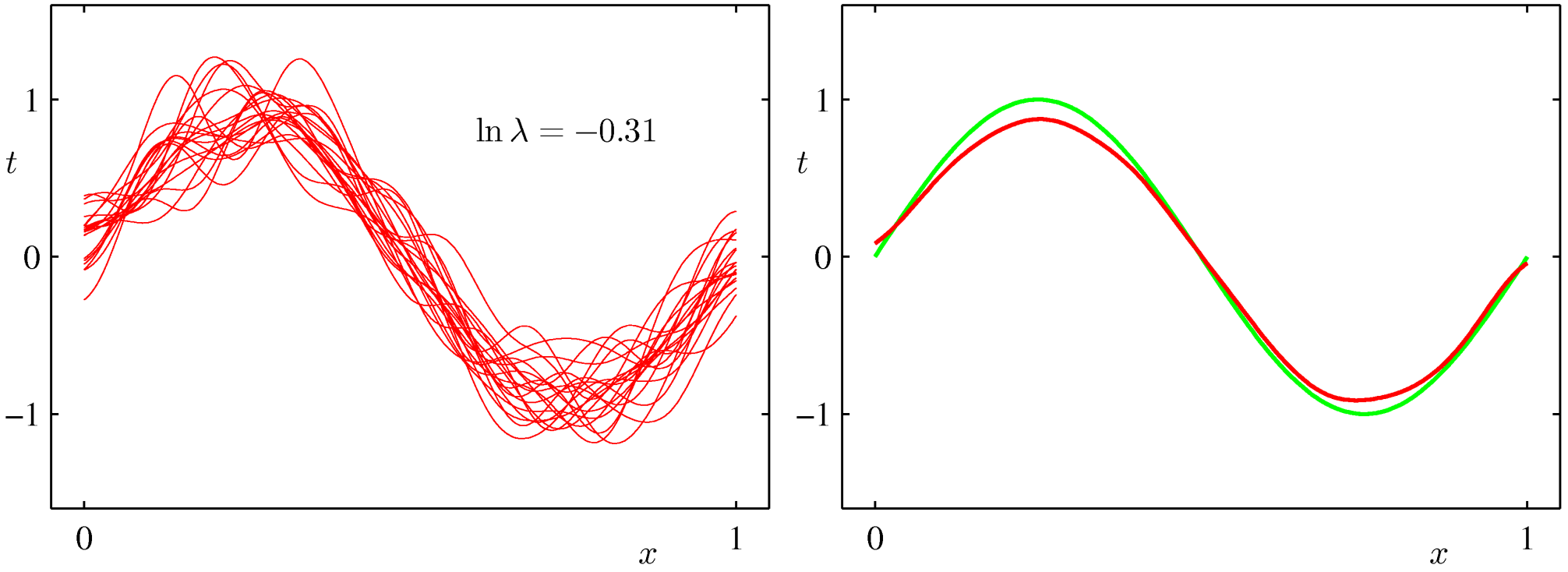
- Example: 100 datasets from the sinusoidal with 25 data points, varying the degree of regularization





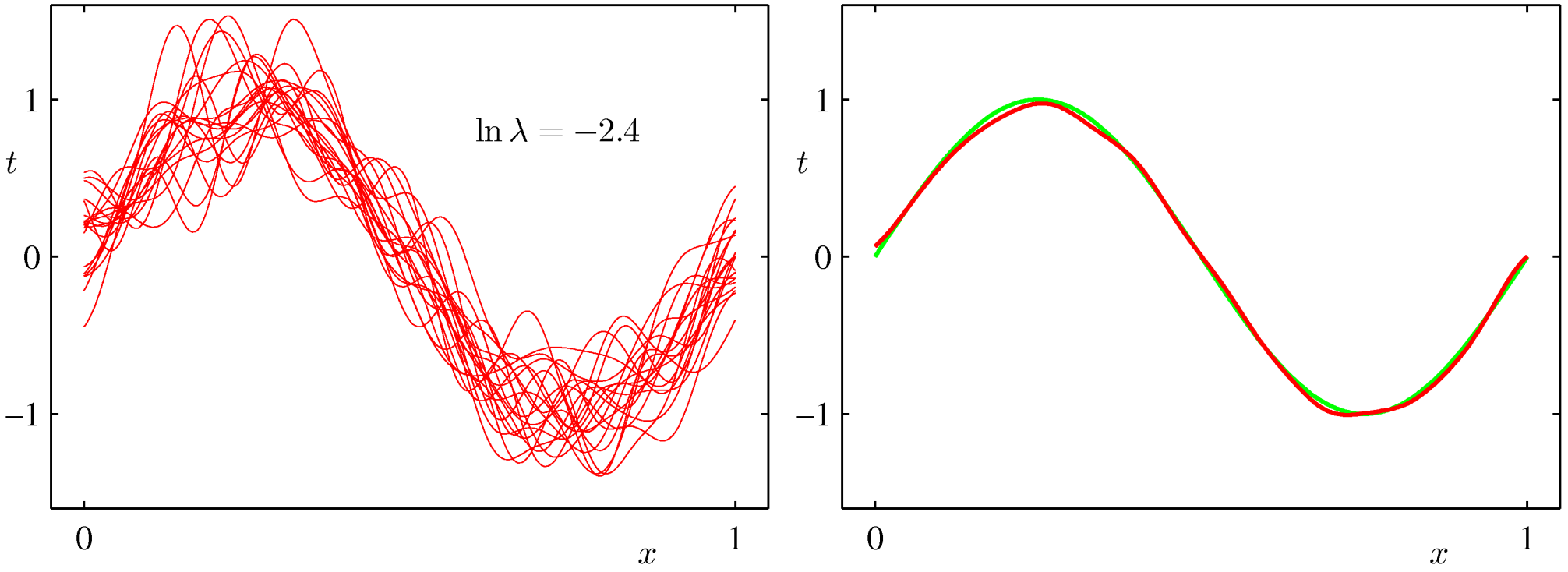
Bias-variance decomposition

- Example: 100 datasets from the sinusoidal with 25 data points, varying the degree of regularization



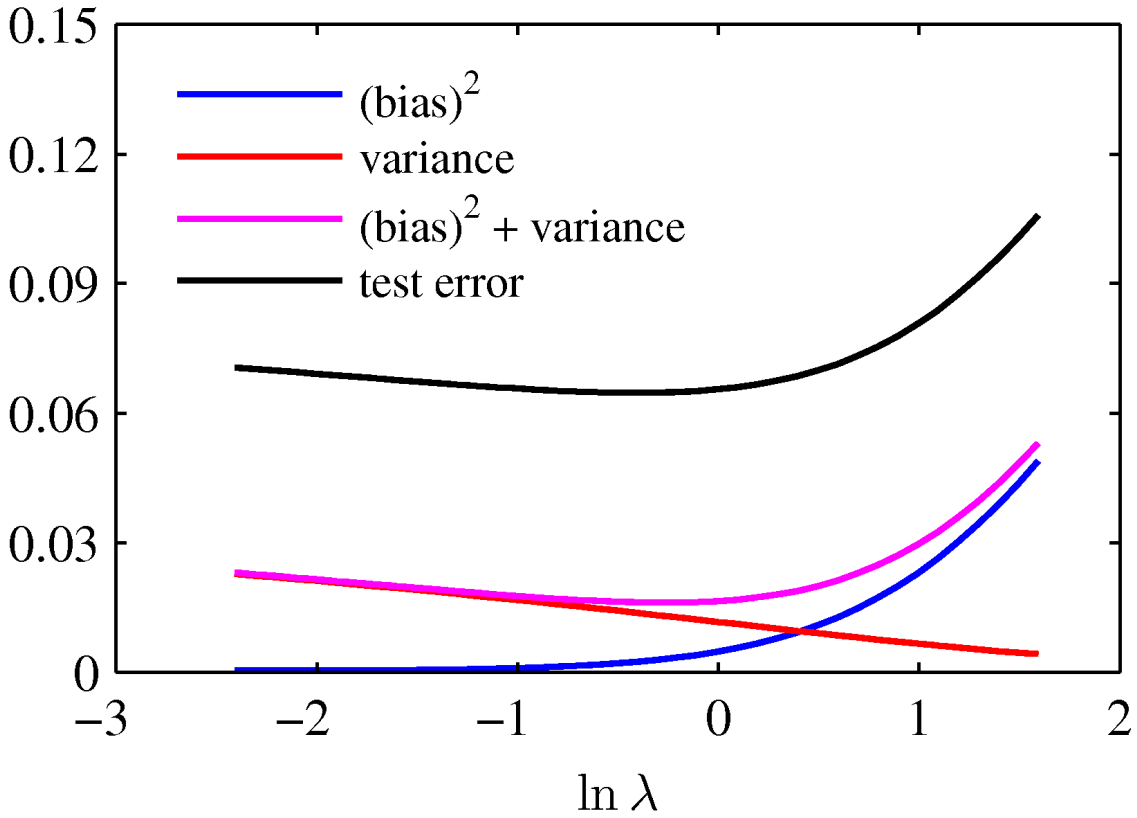
Bias-variance decomposition

- Example: 100 datasets from the sinusoidal with 25 data points, varying the degree of regularization



Bias-variance decomposition

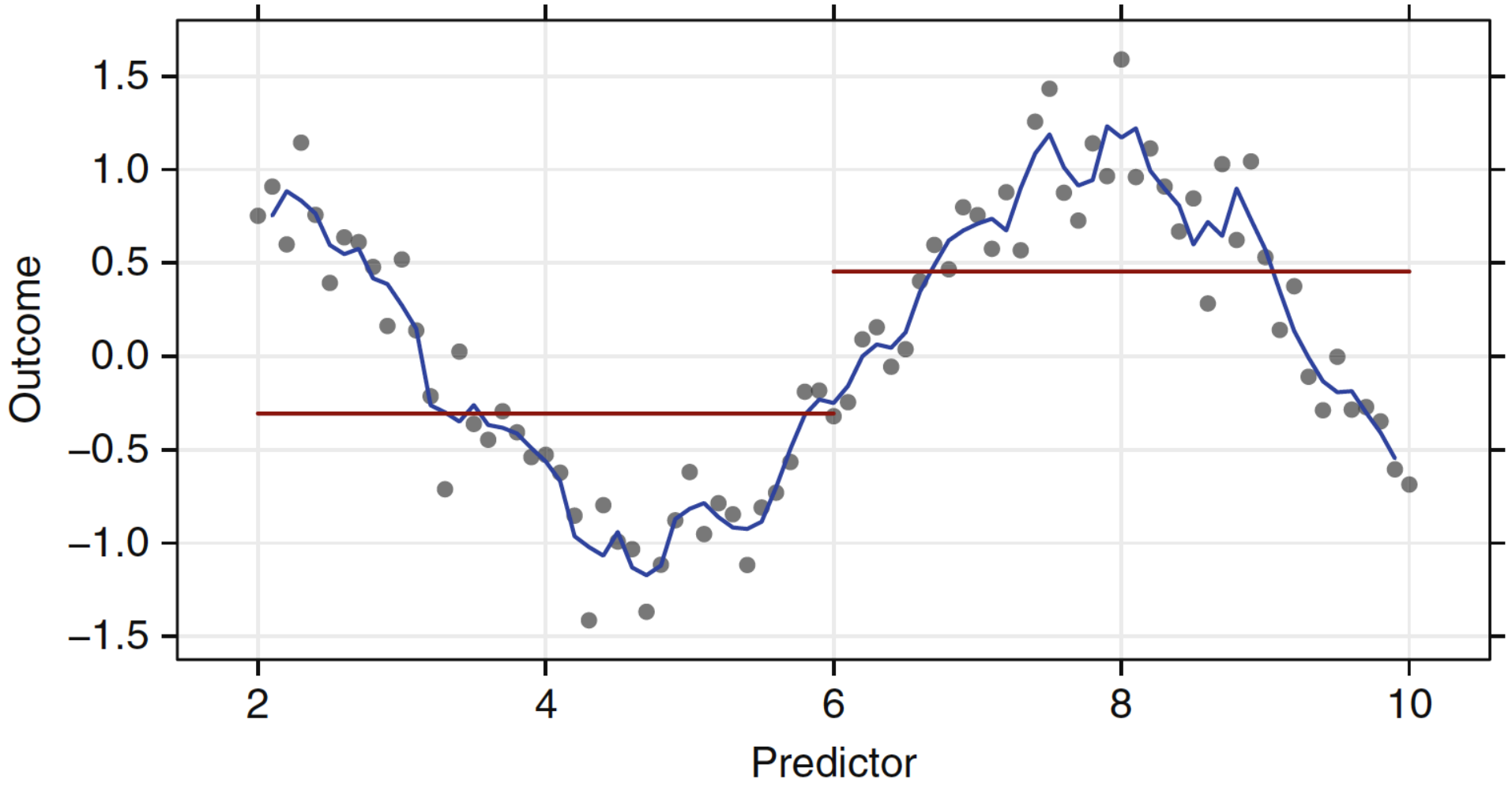
- From these plots, we note that an over-regularized model (large λ) will have a high bias, while an under-regularized model (small λ) will have a high variance



Bias-variance decomposition

- These insights are of limited practical value
- It is based on averages with respect to ensembles of datasets
- In practice, we have only the single observed dataset

Bias-variance tradeoff



Bayesian linear regression

- Bayes' theorem:

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

- Essentially, this leads to

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

- The idea is to use a probability distribution over the weights, and then update the weights based on the observed data

Bayesian linear regression

- Define a conjugate prior over $\mathbf{w}$

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \,|\, \mathbf{m}_0, \mathbf{S}_0)$$

- Combining this with the likelihood function and using results for marginal and conditional Gaussian distributions, gives the posterior

$$p(\mathbf{w} \,|\, \mathbf{t}) = \mathcal{N}(\mathbf{w} \,|\, \mathbf{m}_N, \mathbf{S}_N)$$

where

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \mathbf{\Phi}^\top \mathbf{t} \right)$$

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \mathbf{\Phi}^\top \mathbf{\Phi}$$

Bayesian linear regression

- A common choice for the prior is

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

for which

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

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

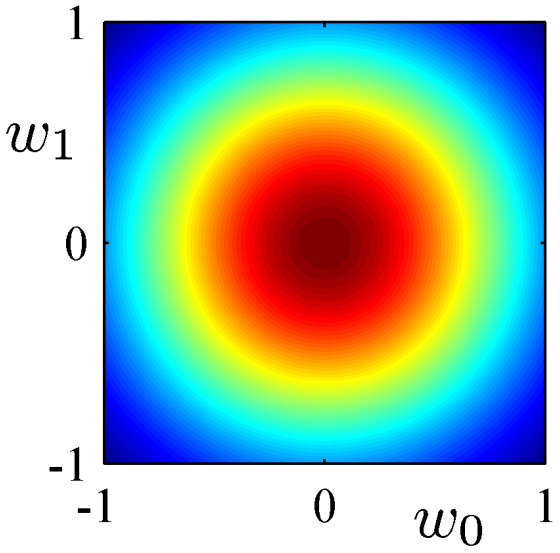
- Consider the following example to make the concept less abstract: $y(x) = -0.3 + 0.5x$

Bayesian linear regression

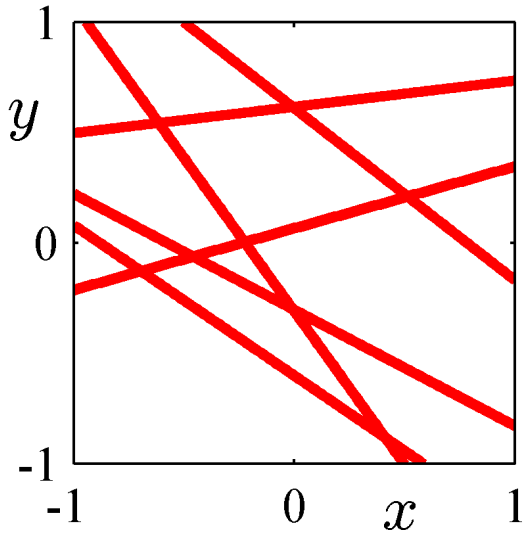
- 0 data points observed

likelihood

prior/posterior



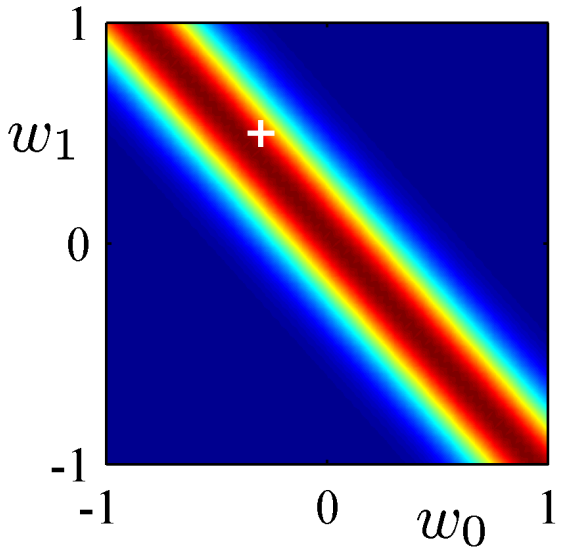
data space



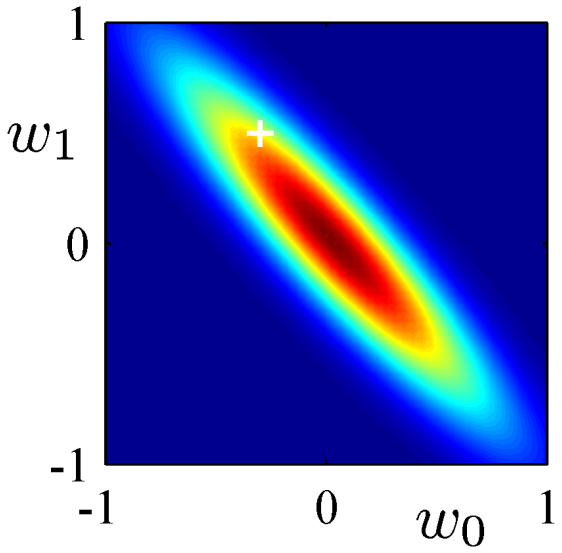
Bayesian linear regression

- 1 data point observed

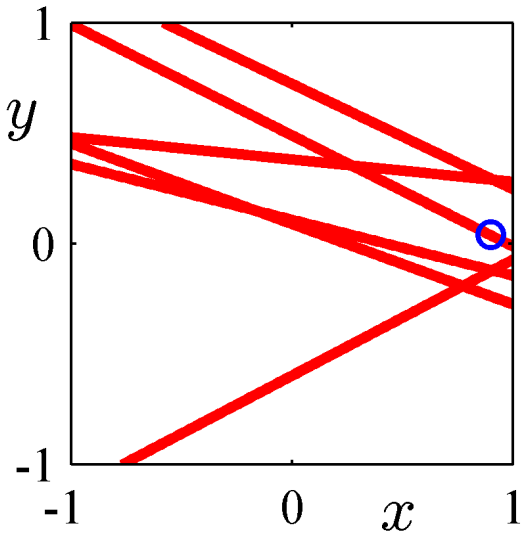
likelihood



prior/posterior

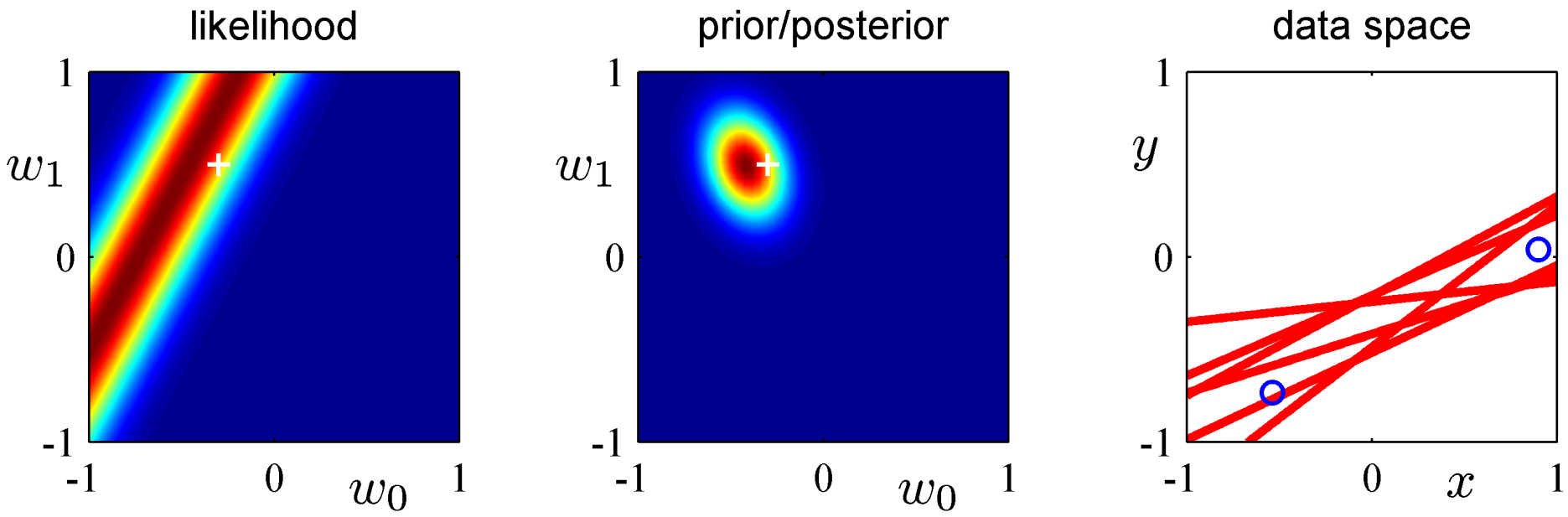


data space



Bayesian linear regression

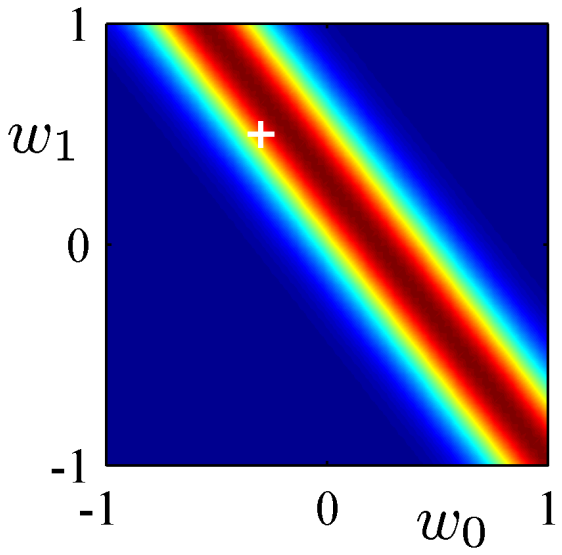
- 2 data points observed



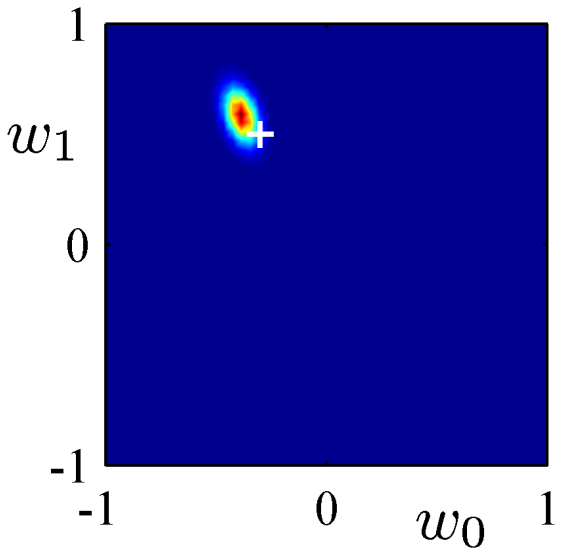
Bayesian linear regression

- 20 data points observed

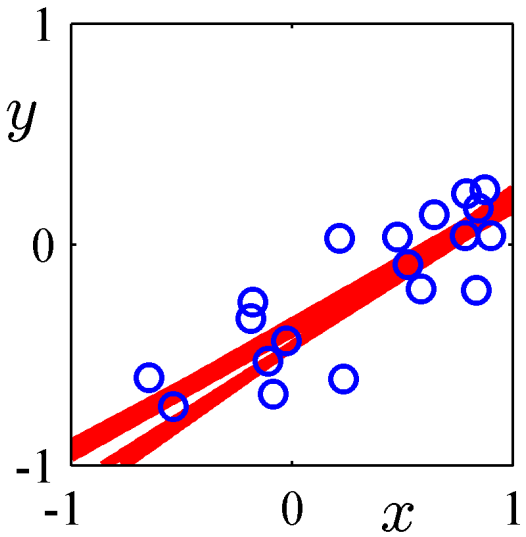
likelihood



prior/posterior



data space



Bayesian linear regression

- A common choice for the prior is

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

for which

$$\mathbf{m}_N = \beta\, \mathbf{S}_N \boldsymbol{\Phi}^\top \mathbf{t}$$

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

- What is the log of the posterior distribution, i.e., $\ln p(\mathbf{w} \,|\, \mathbf{t})$?

$$\ln p(\mathbf{w} \,|\, \mathbf{t}) = \ln \frac{p(\mathbf{t} \,|\, \mathbf{w}) p(\mathbf{w})}{p(\mathbf{t})} = \frac{\beta}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^T \varphi(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^\top\mathbf{w} + \text{const}$$

Bayesian linear regression

- A common choice for the prior is

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

for which

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

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

- What if we have no prior no information, i.e., $\alpha \to 0$?

$$\mathbf{m}_N = (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{t}$$

Bayesian linear regression

- A common choice for the prior is

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

for which

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

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

- What if we have precise prior information, i.e., $\alpha \to \infty$?

$$\mathbf{m}_N = 0$$

Bayesian linear regression

- A common choice for the prior is

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

for which

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

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

- What if we have infinite data, i.e., $N \to \infty$?

$$\lim_{N\to\infty} \mathbf{m}_N = (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{t}$$

Bayesian linear regression

- Predict t for new values of x by integrating over $\mathbf{w}$

$$p(t \mid \mathbf{t}, \alpha, \beta) = \int p(t \mid \mathbf{w}, \beta) p(\mathbf{w} \mid \mathbf{t}, \alpha, \beta) \, \mathrm{d}\mathbf{w}$$

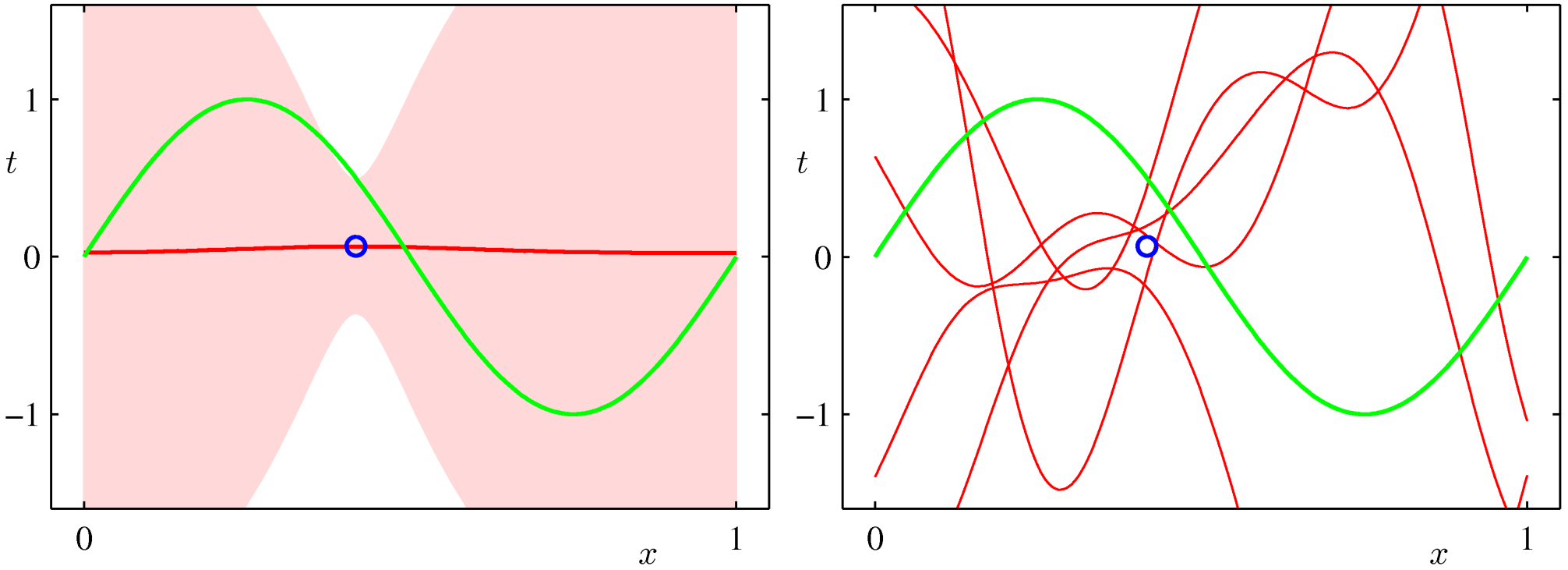
$$= \mathcal{N}(t \mid \mathbf{m}_N^\top \varphi(\mathbf{x}), \sigma_N^2(\mathbf{x}))$$

where

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \varphi(\mathbf{x})^\top \mathbf{S}_N \varphi(\mathbf{x})$$

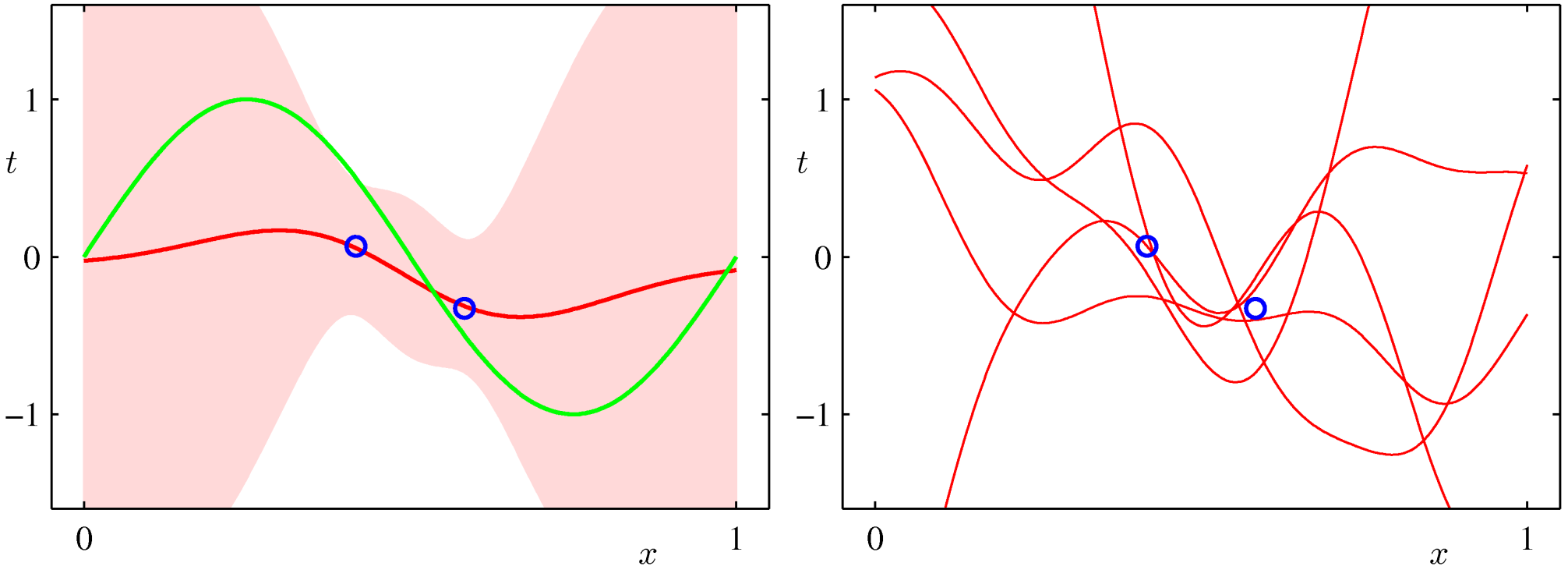
Bayesian linear regression

- Sinusoidal data, 9 Gaussian basis functions: 1 data point



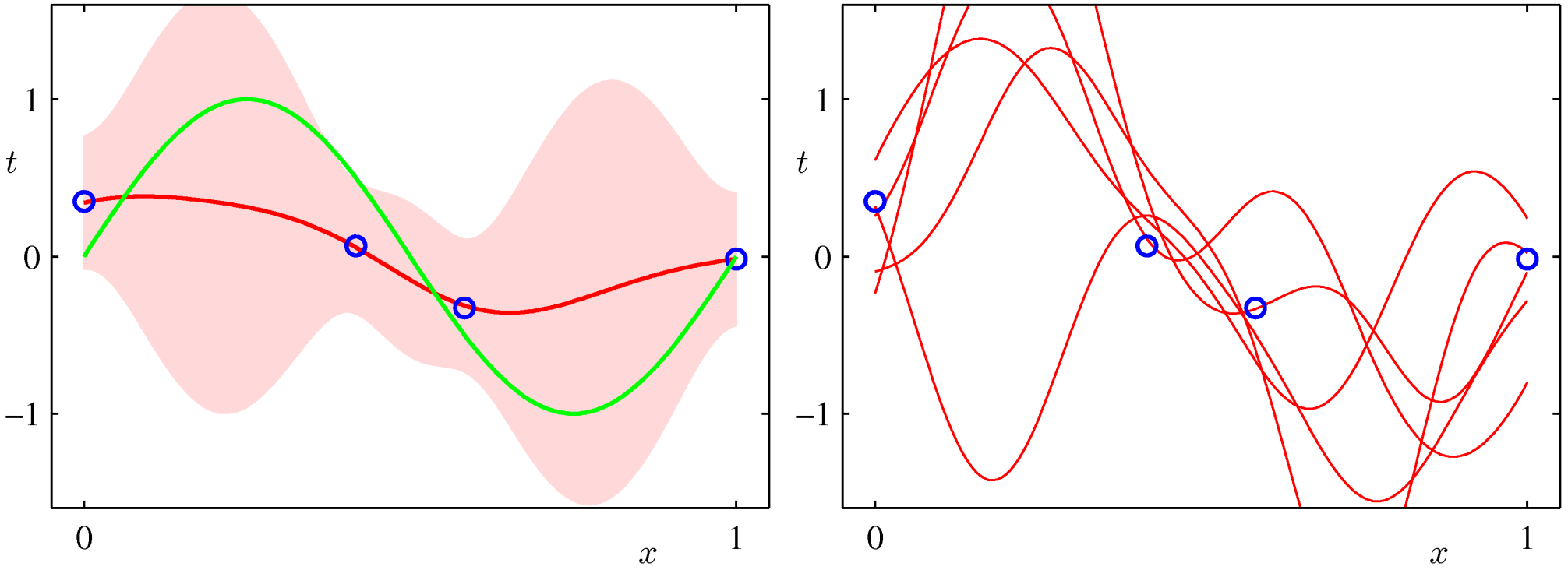
Bayesian linear regression

- Sinusoidal data, 9 Gaussian basis functions: 2 data points



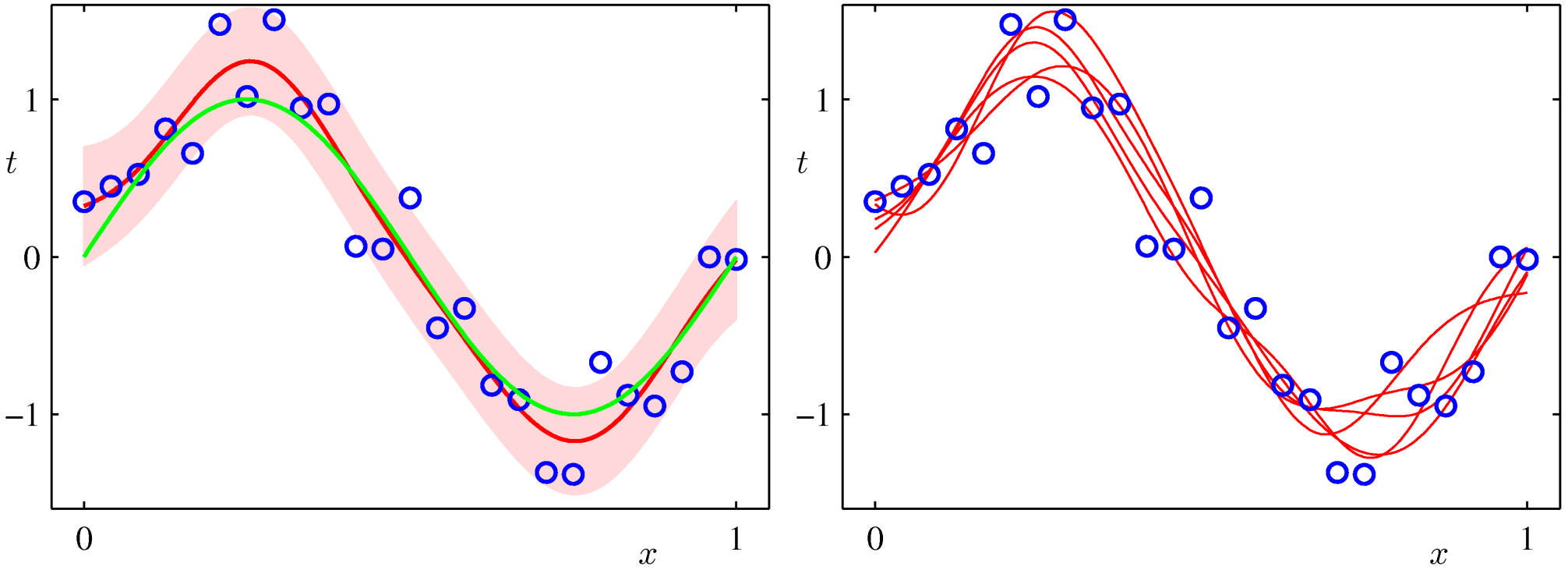
Bayesian linear regression

- Sinusoidal data, 9 Gaussian basis functions: 4 data points



Bayesian linear regression

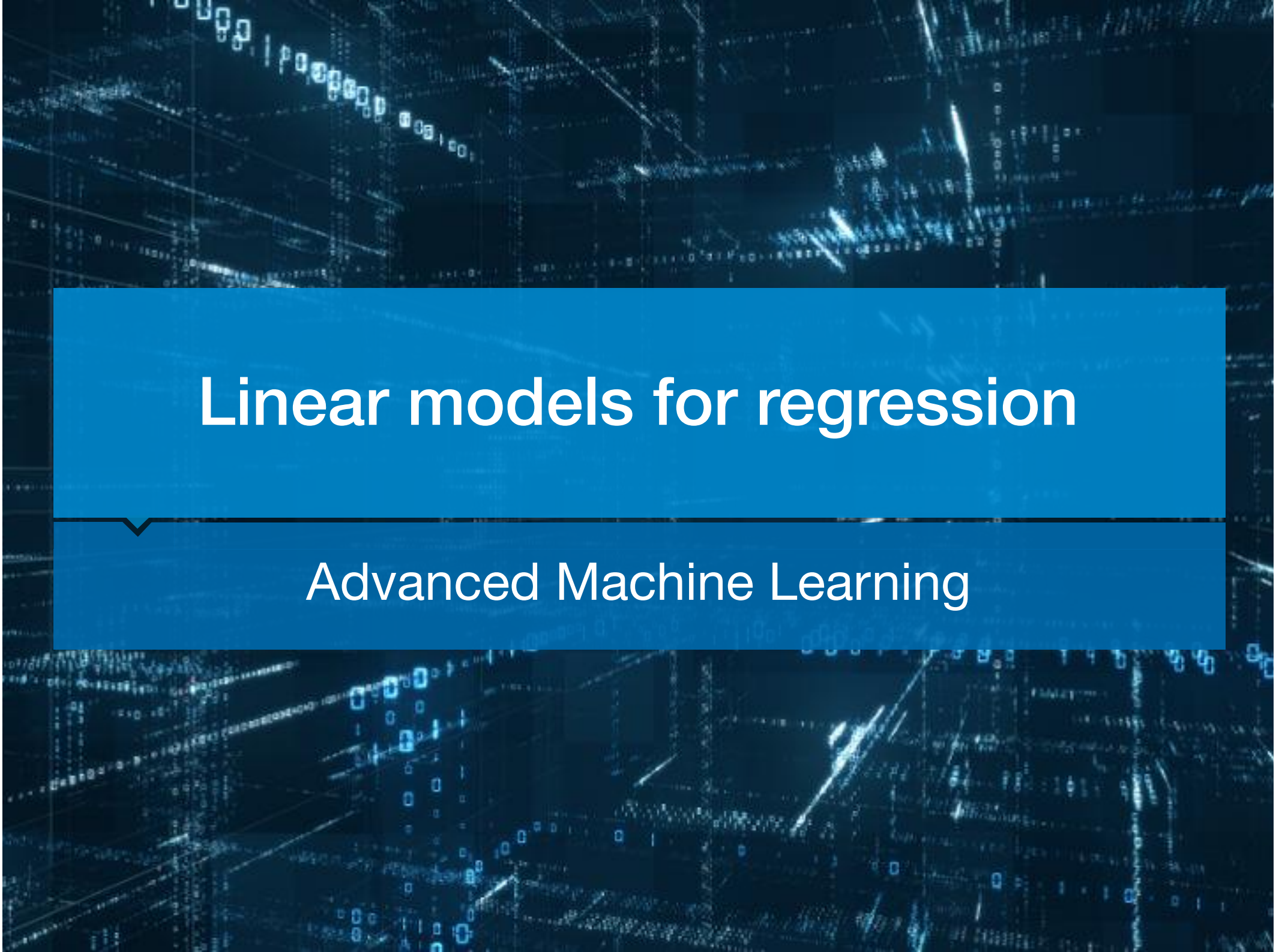
- Sinusoidal data, 9 Gaussian basis functions: 25 data points



Conclusions

- The use of maximum likelihood, or equivalently, least squares, can lead to severe overfitting if complex models are trained using data sets of limited size

- A Bayesian approach to machine learning avoids the overfitting and also quantifies the uncertainty in model parameters

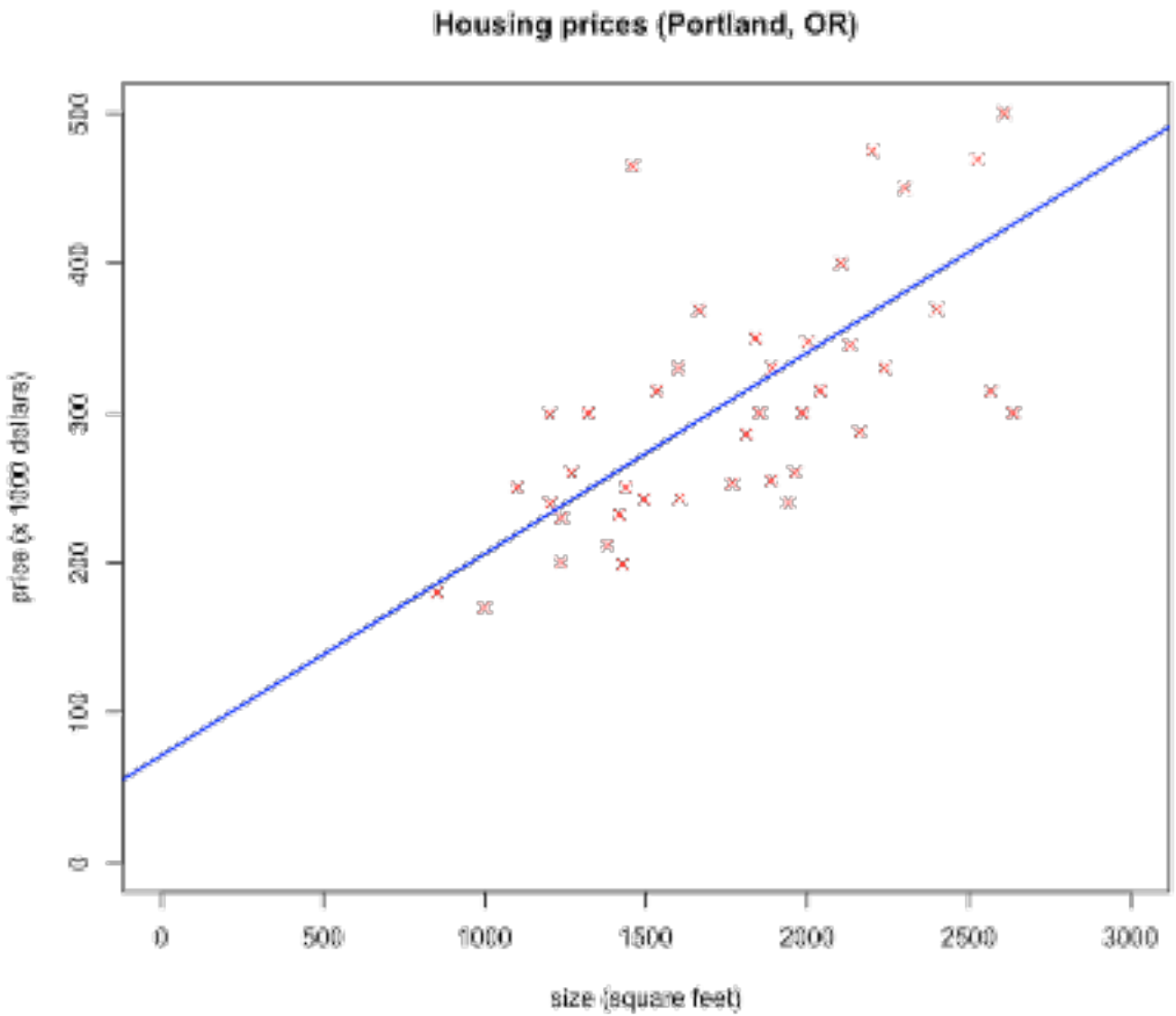


Linear models for regression

Advanced Machine Learning

Linear regression

Linear regression



Linear regression

- General model is:

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \varphi_j(\mathbf{x}) = \mathbf{w}^\top \varphi(\mathbf{x})$$

- $\varphi_j = x^j$
- Take $M = 2$
- Calculate $\mathbf{w}_{\text{ML}} = \left(\Phi^\top \Phi\right)^{-1} \Phi^\top \mathbf{t}$

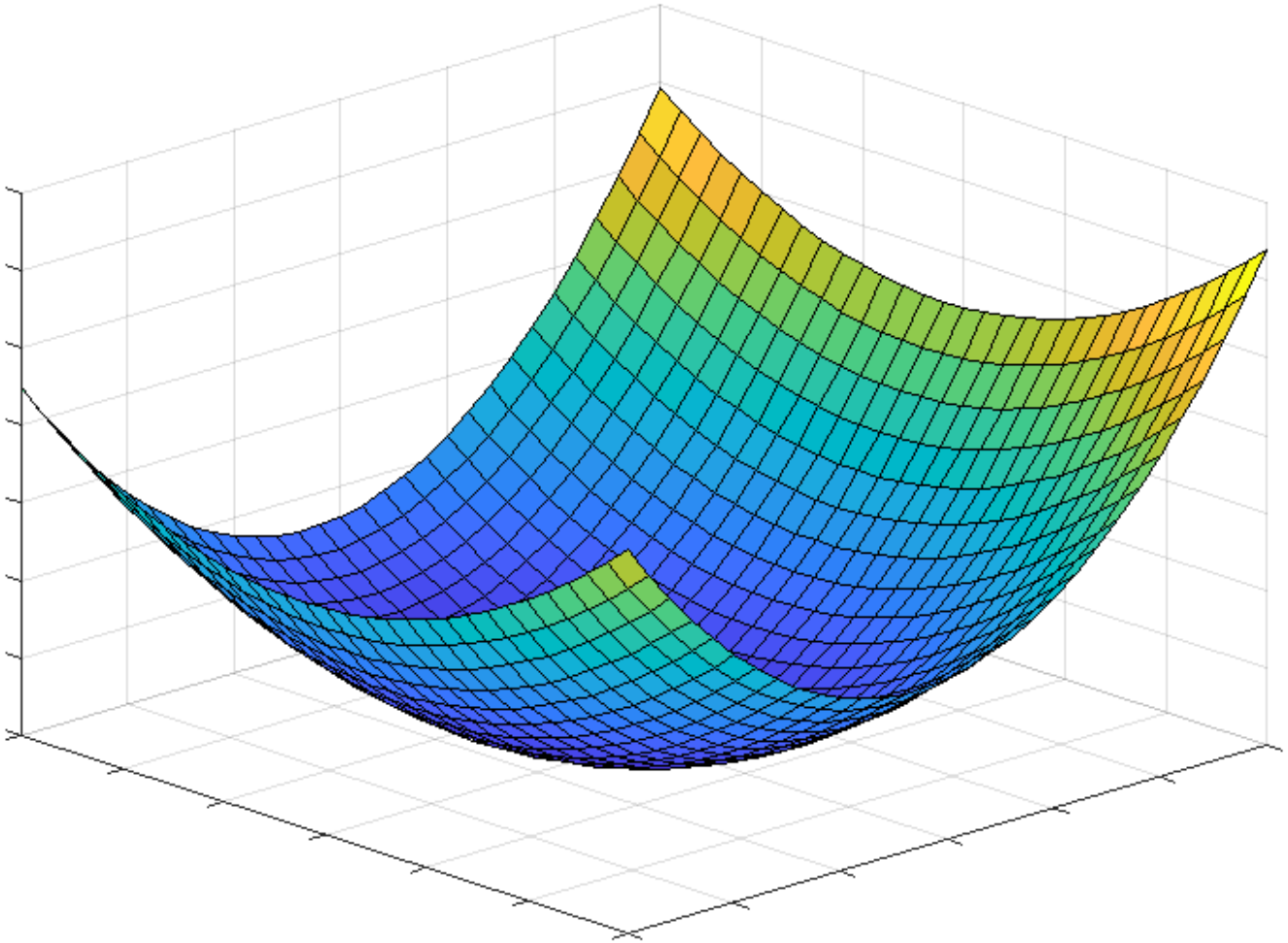
Linear regression

- Thus, we have $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x$
- Performance is measured by

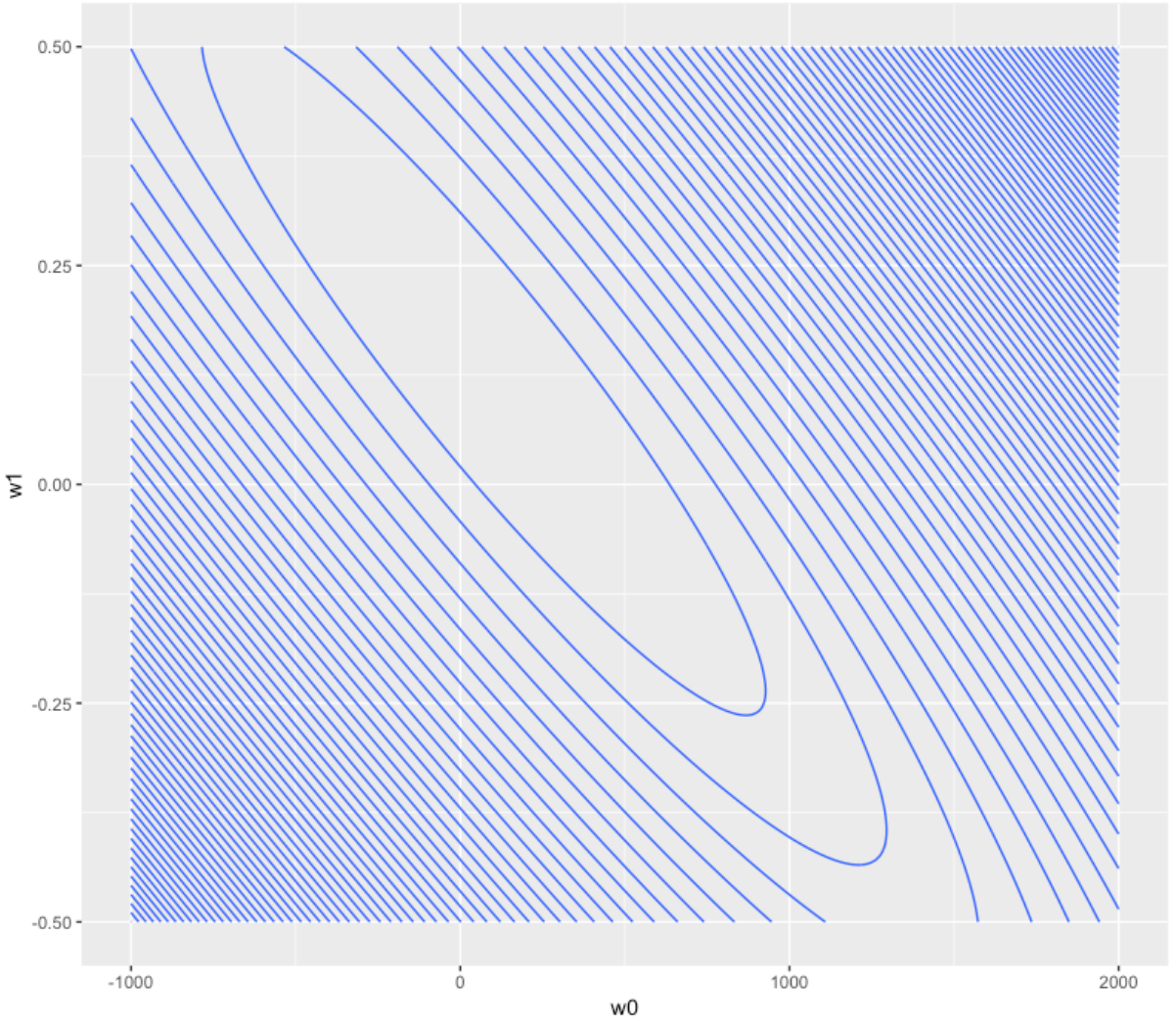
$$E(\mathbf{w}) = \frac{1}{2N} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

- Goal: $\min_{w_0, w_1} E(w_0, w_1)$

Gradient descent



Gradient descent



Gradient descent

- Gradient descent algorithm:

repeat until convergence {

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} E(w_0, w_1)$$

}

Gradient descent

- Correct update:

$$\text{temp0} := w_0 - \alpha \frac{\partial}{\partial w_0} E(w_0, w_1)$$

$$\text{temp1} := w_1 - \alpha \frac{\partial}{\partial w_1} E(w_0, w_1)$$

$$w_0 := \text{temp0}$$

$$w_1 := \text{temp1}$$

Gradient descent

- Incorrect update:

$$\text{temp0} := w_0 - \alpha \frac{\partial}{\partial w_0} E(w_0, w_1)$$

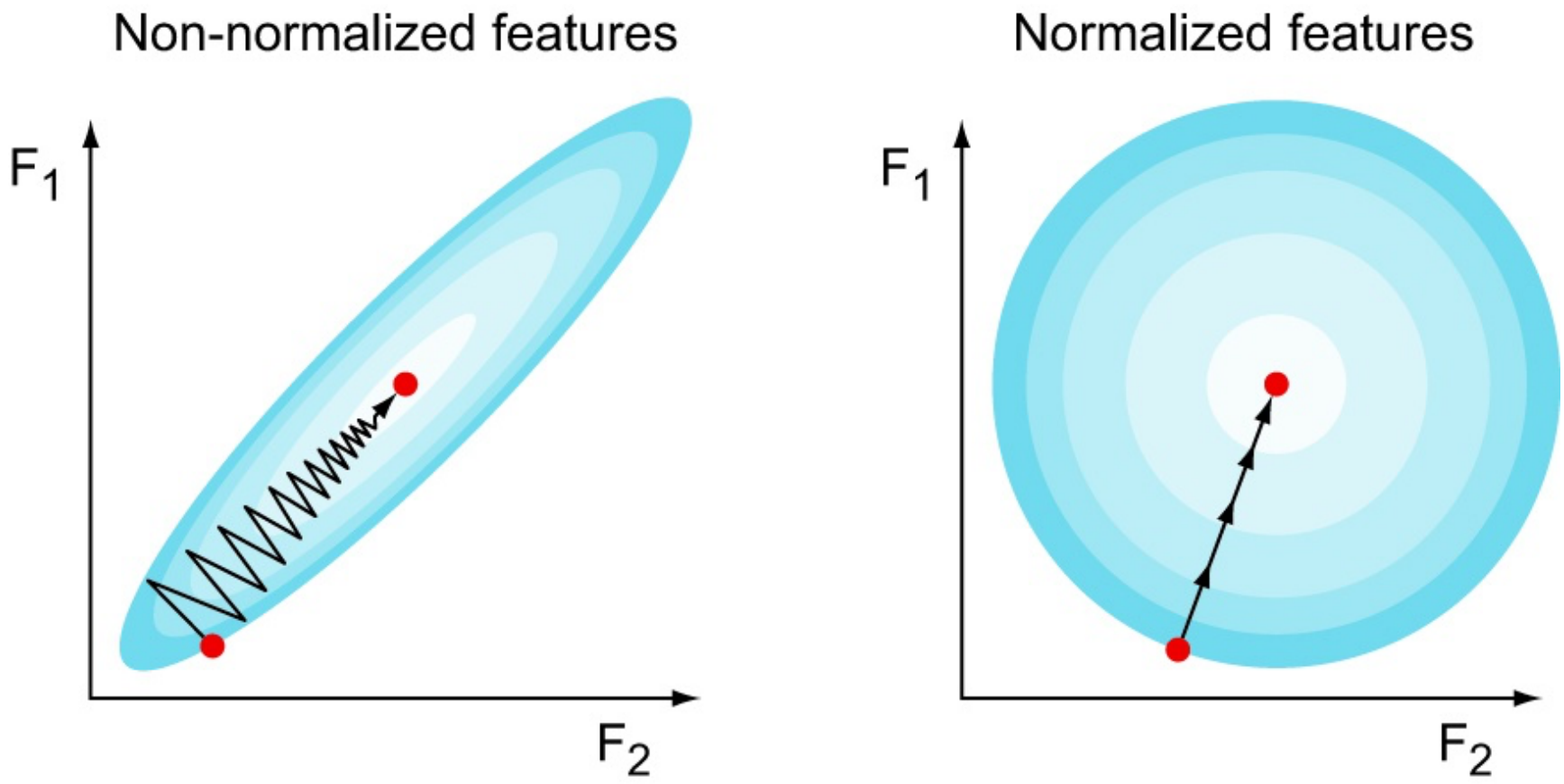
$$w_0 := \text{temp0}$$

$$\text{temp1} := w_1 - \alpha \frac{\partial}{\partial w_1} E(w_0, w_1)$$

$$w_1 := \text{temp1}$$

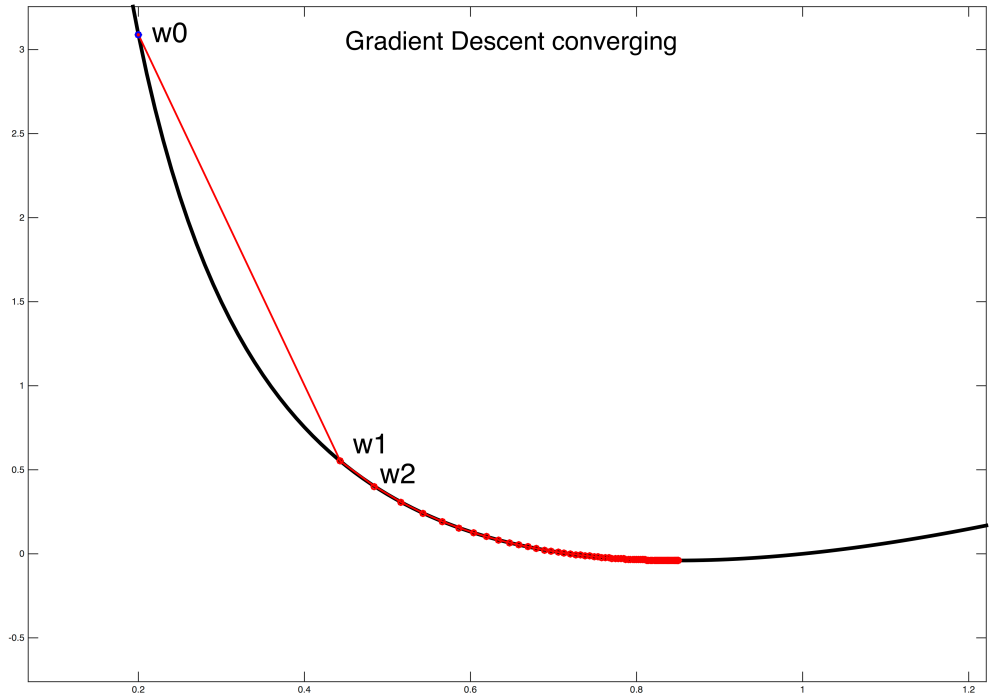
Gradient descent

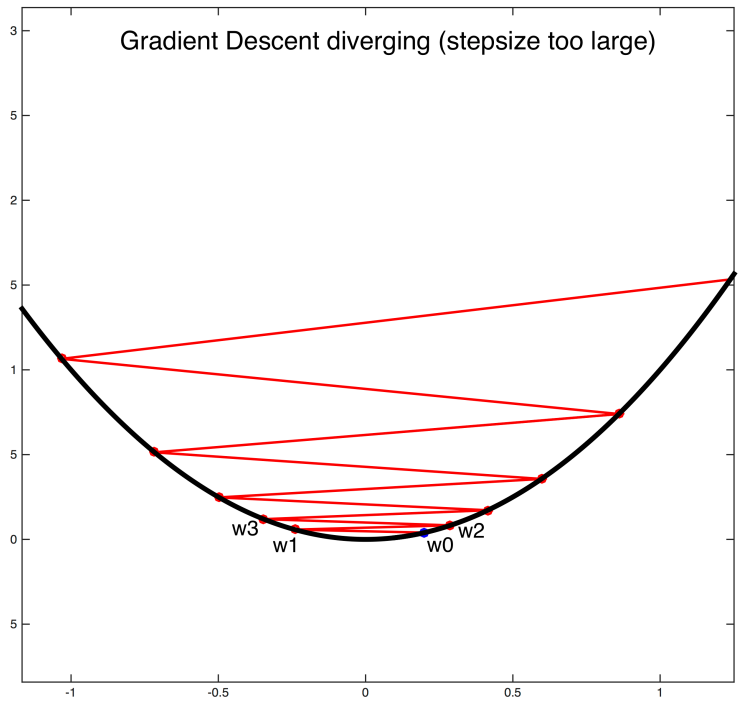
- Feature scaling is important



Gradient descent

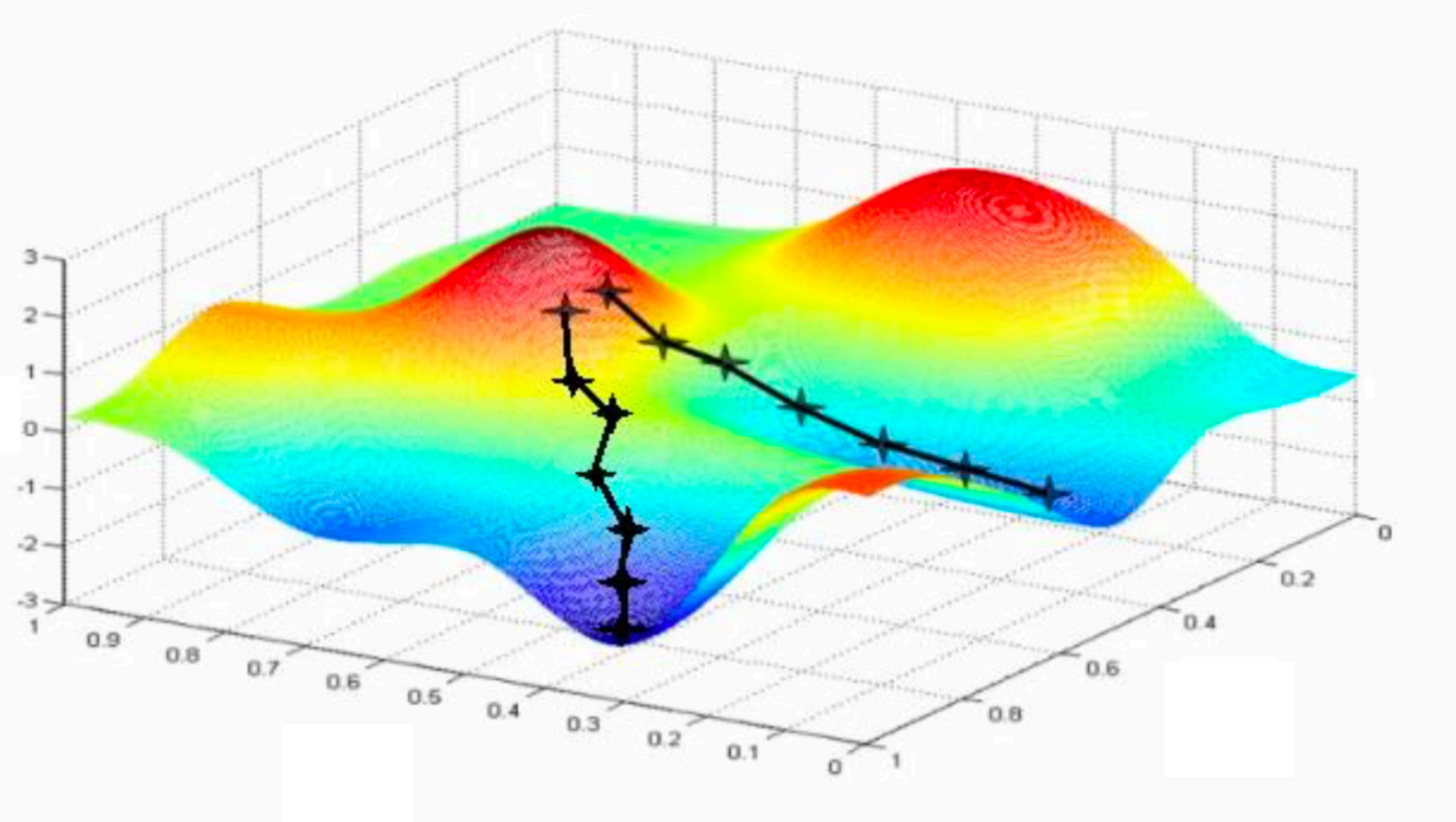
- Step size is important for convergence





Gradient descent

- Convexity of the problem is important for global optimality



Gradient descent for linear regression

- General model is:

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \varphi_j(\mathbf{x}) = \mathbf{w}^\top \varphi(\mathbf{x})$$

- Repeat {

$$w_j := w_j - \alpha \frac{1}{N} \sum_{n=1}^{N} \left(y(x_n, \mathbf{w}) - t_n \right) \varphi_j(x_n)$$

}