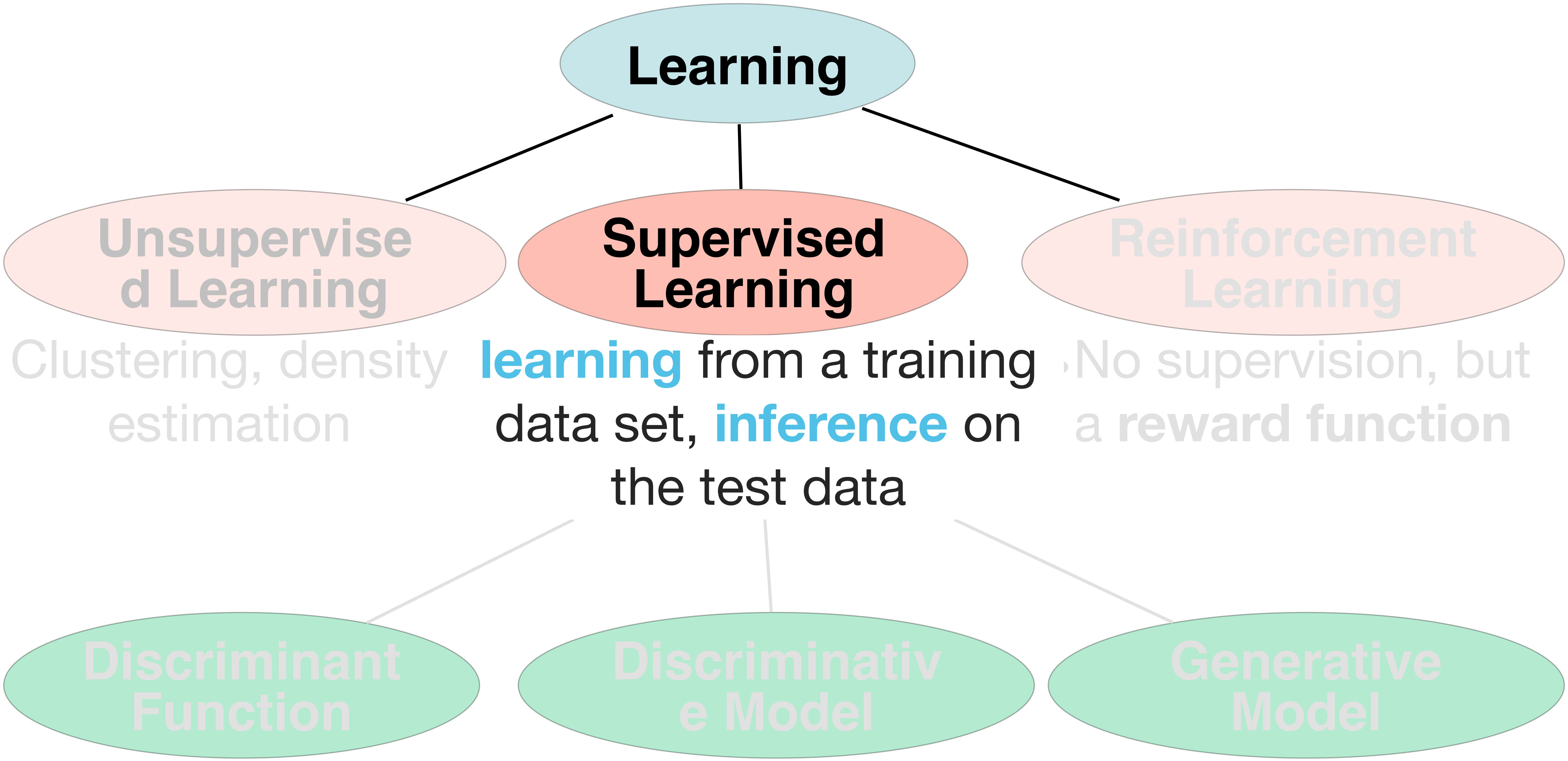
3. Regression

Categories of Learning (Rep.)



Categories of Learning

Learning

- **Unsupervised Learning**
  clustering, density estimation
- **Supervised Learning**
  learning from a training data set, **inference** on the test data
  - **Discriminant Function**
    no prob. formulation, learns a **function** from objects $\mathcal{X}$ to labels $\mathcal{Y}$.
  - **Discriminative Model**
    estimates the **posterior** $p(y_k \mid \mathbf{x})$ for each class
  - **Generative Model**
    est. the **likelihoods** $p(\mathbf{x} \mid y_k)$ and use Bayes rule for the post.
- **Reinforcement Learning**
  no supervision, but a **reward function**

Mathematical Formulation (Rep.)

Suppose we are given a set $\mathcal{X}$ of objects and a set $\mathcal{Y}$ of object categories (classes). In the learning task we search for a mapping $\varphi : \mathcal{X} \rightarrow \mathcal{Y}$ such that ***similar*** elements in $\mathcal{X}$ are mapped to ***similar*** elements in $\mathcal{Y}$.

Difference between regression and classification:

- In regression, $\mathcal{Y}$ is ***continuous***, in classification it is discrete
- Regression learns a **function**, classification usually learns **class labels**

For now we will treat regression

Basis Functions

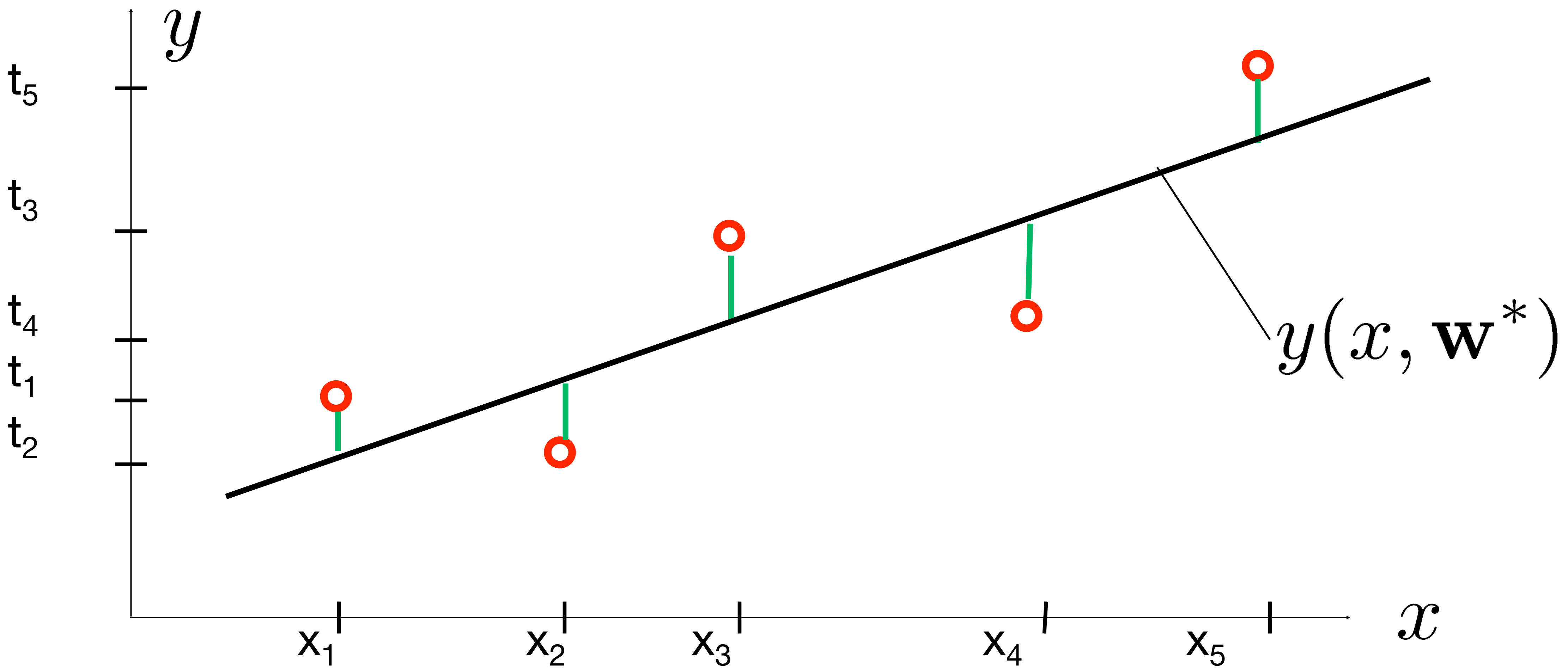
In principal, the elements of $\mathcal{X}$ can be anything (e.g. real numbers, graphs, 3D objects). To be able to treat these objects mathematically we need functions ϕ that map from $\mathcal{X}$ to $\mathbb{R}^N$. We call these the **basis functions**.

We can also interpret the basis functions as functions that extract **features** from the input data.

Features reflect the **properties** of the objects (width, height, etc.).

Simple Example: Linear Regression

- Assume: $\mathcal{X} = \mathbb{R}, \ \mathcal{Y} = \mathbb{R}, \ \phi = I$ (identity)
- **Given:** data points $(x_1, t_1), (x_2, t_2), \ldots$
- **Goal:** predict the value t of a new example x
- Parametric formulation: $y(x, \mathbf{w}) = w_0 + w_1 x$



Linear Regression

To evaluate the function y, we need an error function:

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

"Sum of Squared Errors"

We search for parameters $\mathbf{w}^*$ s.th. $E(\mathbf{w}^*)$ is minimal:

$$\nabla E(\mathbf{w}) = \sum_{i=1}^{N}(y(x_i, \mathbf{w}) - t_i)\nabla y(x_i, \mathbf{w}) \doteq (0 \quad 0)$$

$$y(x_i, \mathbf{w}) = w_0 + w_1 x_i \qquad \Rightarrow \qquad \nabla y(x_i, \mathbf{w}) = (1 \quad x_i)$$

Using vector notation: $\mathbf{x}_i := (1 \quad x_i)^T \qquad y(x_i, \mathbf{w}) = \mathbf{w}^T \mathbf{x}_i$

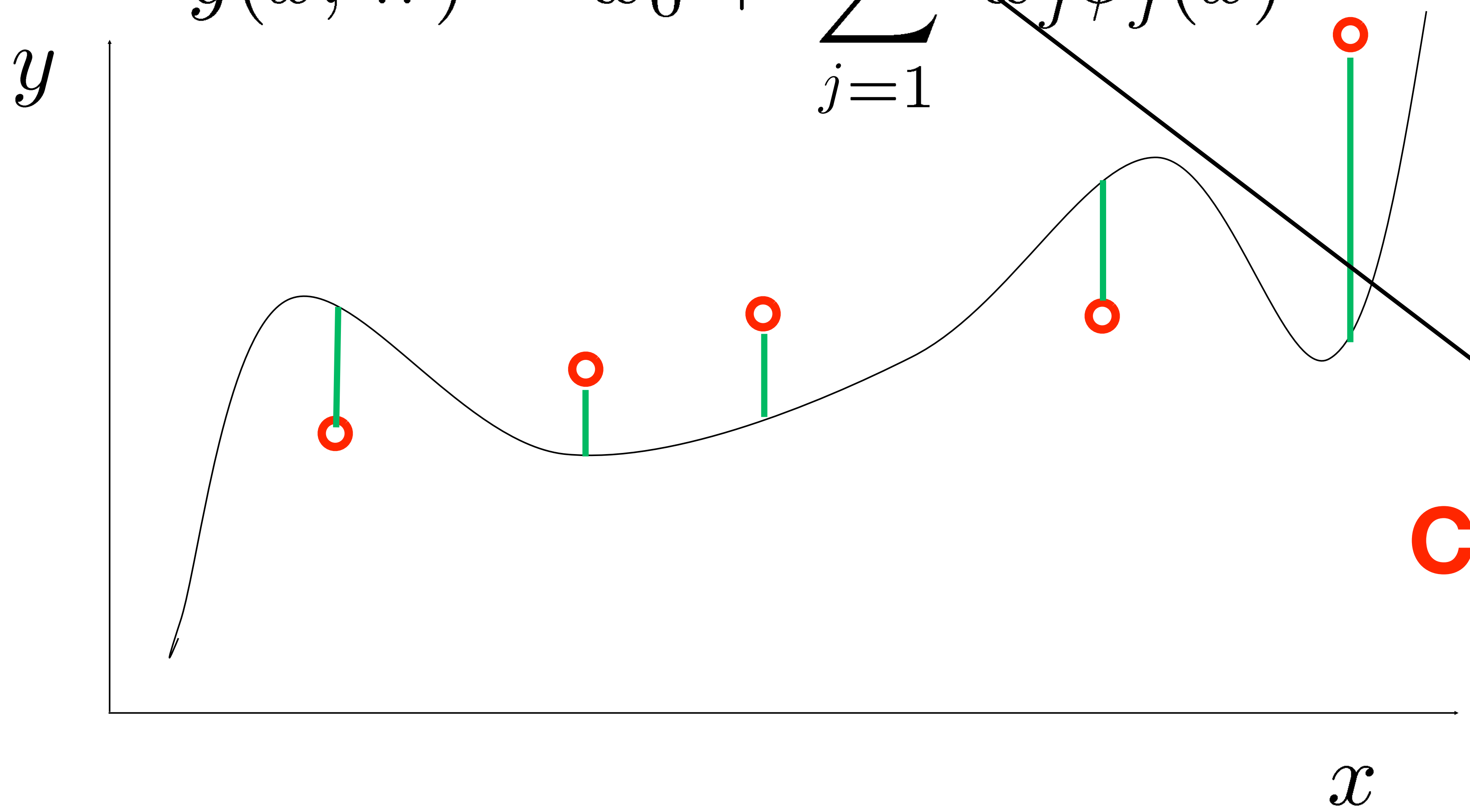
$$\nabla E(\mathbf{w}) = \sum_{i=1}^{N} \mathbf{w}^T \mathbf{x}_i \mathbf{x}_i^T - \sum_{i=1}^{N} t_i \mathbf{x}_i^T = (0 \quad 0) \Rightarrow \mathbf{w}^T \underbrace{\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T}_{=:A^T} = \underbrace{\sum_{i=1}^{N} t_i \mathbf{x}_i^T}_{=:b^T}$$

Polynomial Regression

Now we have: $\mathcal{X} = \mathbb{R}, \ \mathcal{Y} = \mathbb{R}, \ \phi_j(x) = x^j$

Given: data points $(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)$

$$y(x, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x)$$



Data Set Size

Model Complexity

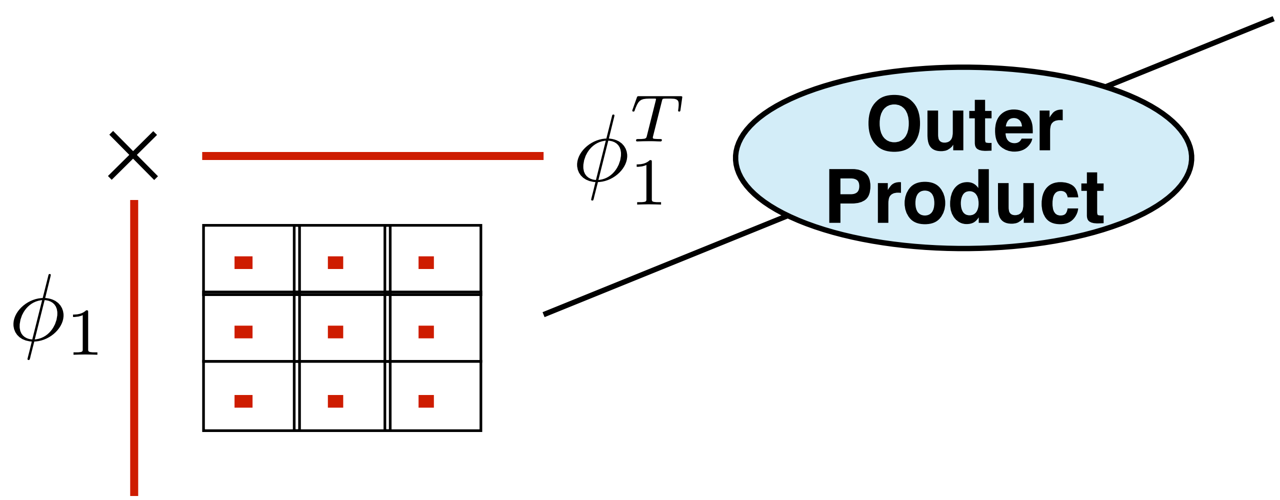
Polynomial Regression

We define: $\boldsymbol{\phi}(x) := (1, \phi_1(x), \dots, \phi_{M-1}(x))^T$ **"Basis functions"**

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

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \phi(x_i) - t_i)^2$$

$$\nabla E(\mathbf{w}) = \mathbf{w}^T \left(\sum_{i=1}^{N} \phi(x_i) \phi(x_i)^T \right) - \sum_{i=1}^{N} t_i \phi(x_i)^T$$



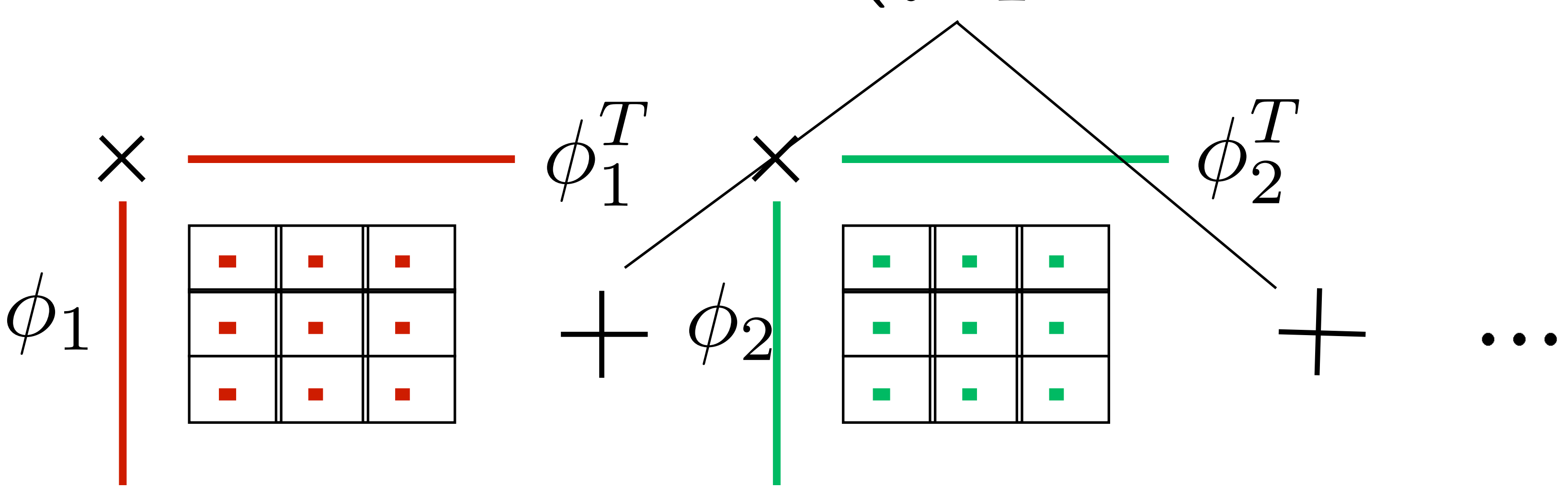
Polynomial Regression

We define: $\boldsymbol{\phi}(x) := (1, \phi_1(x), \dots, \phi_{M-1}(x))^T$

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

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{w}^T\phi(x_i) - t_i)^2$$

$$\nabla E(\mathbf{w}) = \mathbf{w}^T\left(\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T\right) - \sum_{i=1}^{N} t_i\phi(x_i)^T$$



Polynomial Regression

We define: $\phi(x) := (1, \phi_1(x), \dots, \phi_{M-1}(x))^T$

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

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \phi(x_i) - t_i)^2$$

$$\nabla E(\mathbf{w}) = \mathbf{w}^T \left(\sum_{i=1}^{N} \phi(x_i)\phi(x_i)^T \right) - \sum_{i=1}^{N} t_i \phi(x_i)^T$$

Polynomial Regression

Thus, we have: $$\sum_{i=1}^{N} \phi(x_i)\phi(x_i)^T = \Phi^T \Phi$$

where $$\Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \dots & \phi_{M-1}(x_N) \end{pmatrix}$$

$$\nabla E(\mathbf{w}) = \mathbf{w}^T \Phi^T \Phi - \mathbf{t}^T \Phi \qquad \Rightarrow \qquad \Phi^T \Phi \mathbf{w} = \Phi^T \mathbf{t}$$

"Normal Equation"

It follows:

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

"Pseudoinverse" Φ^+

Computing the Pseudoinverse

Mathematically, a pseudoinverse Φ^+ exists for every matrix Φ.

However: If Φ is (close to) singular the direct solution of Φ is numerically unstable.

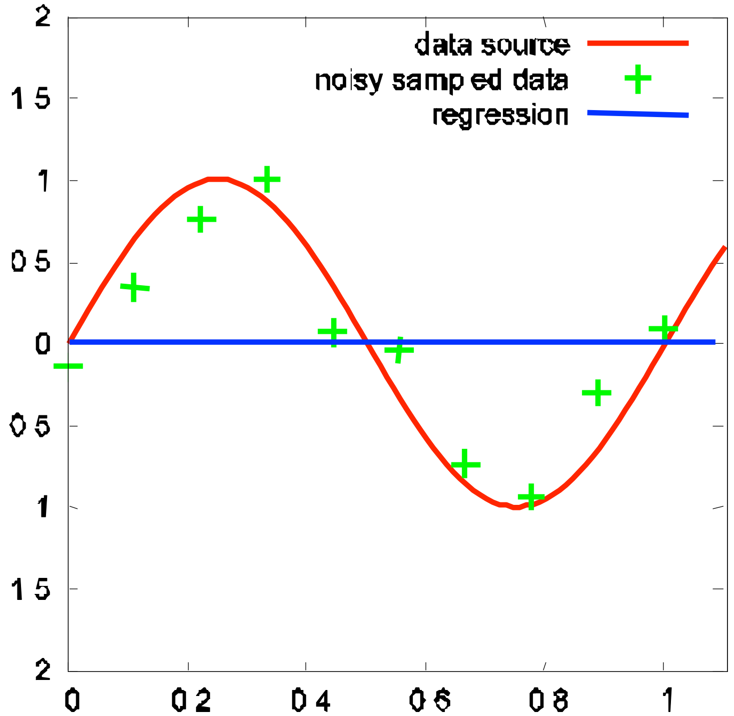
Therefore: Singular Value Decomposition (SVD) is used: $\Phi = UDV^T$ where

- matrices U and V are orthogonal matrices
- D is a diagonal matrix

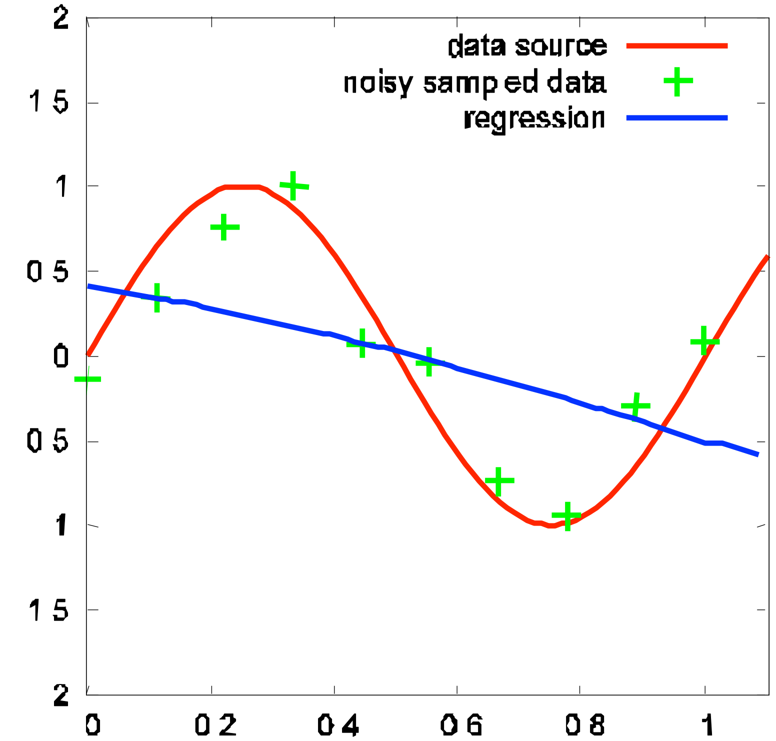
Then: $\Phi^+ = VD^+U^T$ where D^+ contains the ***reciprocal*** of all non-zero elements of D

A Simple Example

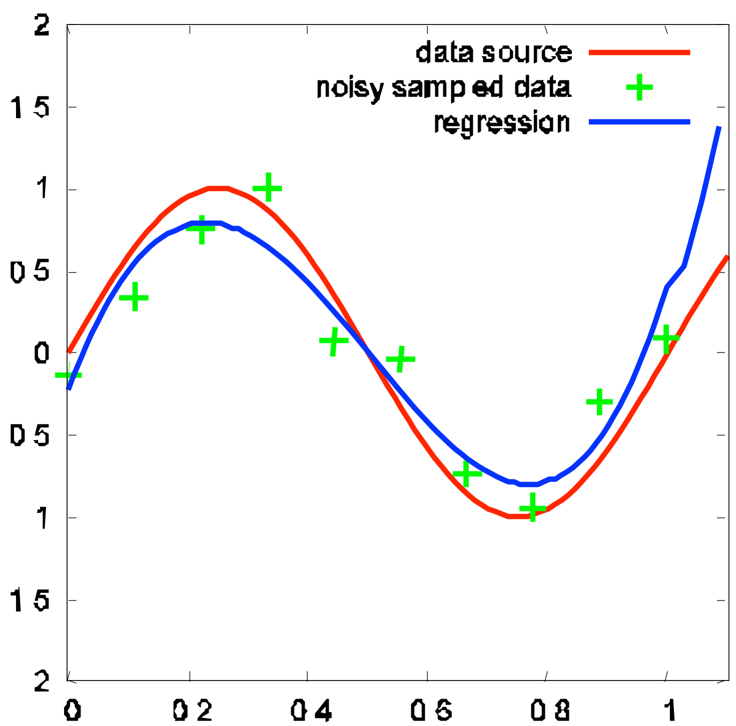
$$\phi_j(x) = x^j$$



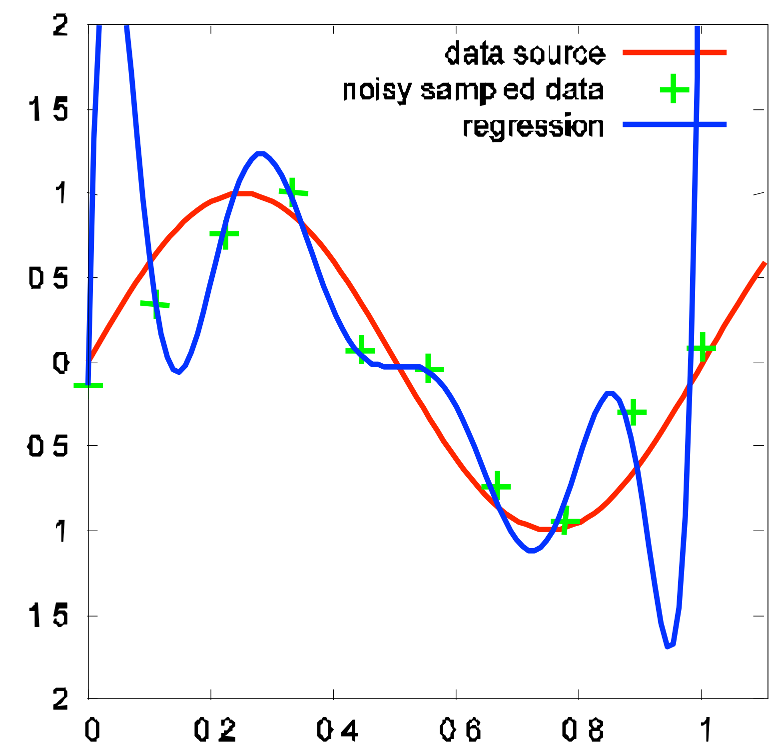
$$N = 10$$
$$M = 1$$



$$N = 10$$
$$M = 3$$

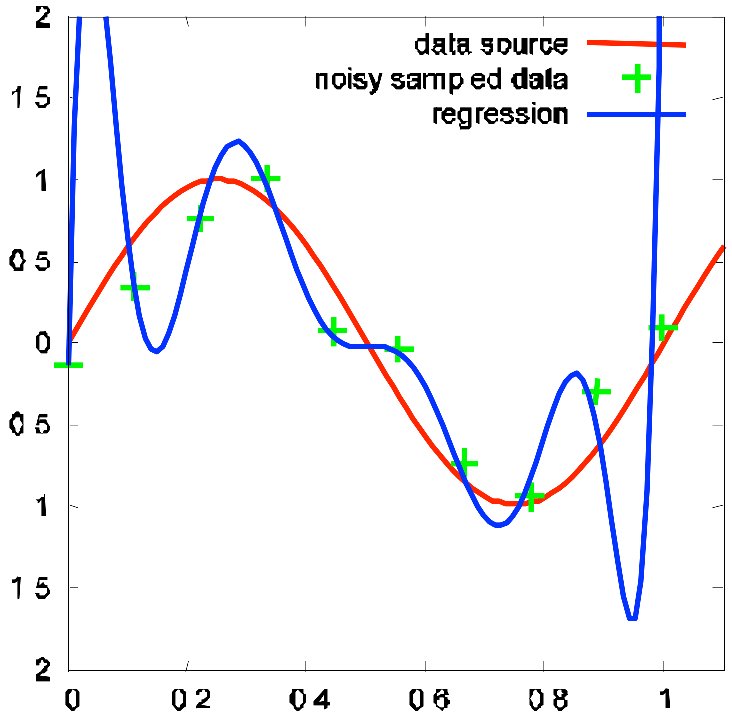


$$N = 10$$
$$M = 5$$

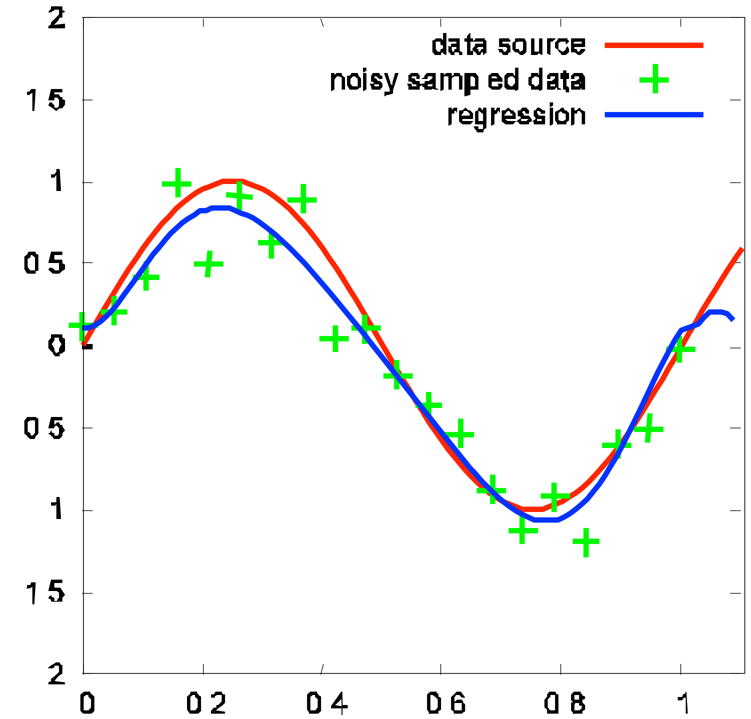


$$N = 10$$
$$M = 10$$

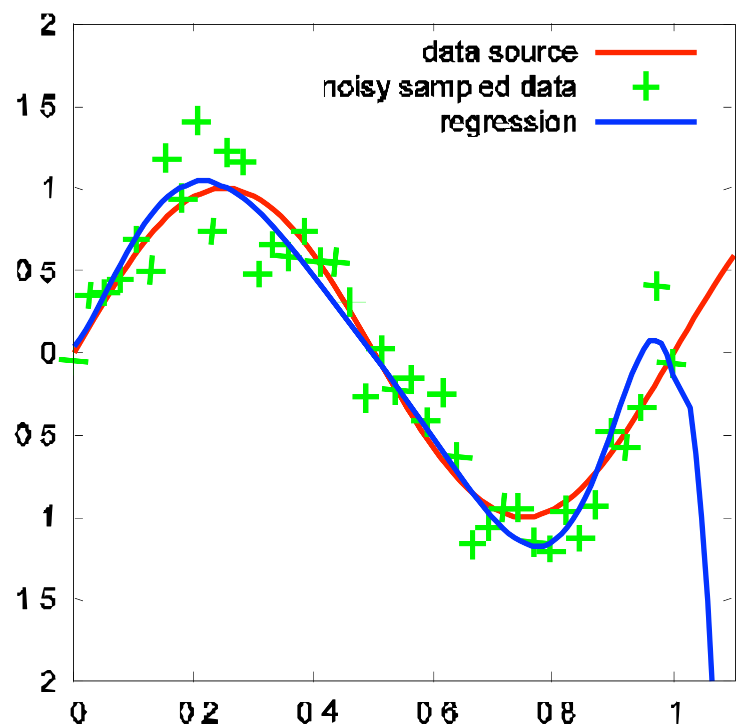
Varying the Sample Size



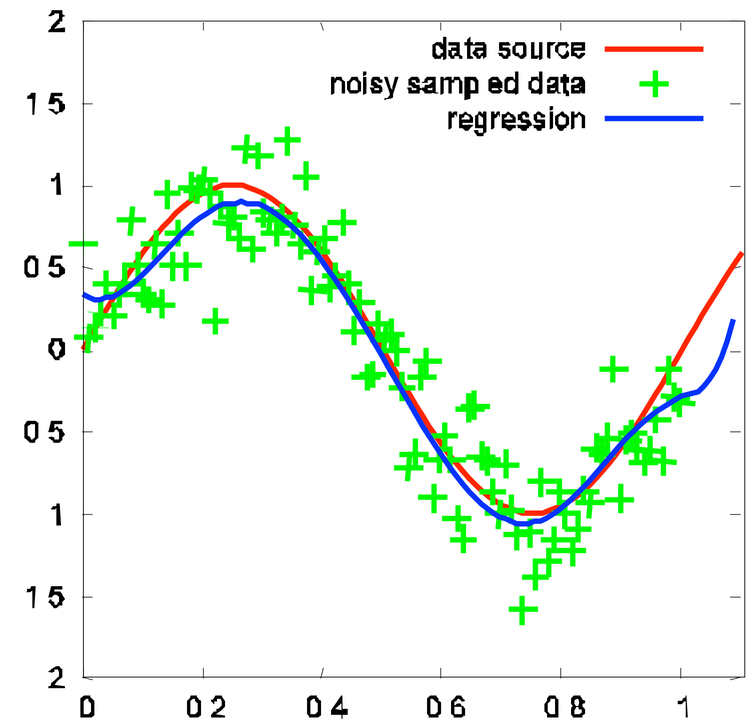
$$N = 10$$
$$M = 10$$



$$N = 20$$
$$M = 10$$

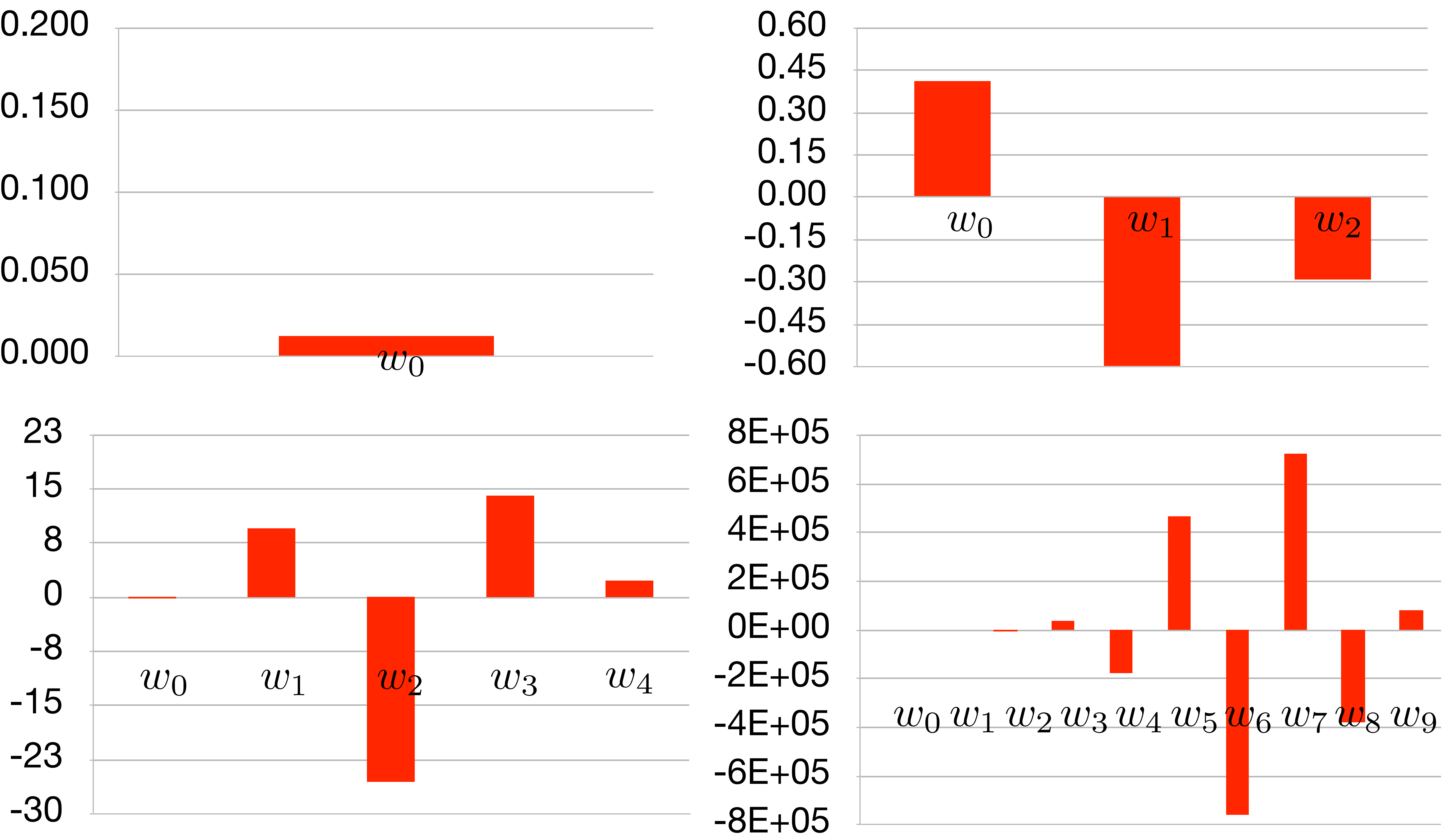


$$N = 40$$
$$M = 10$$



$$N = 100$$
$$M = 10$$

The Resulting Model Parameters



Other Basis Functions

Other basis functions are possible:

- Gaussian basis function:

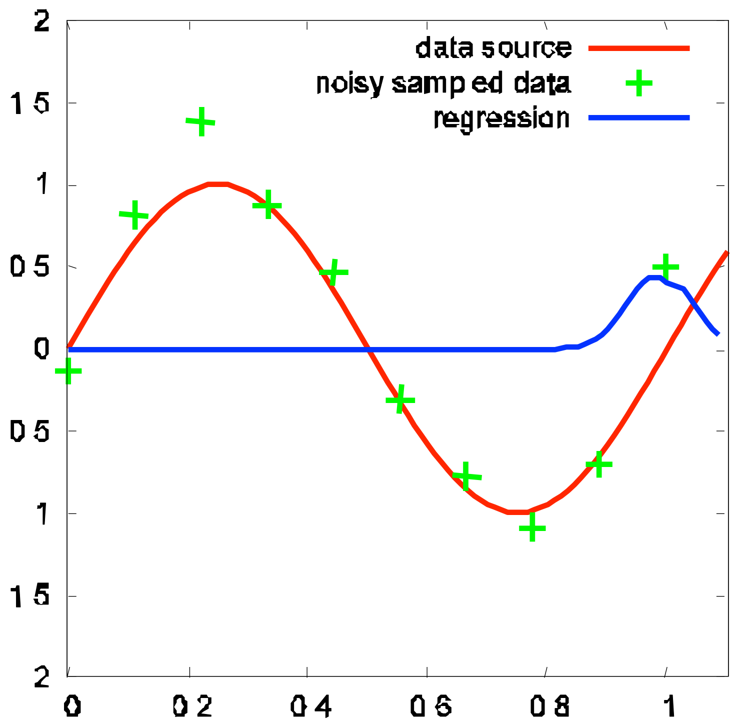
$$\phi_j(x) := \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$$ where $\mu_j \triangleq$ mean val, $s \triangleq$ scale

- Sigmoidal basis function:

$$\phi_j(x) := \sigma\left(\frac{x-\mu_j}{s}\right)$$ where $$\sigma(a) = \frac{1}{1+\exp(-a)}$$

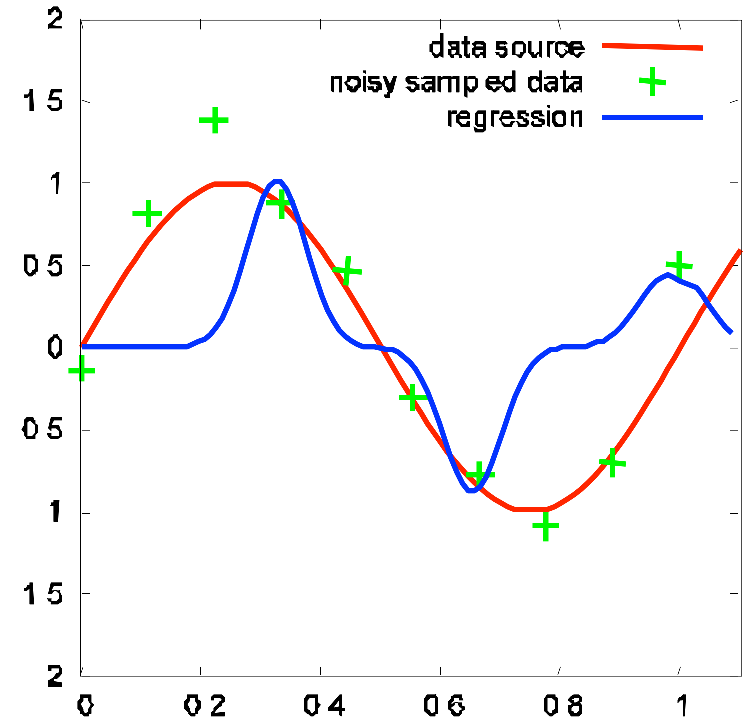
In both cases a set of mean values is required. These define the **locations** of the basis functions.

Gaussian Basis Functions



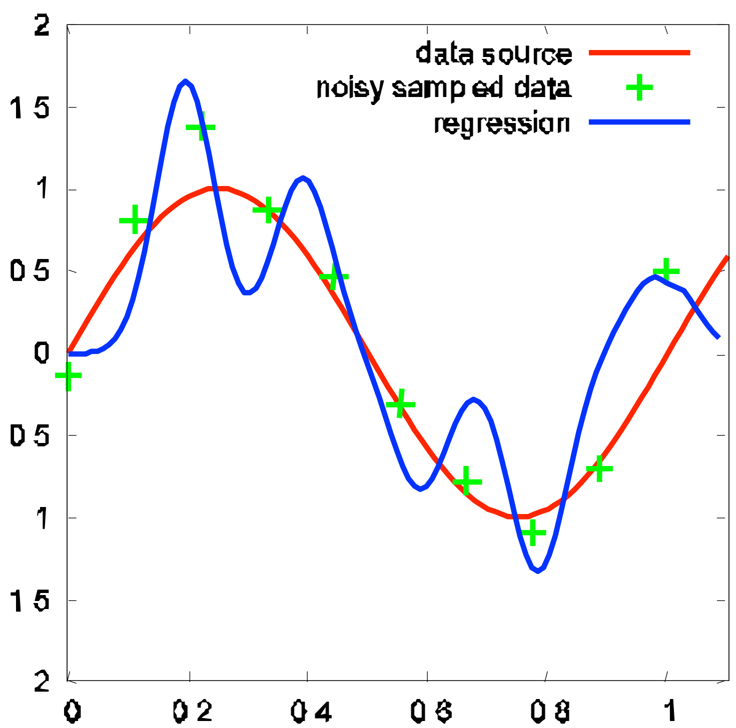
$N = 10$

$M = 1$



$N = 10$

$M = 3$



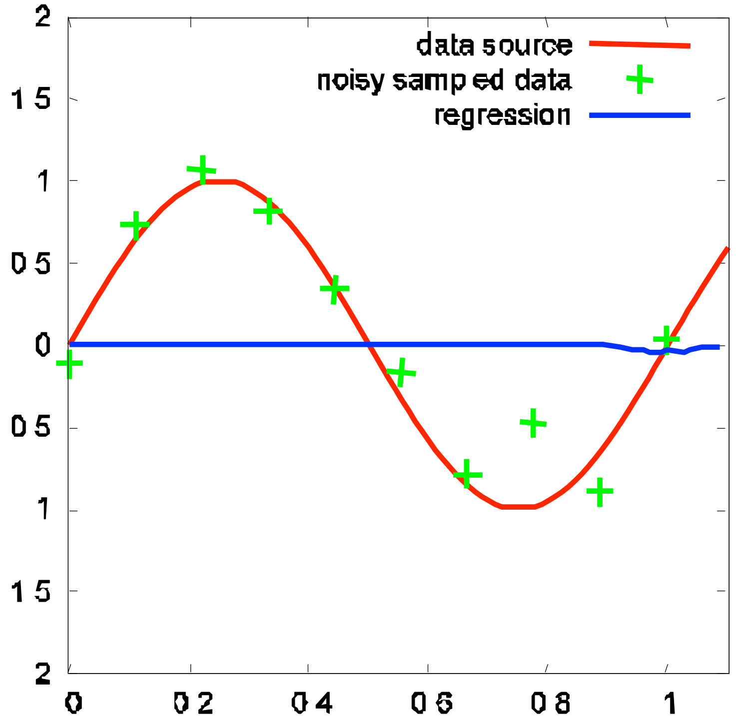
$N = 10$

$M = 5$

$N = 10$

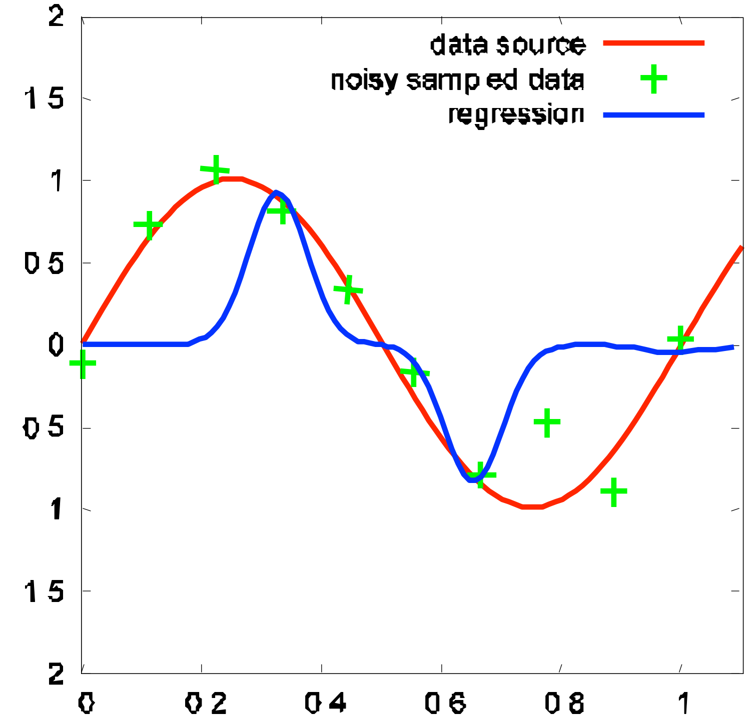
$M = 10$

Sigmoidal Basis Functions



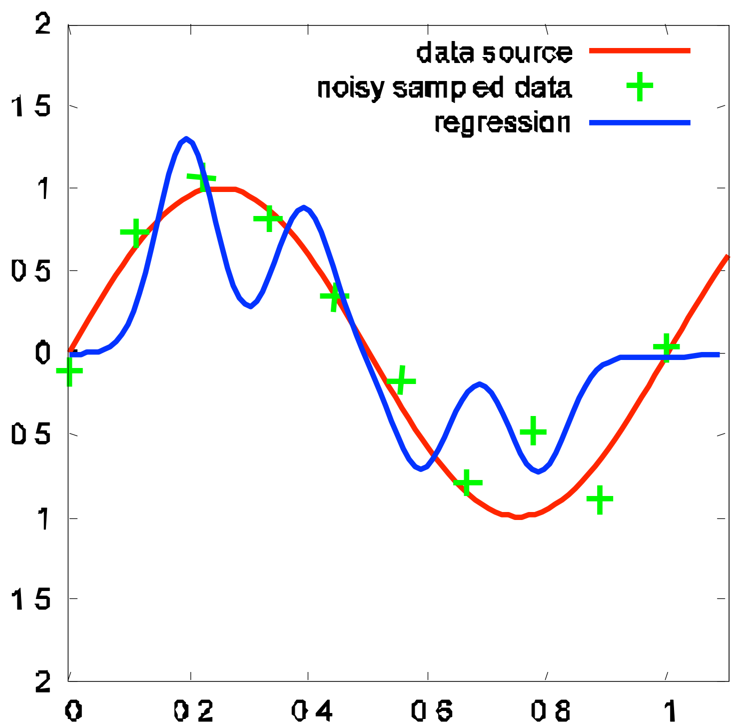
$$N = 10$$

$$M = 1$$



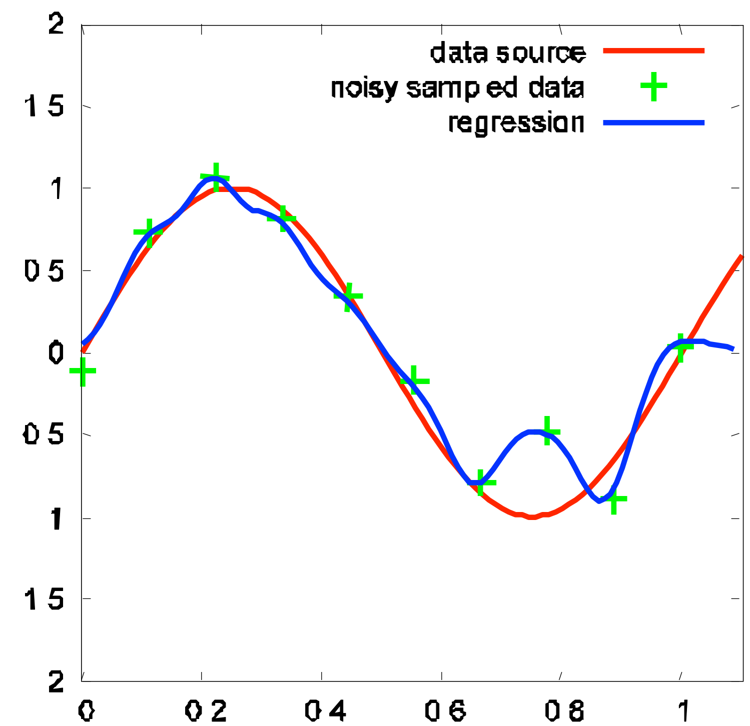
$$N = 10$$

$$M = 3$$



$$N = 10$$

$$M = 5$$



$$N = 10$$

$$M = 10$$

Observations

- The higher the model complexity grows, the better is the fit to the data
- If the model complexity is too high, all data points are explained well, but the resulting model oscillates very much. It can not generalize well.
  This is called ***overfitting.***
- By increasing the size of the data set (number of samples), we obtain a better fit of the model
- More complex models have larger parameters

Problem: How can we find a good model complexity for a given data set with a fixed size?

Regularization

We observed that complex models yield large parameters, leading to oscillation. Idea:

Minimize the error function and the magnitude of the parameters simultaneously

We do this by adding a regularization term :

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(\mathbf{w}^T\phi(x) - t_i\right)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where λ rules the influence of the regularization.

Regularization

As above, we set the derivative to zero:

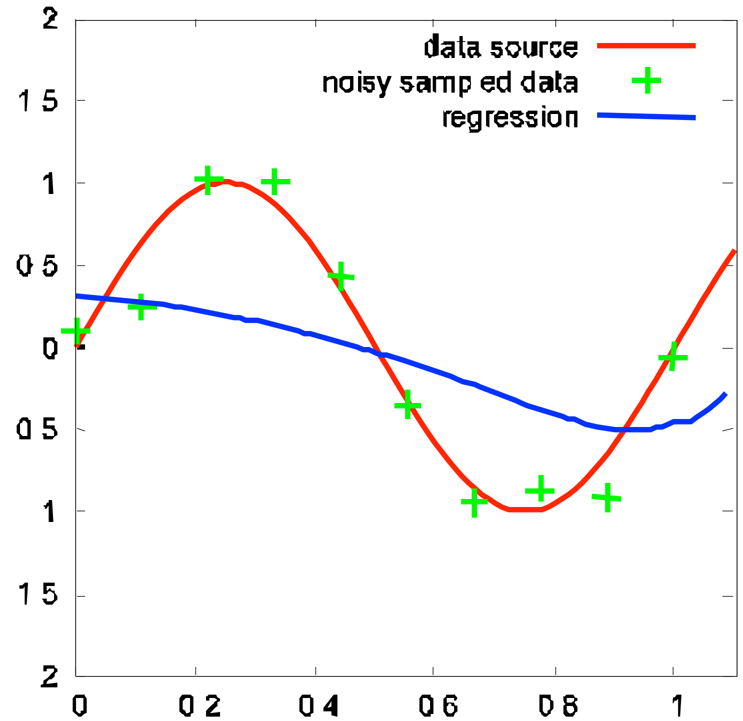
$$\nabla \tilde{E}(\mathbf{w}) = \sum_{i=1}^{N} \left(\mathbf{w}^T \phi(x) - t_i\right) \phi(x)^T + \lambda \mathbf{w}^T \doteq \mathbf{0}^T$$

$$\mathbf{w}^T \Phi^T \Phi + \lambda \mathbf{w}^T = \mathbf{t}^T \Phi \quad \Rightarrow \quad (\lambda I + \Phi^T \Phi)\mathbf{w} = \Phi^T \mathbf{t}$$

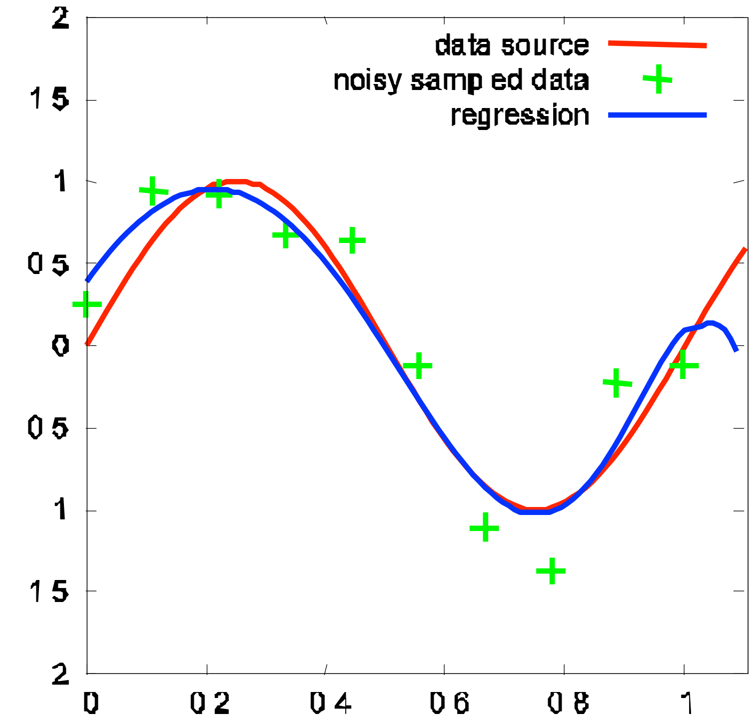
$$\mathbf{w} = (\lambda I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

With regularization, we can find a complex model for a small data set. However, the problem now is to find an appropriate regularization coefficient λ.

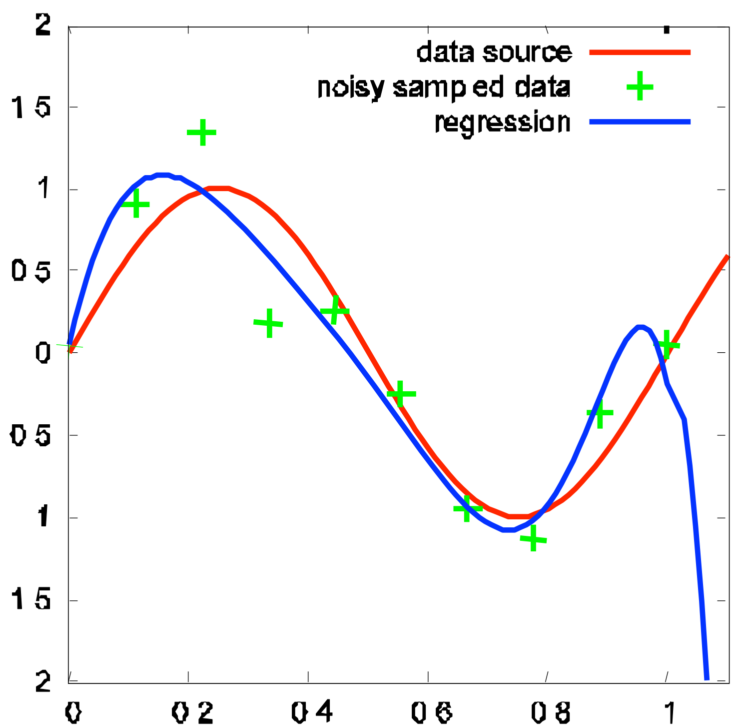
Regularized Results



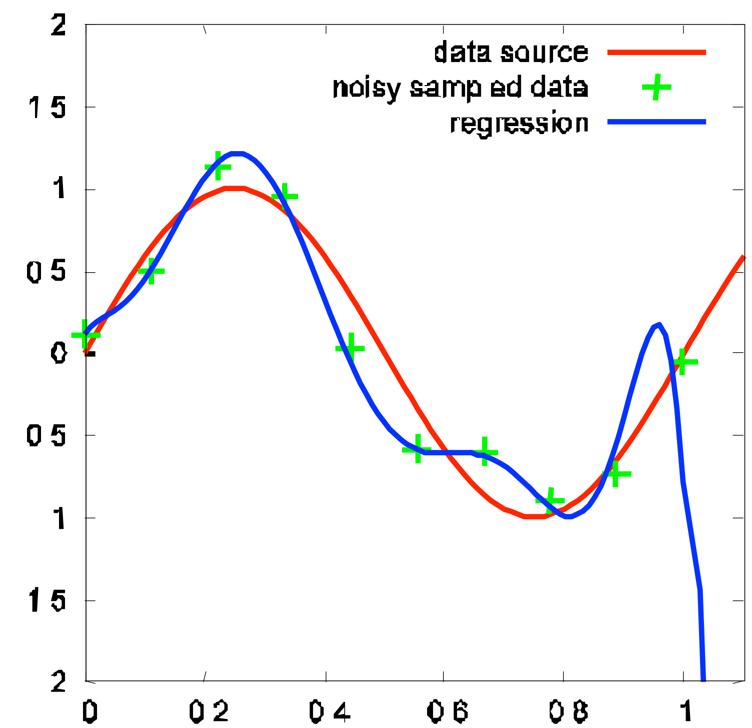
$$N = 10$$
$$M = 10$$
$$\lambda = 1$$



$$N = 10$$
$$M = 10$$
$$\lambda = 10^{-3}$$



$$N = 10$$
$$M = 10$$
$$\lambda = 10^{-6}$$



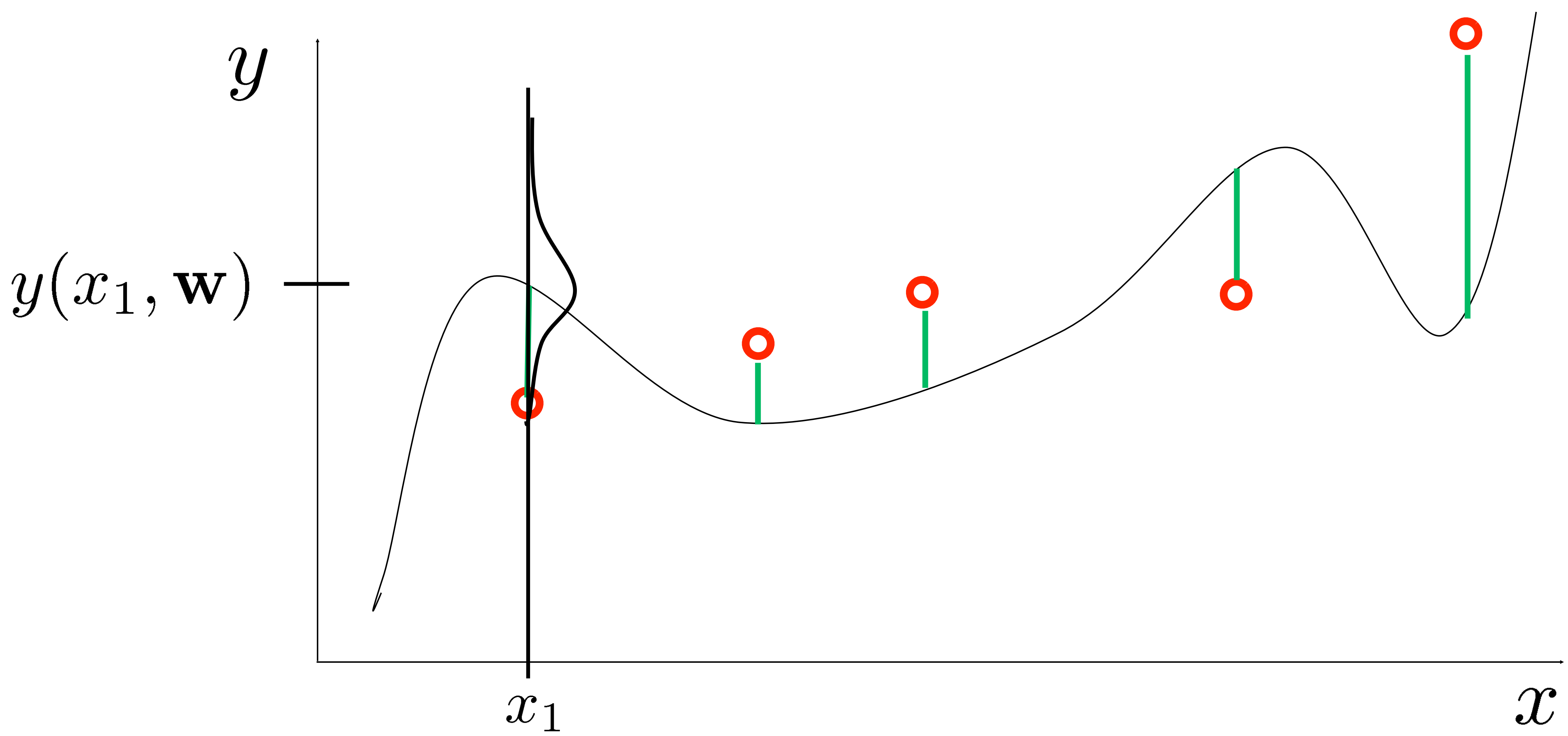
$$N = 10$$
$$M = 10$$
$$\lambda = 10^{-11}$$

The Problem from a Different View

Assume that y is affected by Gaussian noise :

$$t = y(x, \mathbf{w}) + \epsilon \quad \text{where} \quad \epsilon \rightsquigarrow \mathcal{N}(.; 0, \sigma^2)$$

Thus, we have $p(t \mid x, \mathbf{w}, \sigma) = \mathcal{N}(t; y(x, \mathbf{w}), \sigma^2)$



Maximum Likelihood Estimation

Aim: we want to find the $\mathbf{w}$ that maximizes p.

$p(t \mid x, \mathbf{w}, \sigma)$ is the ***likelihood*** of the measured data given a model. Intuitively:

Find parameters $\mathbf{w}$ that maximize the probability of measuring the already measured data t.

"Maximum Likelihood Estimation"

We can think of this as fitting a model $\mathbf{w}$ to the data t.

Note: σ is also part of the model and can be estimated.

For now, we assume σ is known.

Maximum Likelihood Estimation

Given data points: $(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)$

Assumption: points are drawn independently from p:

$$
\begin{aligned}
p(\mathbf{t} \mid \mathbf{x}, \mathbf{w}, \sigma) &= \prod_{i=1}^{N} p(t_i \mid \mathbf{x}, \mathbf{w}, \sigma) \\
&= \prod_{i=1}^{N} \mathcal{N}(t_i; \mathbf{w}^T \boldsymbol{\phi}(x_i), \sigma^2)
\end{aligned}
$$

where:

$$
\begin{aligned}
\mathbf{x} &= (x_1, x_2, \ldots, x_N) \\
\mathbf{t} &= (t_1, t_2, \ldots, t_N)
\end{aligned}
$$

Instead of maximizing p we can also maximize its **logarithm** (monotonicity of the logarithm)

Maximum Likelihood Estimation

$$\mathcal{N} \rightarrow \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

$$\ln p(\mathbf{t} \mid \mathbf{x}, \mathbf{w}, \sigma) = \sum_{i=1}^{N} \ln p(t_i \mid \mathbf{x}, \mathbf{w}, \sigma)$$

$$= \frac{1}{2} \sum_{i=1}^{N} -\ln(\sigma^2) - \ln(2\pi) - \frac{1}{\sigma^2}(\mathbf{w}^T \boldsymbol{\phi}(x_i) - t_i)^2$$

$$= \frac{-N(\ln(\sigma^2) + \ln(2\pi))}{2} - \frac{1}{\sigma^2} \sum_{i=1}^{N} (\mathbf{w}^T \boldsymbol{\phi}(x_i) - t_i)^2$$

Constant for all **w**

Is equal to $E(\mathbf{w})$

The parameters that maximize the likelihood are equal to the minimum of the sum of squared errors

Maximum Likelihood Estimation

$$\ln p(\mathbf{t} \mid \mathbf{x}, \mathbf{w}, \sigma) = \sum_{i=1}^{N} \ln p(t_i \mid \mathbf{x}, \mathbf{w}, \sigma)$$

$$\mathcal{N} \rightarrow \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

$$= \frac{1}{2} \sum_{i=1}^{N} -\ln(\sigma^2) - \ln(2\pi) - \frac{1}{\sigma^2}(\mathbf{w}^T \boldsymbol{\phi}(x_i) - t_i)^2$$

$$= \frac{-N(\ln(\sigma^2) + \ln(2\pi))}{2} - \frac{1}{\sigma^2} \sum_{i=1}^{N} (\mathbf{w}^T \boldsymbol{\phi}(x_i) - t_i)^2$$

$$\mathbf{w}_{ML} := \arg\max_{\mathbf{w}} \ln p(\mathbf{t} \mid \mathbf{x}, \mathbf{w}, \sigma) = \arg\min_{\mathbf{w}} E(\mathbf{w}) = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

The ML solution is obtained using the Pseudoinverse

Maximum A-Posteriori Estimation

So far, we searched for parameters $\mathbf{w}$, that maximize the data likelihood. Now, we assume a Gaussian ***prior***:

$$p(\mathbf{w} \mid \sigma_2) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma_2 I)$$

Using this, we can compute the ***posterior*** (Bayes):

$$p(\mathbf{w} \mid x, \mathbf{t}, \sigma_1, \sigma_2) \propto p(t \mid x, \mathbf{w}, \sigma_1) p(\mathbf{w} \mid \sigma_2)$$

Posterior Likelihood Prior

"Maximum A-Posteriori Estimation (MAP)"

Maximum A-Posteriori Estimation

So far, we searched for parameters $\mathbf{w}$, that maximize the data likelihood. Now, we assume a Gaussian ***prior***:

$$p(\mathbf{w} \mid \sigma_2) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma_2 I)$$

Using this, we can compute the ***posterior*** (Bayes):

$$p(\mathbf{w} \mid x, \mathbf{t}, \sigma_1, \sigma_2) \propto p(t \mid x, \mathbf{w}, \sigma_1)p(\mathbf{w} \mid \sigma_2)$$

strictly:

$$p(\mathbf{w} \mid x, \mathbf{t}, \sigma_1, \sigma_2) = \frac{p(t \mid x, \mathbf{w}, \sigma_1)p(\mathbf{w} \mid \sigma_2)}{\int p(t \mid x, \mathbf{w}, \sigma_1)p(\mathbf{w} \mid \sigma_2)d\mathbf{w}}$$

but the denominator is independent of $\mathbf{w}$ and we want to maximize p.

Maximum A-Posteriori Estimation

$$\ln p(\mathbf{w} \mid x, \mathbf{t}, \sigma_1, \sigma_2) \propto \ln p(t \mid x, \mathbf{w}, \sigma_1) + \ln p(\mathbf{w} \mid \sigma_2)$$

$$\text{const.} - \frac{1}{\sigma_1^2} \sum_{i=1}^{N} (\mathbf{w}^T \phi(x) - t_i)^2$$

$$\text{const.} - \frac{1}{2\sigma_2^2} \mathbf{w}^T \mathbf{w}$$

$$\propto -\frac{1}{\sigma_1^2} \left(\sum_{i=1}^{N} (\mathbf{w}^T \phi(x) - t_i)^2 + \frac{\sigma_1^2}{\sigma_2^2} \mathbf{w}^T \mathbf{w} \right)$$

This is equal to the regularized error minimization.

The MAP Estimate corresponds to a regularized error minimization where $\lambda = (\sigma_1 / \sigma_2)^2$

Summary

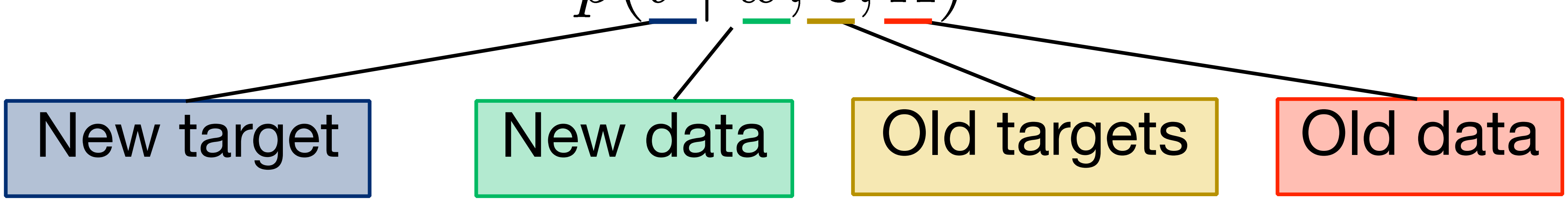
- Regression is a method to find a mathematical model (function) for a given data set
- Regression can be done by minimizing the sum of squared (SSE) errors, i.e. the distances to the data
- Maximum-likelihood estimation uses a probabilis-tic representation to fit a model into noisy data
- Maximum-likelihood under Gaussian noise is equivalent to SSE regression.
- Maximum-a-posteriori (MAP) estimation assumes a (Gaussian) prior on the model parameters
- MAP is solved by regularized regression
-

Bayesian Linear Regression

Bayesian Linear Regression

- Using MAP, we can find optimal model parameters, but for practical applications two questions arise:
- What happens in the case of **sequential data**, i.e. the data points are observed subsequently?
- Can we model the probability of measuring a new data point, given all old data points? This is called the **predictive distribution**:

$$p(t \mid x, \mathbf{t}, \mathbf{x})$$



When Bayes Meets Gauß

If we are given this:

I. $$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \mu, \Sigma_1)$$

II. $$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y} \mid A\mathbf{x} + \mathbf{b}, \Sigma_2)$$

Then it follows (properties of Gaussians):

III. $$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid A\mu + \mathbf{b}, \Sigma_2 + A\Sigma_1 A^T)$$

IV. $$p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}(\mathbf{x} \mid \Sigma(A^T\Sigma_2^{-1}(\mathbf{y} - \mathbf{b}) + \Sigma_1^{-1}\mu), \Sigma)$$

where

$$\Sigma = (\Sigma_1^{-1} + A^T\Sigma_2^{-1}A)^{-1}$$

Sequential Data

Given: Prior mean $\mathbf{m}_0$ and covariance S_0, noise covariance σ $p_0(\mathbf{w} \mid S_0) = \mathcal{N}(\mathbf{w}; \mathbf{m}_0, S_0)$

1. Set $i = 0$
2. Observe data point (x_i, t_i)
3. Formulate the likelihood $p(t_i \mid x_i, \mathbf{w})$ as a function of $\mathbf{w}$ (= Gaussian with mean $\phi(x_i)^T\mathbf{w}$ and covariance σ)
4. Multiply the likelihood with the prior $p_i(\mathbf{w} \mid S_i)$ and normalize (= Gaussian with $\mathbf{m}_{i+1}$ and S_{i+1})
5. This results in a new prior $p_{i+1}(\mathbf{w} \mid S_{i+1})$
6. Go back to 1. if there are still data points available

Comparison: the Standard Bayes Filter

$$\mathrm{Bel}(x_t) = p(x_t \mid u_1, z_1, \ldots, u_t, z_t)$$

(Bayes) $$= \eta\ p(z_t \mid x_t, u_1, z_1, \ldots, u_t) p(x_t \mid u_1, z_1, \ldots, u_t)$$

(Markov) $$= \eta\ p(z_t \mid x_t) p(x_t \mid u_1, z_1, \ldots, u_t)$$

(Tot. prob.) $$= \eta\ p(z_t \mid x_t) \int p(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \, p(x_{t-1} \mid u_1, z_1, \ldots, u_t) dx_{t-1}$$

(Markov) $$= \eta\ p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid u_1, z_1, \ldots, u_t) dx_{t-1}$$

(Markov) $$= \eta\ p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) dx_{t-1}$$

$$= \eta\ p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) \mathrm{Bel}(x_{t-1}) dx_{t-1}$$

Comparison: the Standard Bayes Filter

$$\text{Bel}(x_t) = p(x_t \mid u_1, z_1, \dots, u_t, z_t)$$

(Bayes) $$= \eta\ p(z_t \mid x_t, u_1, z_1, \dots, u_t) p(x_t \mid u_1, z_1, \dots, u_t)$$

(Markov) $$= \eta\ p(z_t \mid x_t) p(x_t \mid u_1, z_1, \dots, u_t)$$

Note: Different Notation!

A Simple Example

Our aim is to **fit a straight line** into a set of data points.

Assume we have:

Basis functions are equal to identity $\phi(\mathbf{x}) = \mathbf{x}$

Prior mean is zero, prior covariance $\sigma_2^2 = 0.5$, noise variance is $\sigma_1^2 = 0.2^2$

Ground truth is $f(x, \mathbf{a}) = a_0 + a_1 x$ where $a_1 = 0.5$

Data points are sampled from ground truth $a_0 = -0.3$

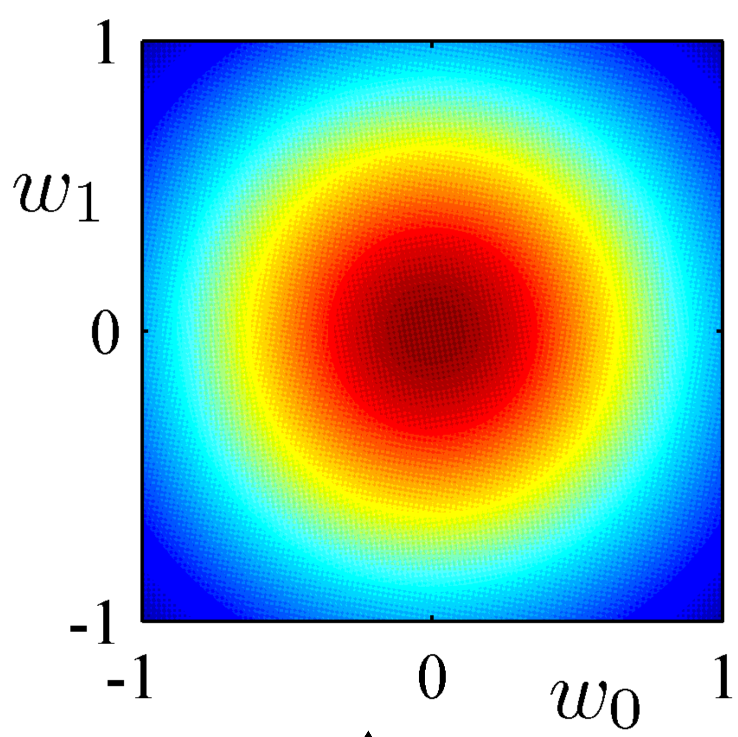
Thus:

We want to recover a_0 and a_1 from the sequentially incoming data points $(x_1, t_1), (x_2, t_2), \ldots$

Bayesian Line Fitting

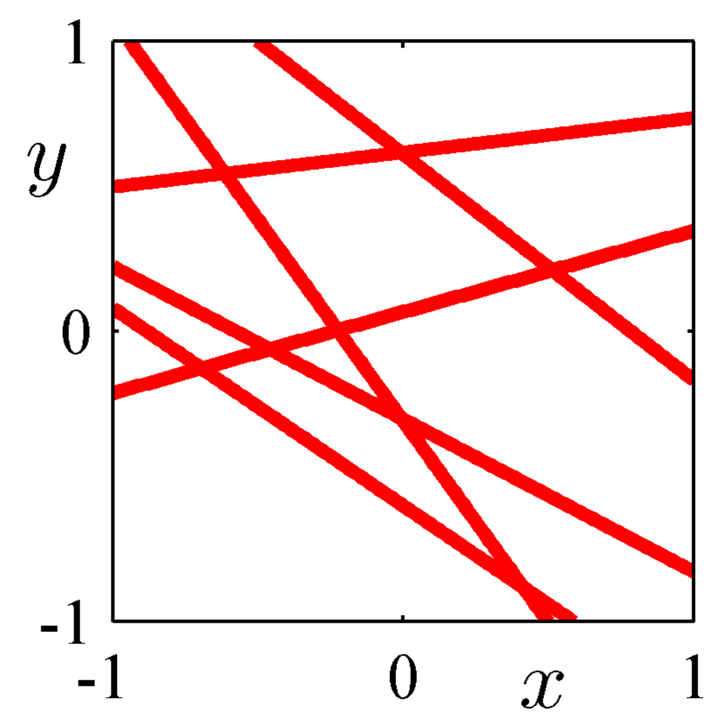
No data points observed

Prior



“Hough Space”

Data Space



Line examples drawn from the prior

From: C.M. Bishop

Bayesian Line Fitting

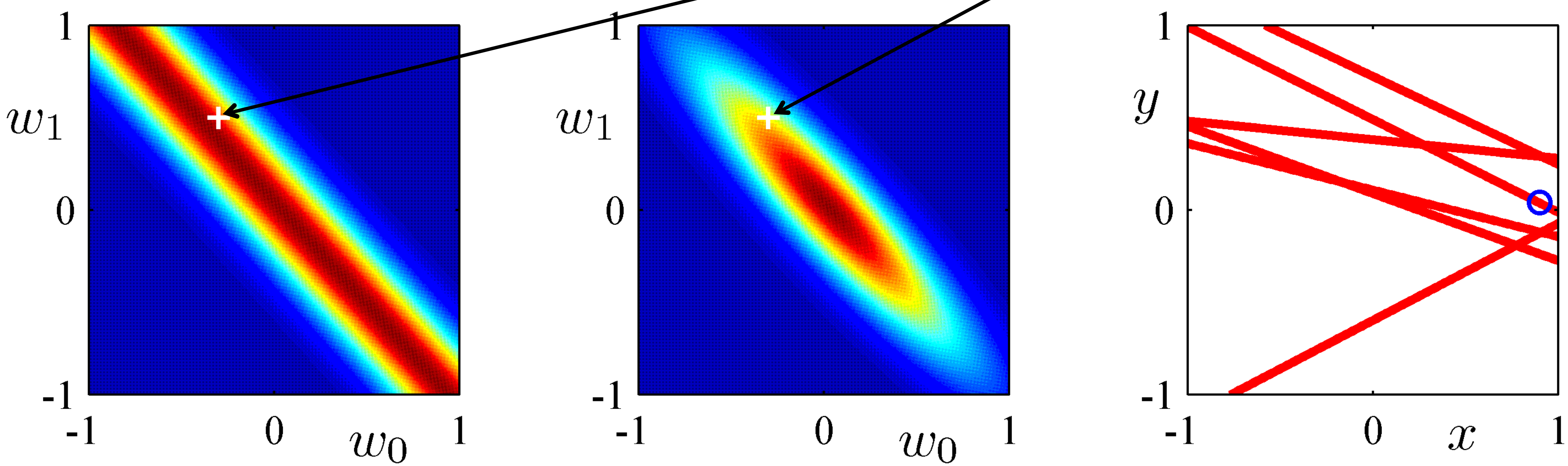
One data point observed

Ground Truth

Likelihood

Prior

Data Space



Line examples drawn from the prior

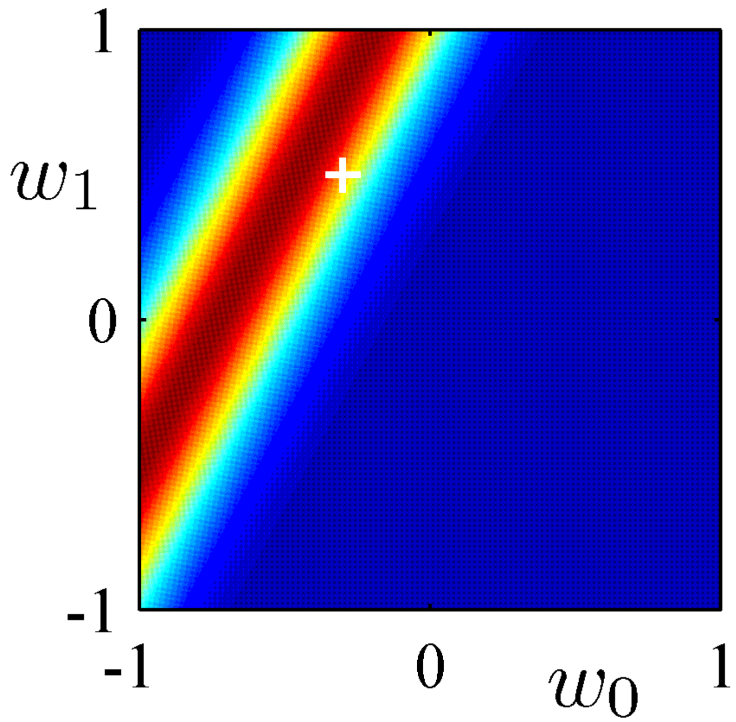
"Hough Space"

From: C.M. Bishop

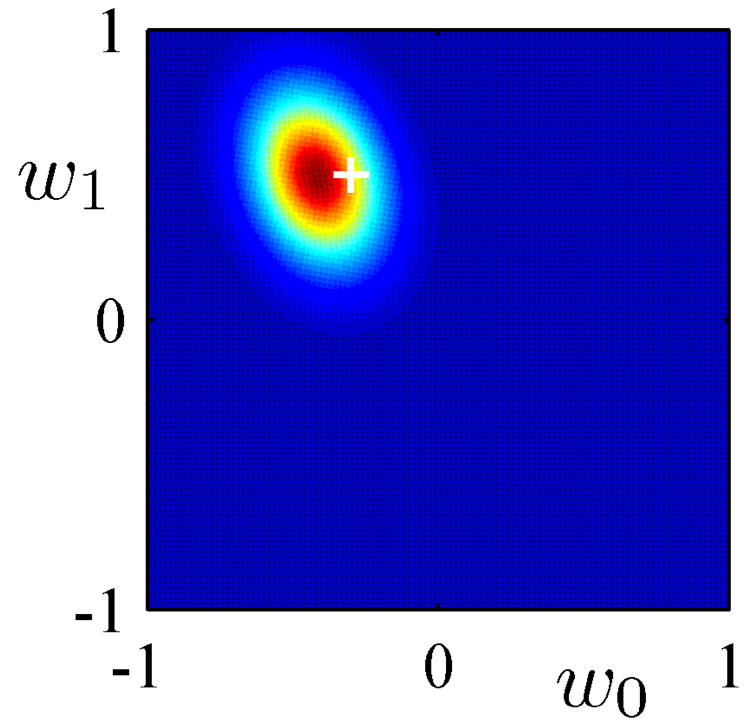
Bayesian Line Fitting

Two data points observed

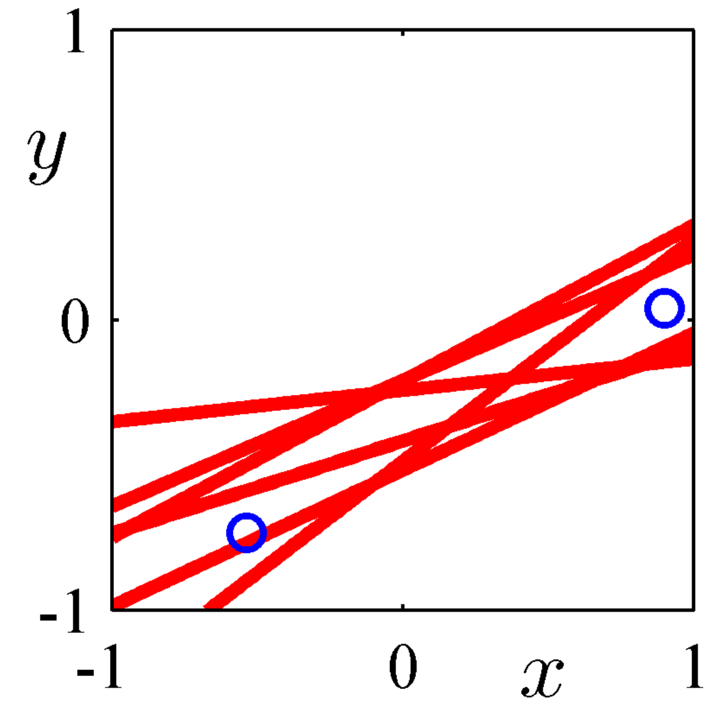
Likelihood



Prior



Data Space



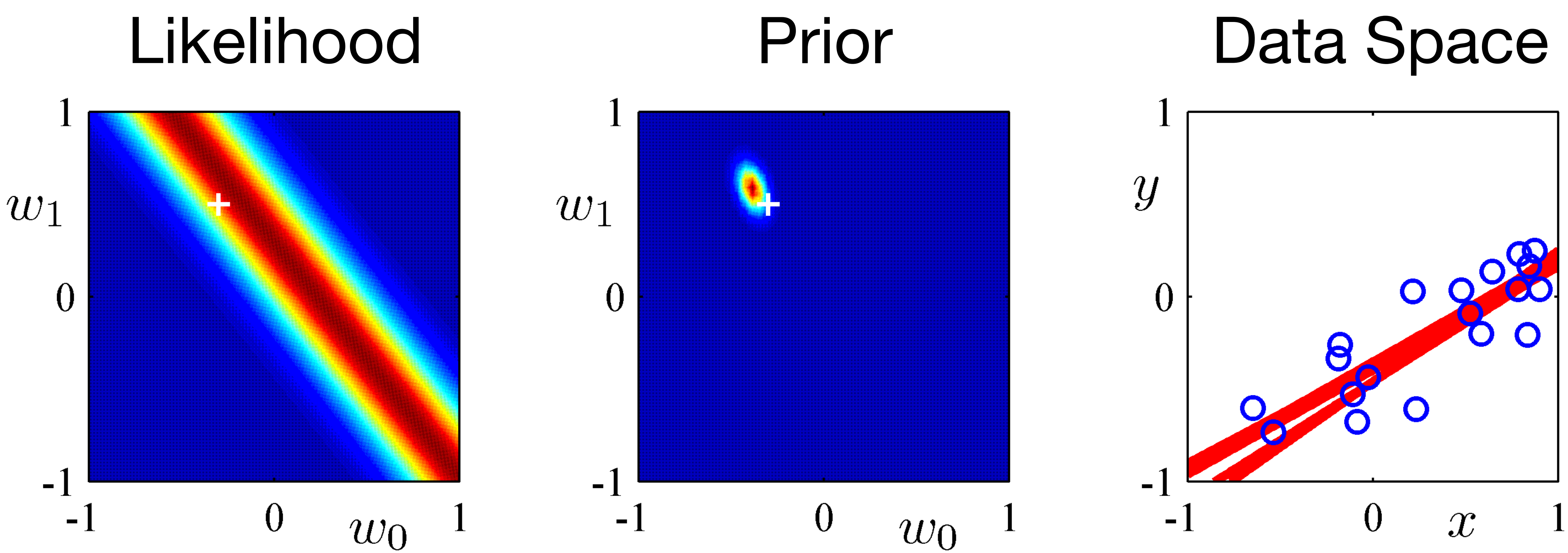
"Hough Space"

Line examples drawn from the prior

From: C.M. Bishop

Bayesian Line Fitting

20 data points observed



"Hough Space"

Line examples drawn from the prior

From: C.M. Bishop

The Predictive Distribution

We obtain the predictive distribution by integrating over all possible model parameters:

$$p(t \mid x, \mathbf{t}, \mathbf{x}) = \int p(t \mid x, \mathbf{w}) p(\mathbf{w} \mid \mathbf{x}, \mathbf{t}) d\mathbf{w}$$

New data likelihood

Old data posterior

As before the posterior is prop. to the likelihood times the prior. But now, we don’t maximize. The posterior can be computed analytically, as the prior is Gaussian.

$p(\mathbf{w} \mid \mathbf{x}, \mathbf{t}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, S_N)$ where

$$S_N^{-1} = S_0^{-1} + \sigma^{-2} \Phi^T \Phi$$

Prior cov

Prior mean

$$\mathbf{w}_N = S_N (S_0^{-1} \mathbf{m}_0 + \sigma^{-2} \Phi^T \mathbf{t})$$

The Predictive Distribution

Using formula III. from above,

$$p(t \mid x, \mathbf{t}, \mathbf{x}) = \int p(t \mid x, \mathbf{w}) p(\mathbf{w} \mid \mathbf{x}, \mathbf{t}) d\mathbf{w}$$

$$= \int \mathcal{N}(t; \mathbf{w}^T \phi(x), \sigma) \mathcal{N}(\mathbf{w}; \mathbf{m}_N, S_N)$$

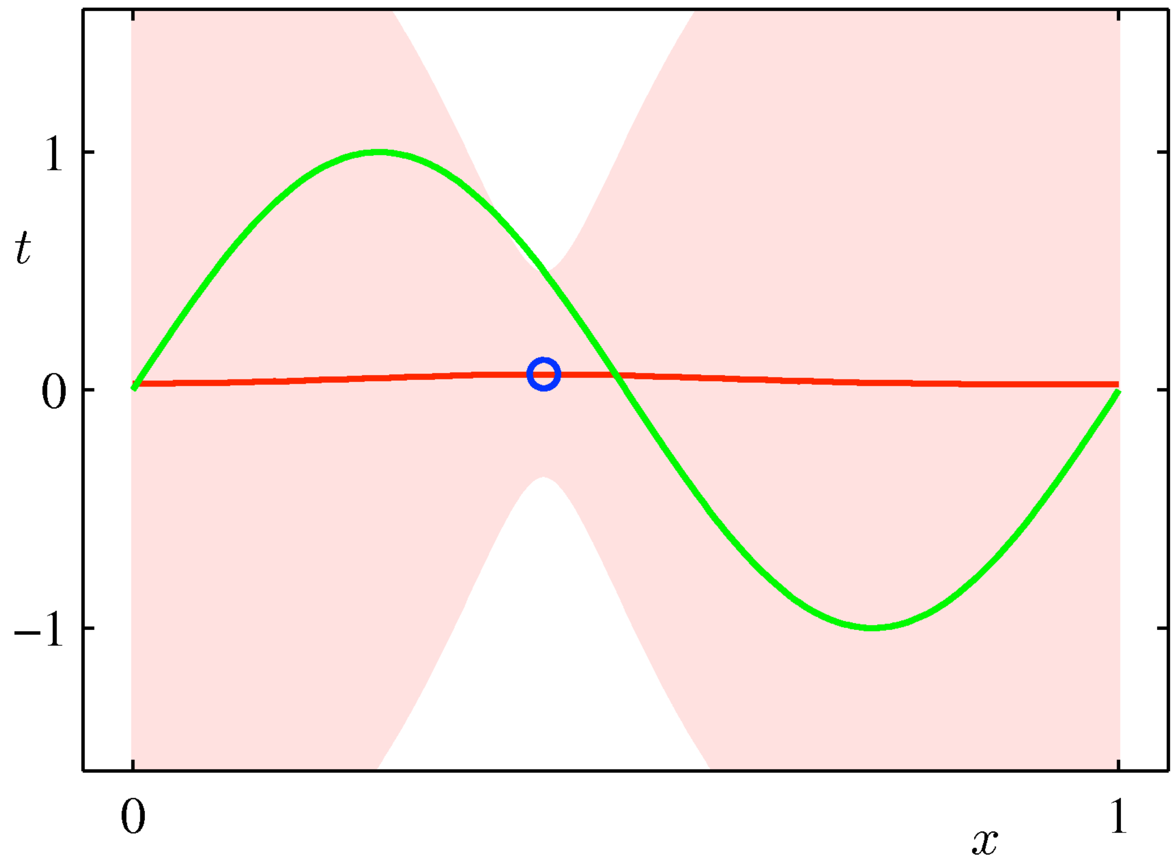
$$= \mathcal{N}(t; \mathbf{m}_N^T \phi(x), \sigma_N^2(x))$$

where

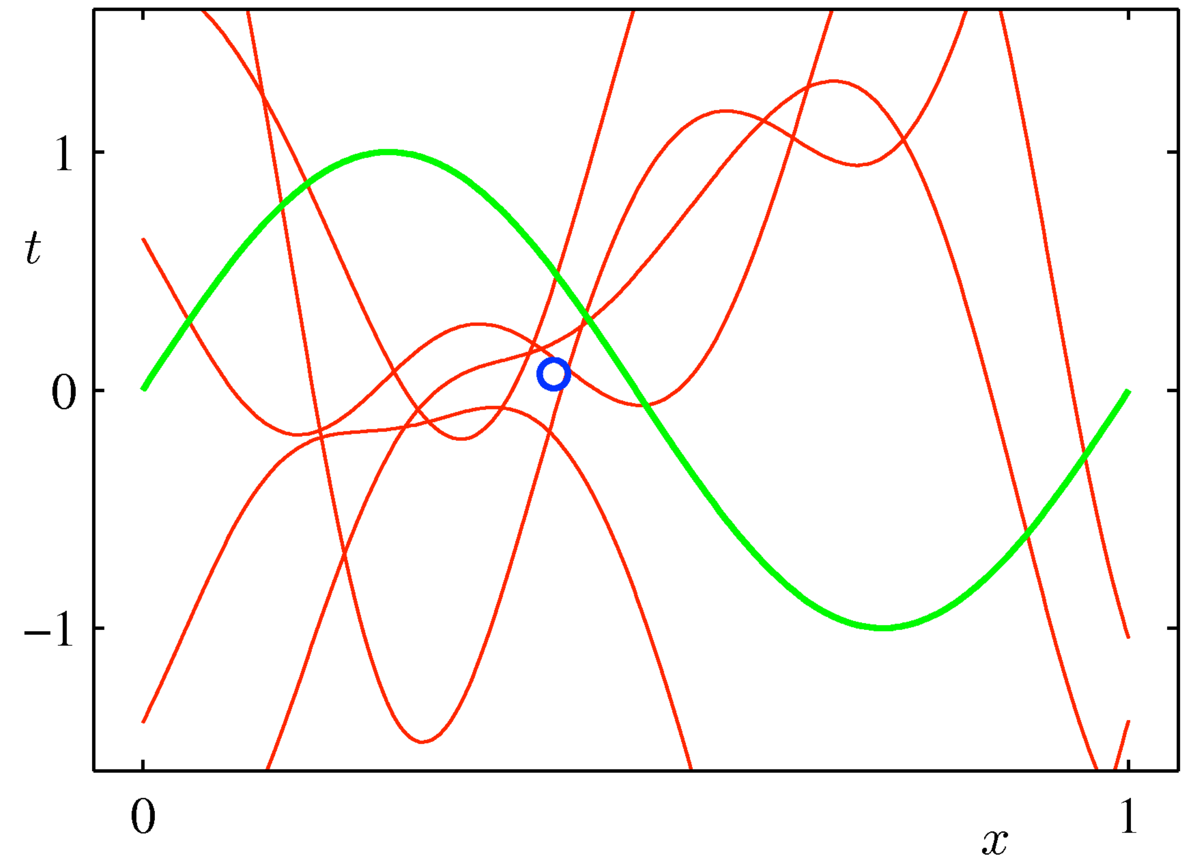
$$\sigma_N^2(x) = \sigma^2 + \phi(x)^T S_N \phi(x)$$

The Predictive Distribution (2)

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



The predictive distribution

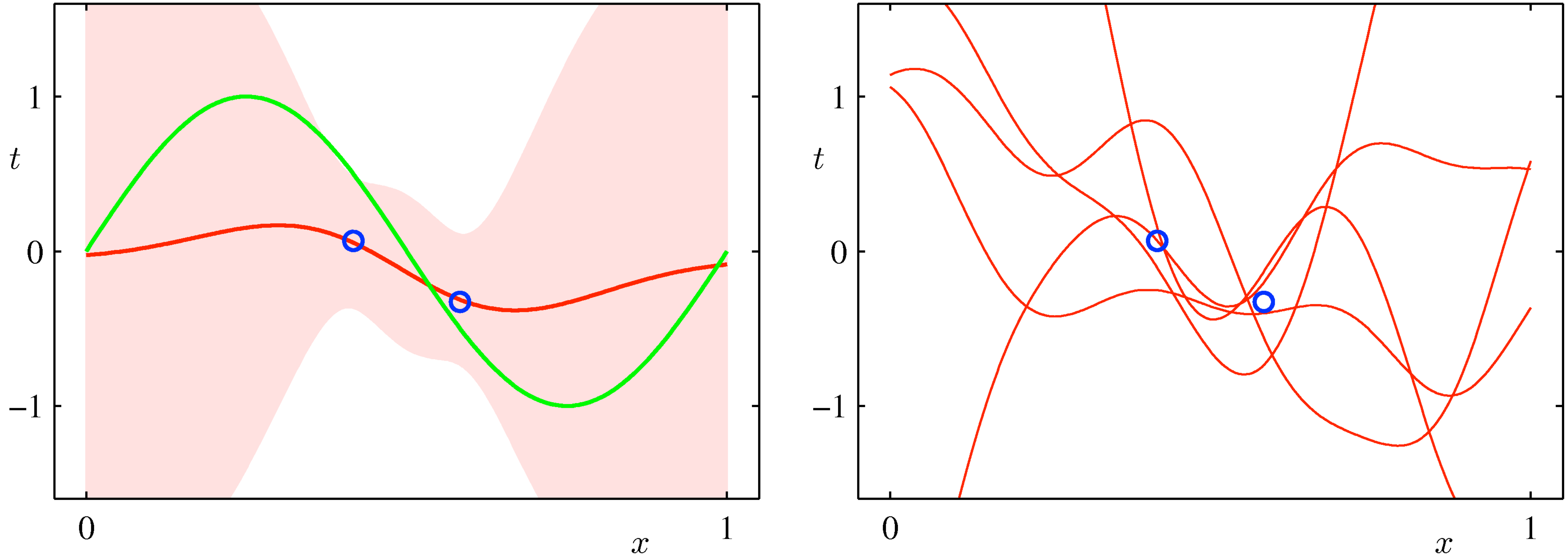


Some samples from the posterior

From: C.M. Bishop

Predictive Distribution (3)

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



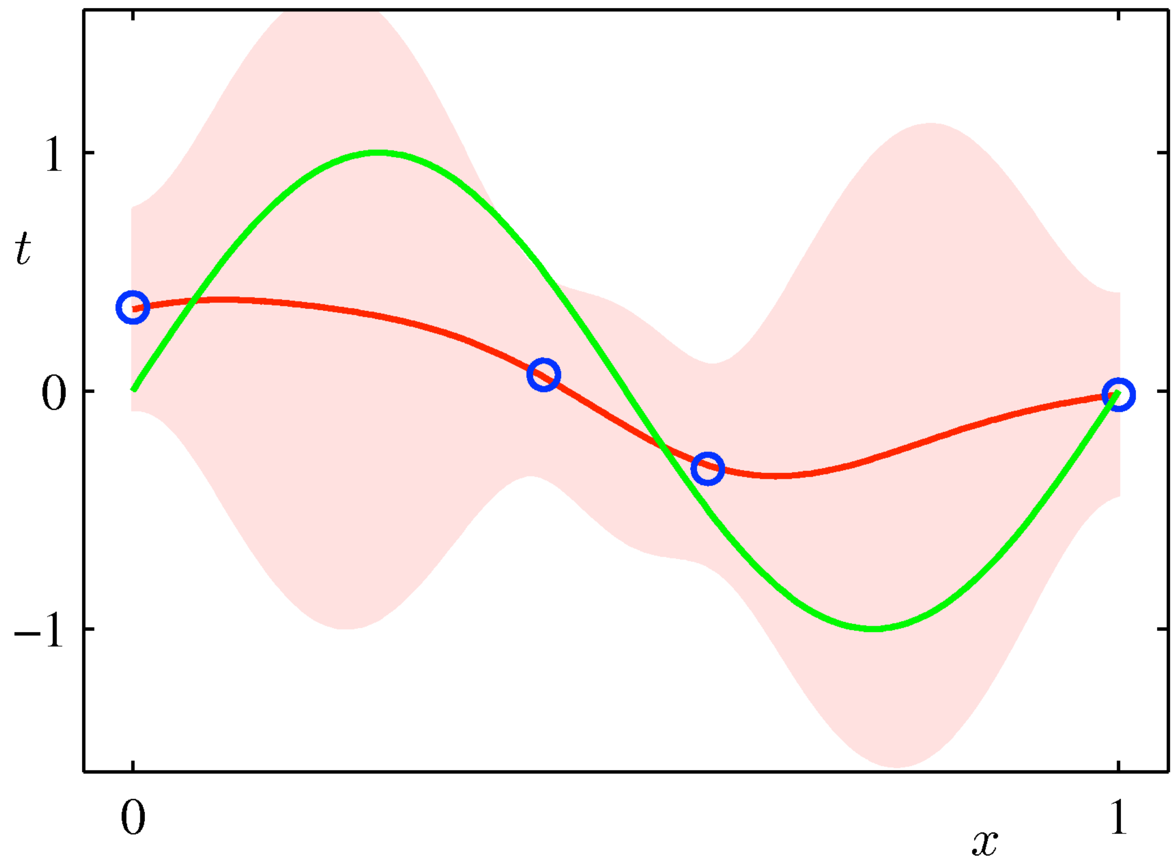
The predictive distribution

Some samples from the posterior

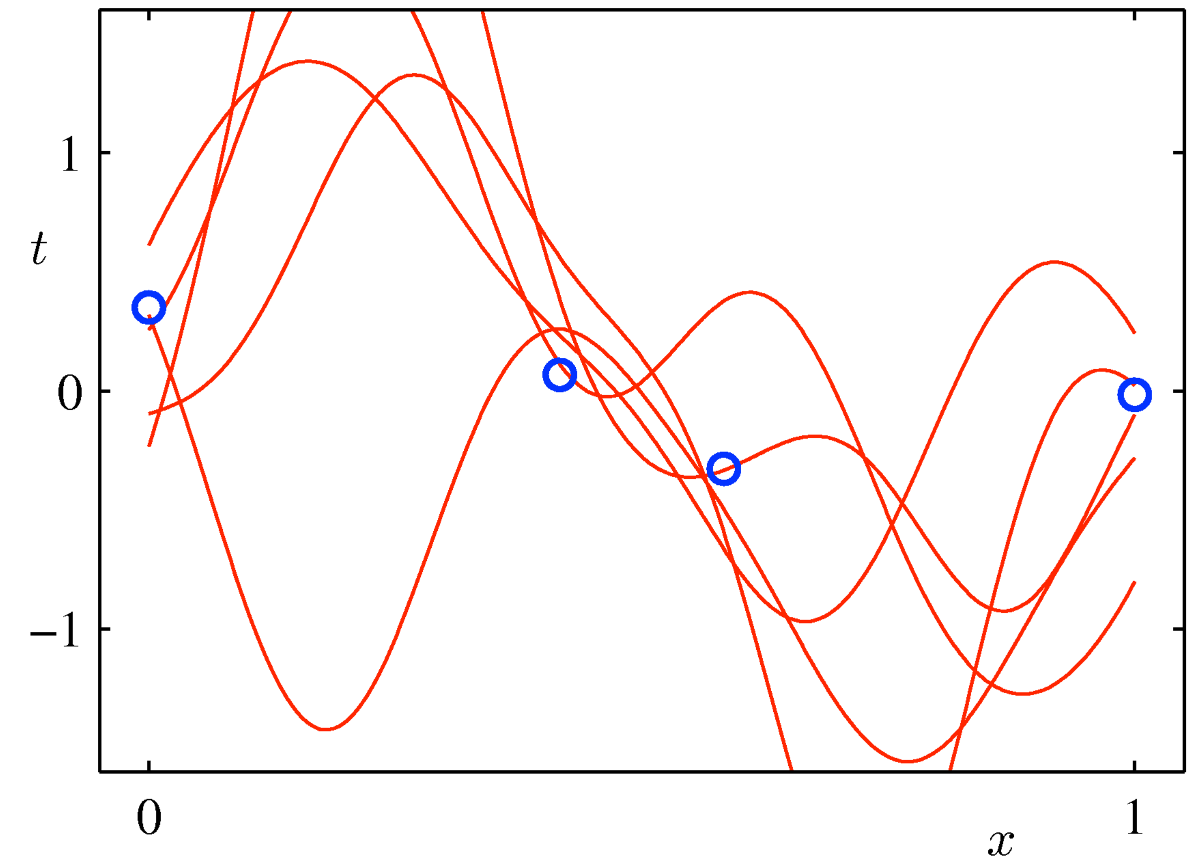
From: C.M. Bishop

Predictive Distribution (4)

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



The predictive distribution

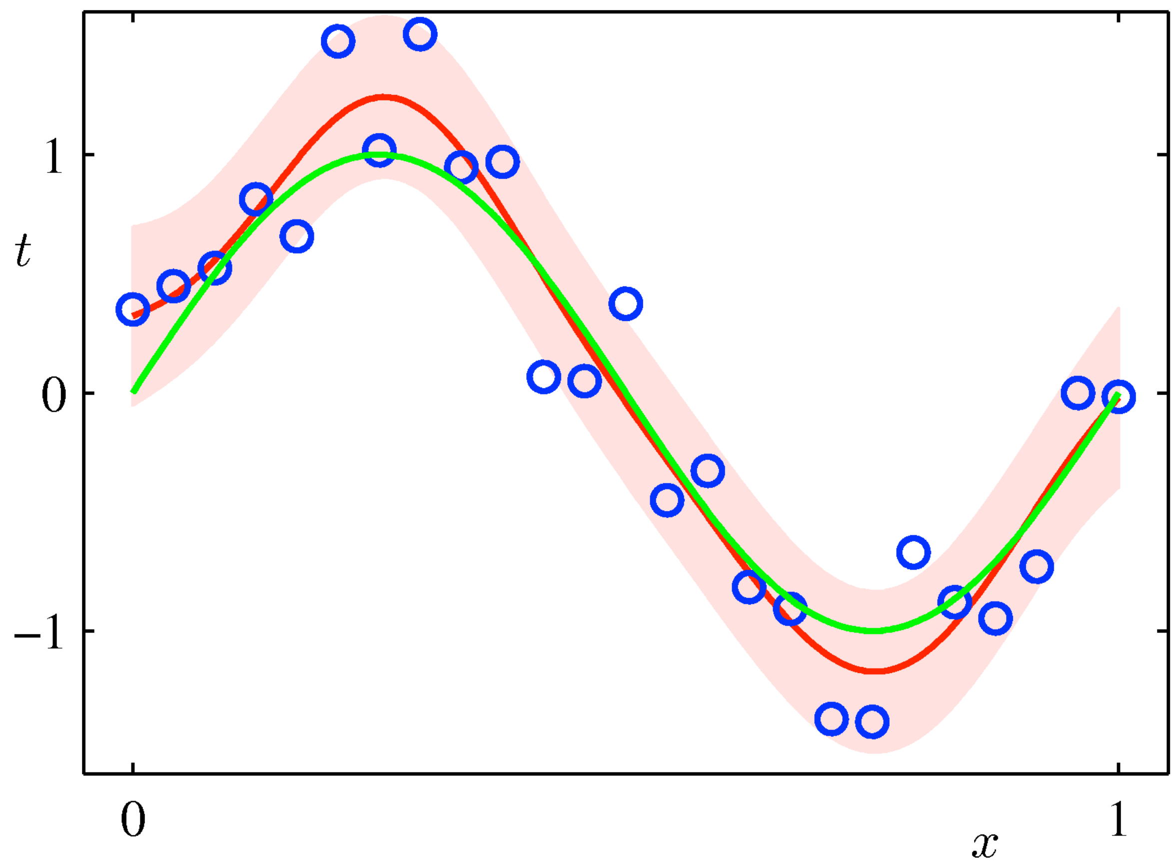


Some samples from the posterior

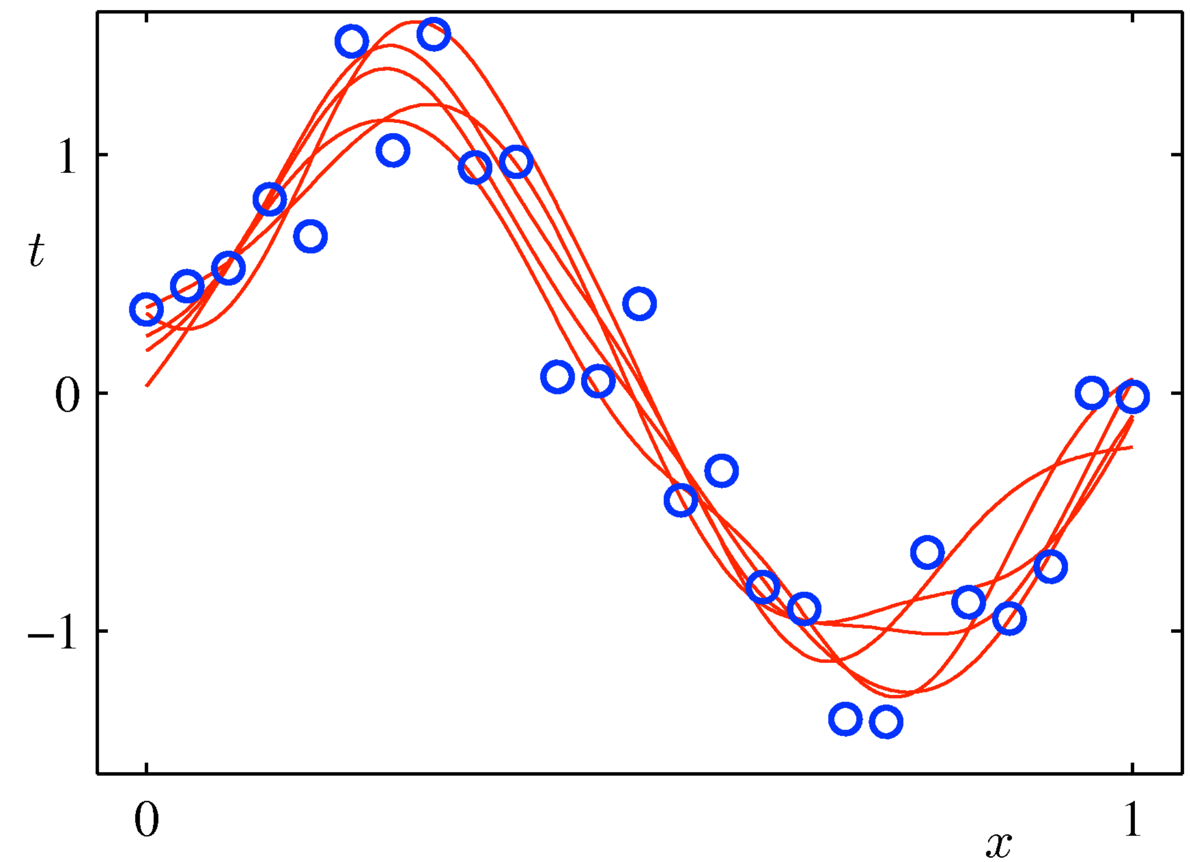
From: C.M. Bishop

Predictive Distribution (5)

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



The predictive distribution



Some samples from the posterior

From: C.M. Bishop

Summary

- Regression can be expressed as a least-squares problem
- To avoid overfitting, we need to introduce a regularisation term with an additional parameter λ
- Regression without regularisation is equivalent to Maximum Likelihood Estimation
- Regression + reg = Maximum A-Posteriori
- Bayesian Linear Regression operates on sequential data and provides the predictive distribution
- When using Gaussian priors (and Gaussian noise), all computations can be done analytically