The week ahead

- **Quiz 7:** mean is 78% and average completion time 5min 48sec *(please, check practice questions!)*
- **Focus videos:** PCA and linear regression available on the class website Tue, Oct 13th
- **Quiz 8, Friday, Oct 10th 6am until Oct 10th 11:59am (noon)**
  - Regularization and Naïve Bayes

Coming up soon

- **Assignment 3 Early bird special** → 1 complete programming question by Mon, Oct 19th 11:59pm (midnight)

CS4641B Machine Learning

Lecture 15: Regularized linear regression

Rodrigo Borela ▸ rborelav@gatech.edu

These slides are adapted based on slides from Andrew Zisserman, Jonathan Taylor, Chao Zhang and Yaser Abu-Mostafa and Mahdi Roozbahani

Outline

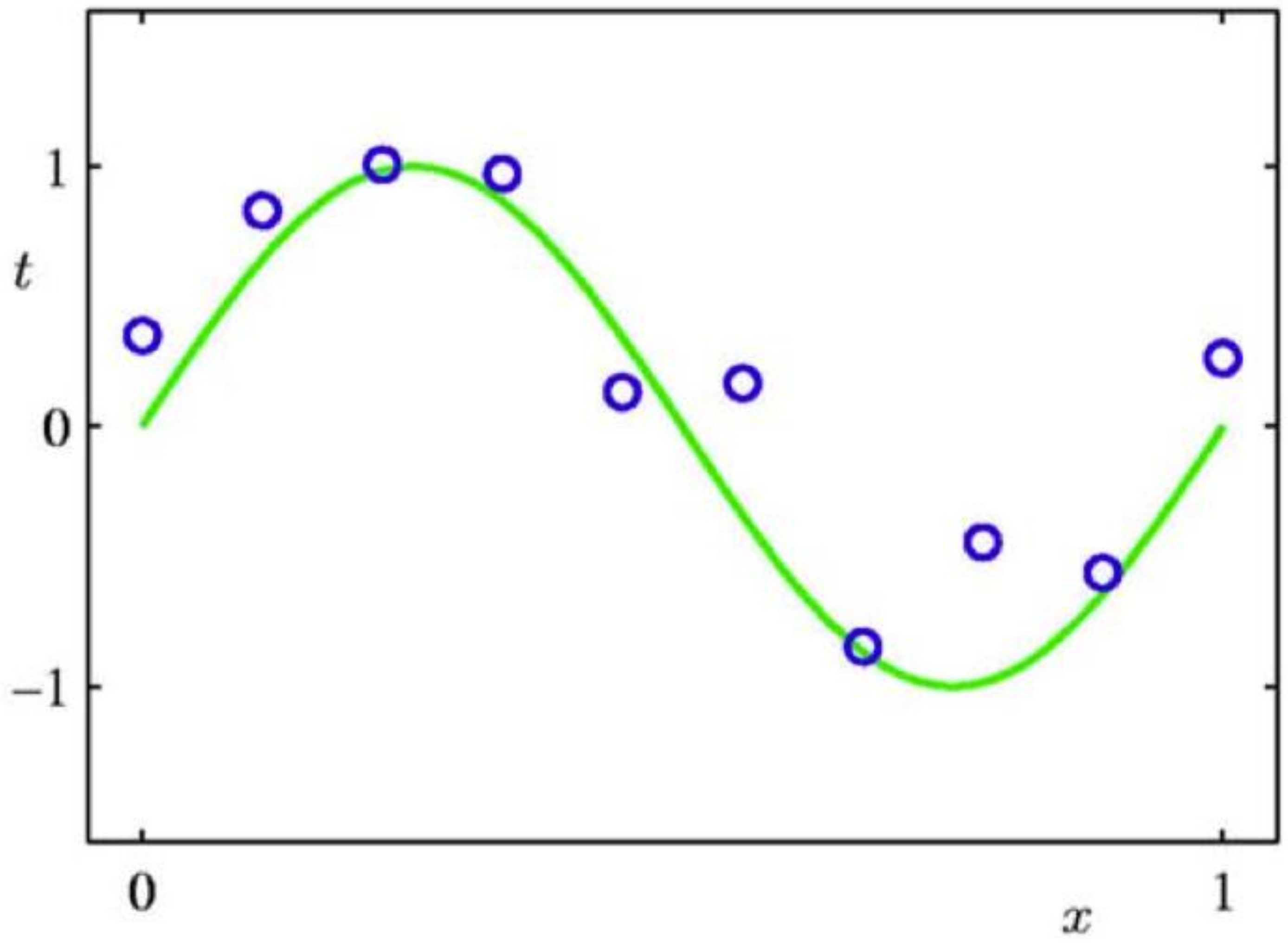
- Overfitting and regularized learning
- Ridge regression
- Lasso regression
- Determining regularization strength

- *Complementary reading: Bishop PRML – Chapter 1, Section 1.1; Chapter 3, Section 3.1 through 3.2.*

Outline

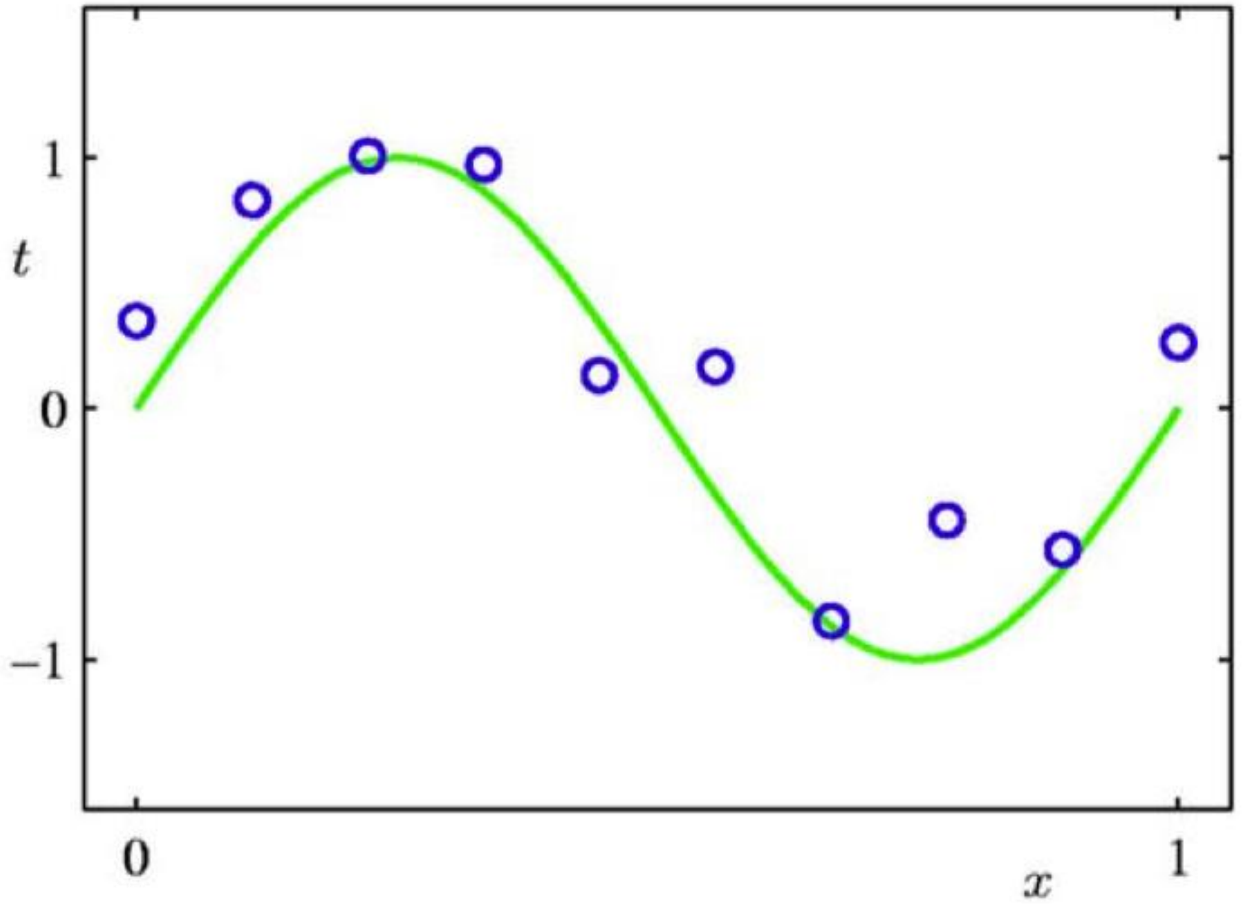
- Overfitting and regularized learning
- Ridge regression
- Lasso regression
- Determining regularization strength

Regression: recap



- Suppose we are given a training set of N observations and D features: $\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\}$, $\mathbf{x}_n \in \mathbb{R}^D$ and $t_n \in \mathbb{R}$
- Assuming $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$
- Regression problem is to estimate $y(\mathbf{x}, \mathbf{w})$ from this data

Regression: recap



- If y is a linear function of $\mathbf{x}$, we have $y(\mathbf{x},\mathbf{w}) = w_0 + w_1x_1 + \cdots + w_Dx_D$
- Or using a basis function ϕ_m, we have:

$$y(\mathbf{x},\mathbf{w}) = \sum_{m=0}^{M-1} w_m\phi_m(\mathbf{x}) \; = \mathbf{w}^{\boldsymbol{T}}\boldsymbol{\phi}(\mathbf{x}), \text{ where } \mathbf{w} \in \mathbb{R}^M$$

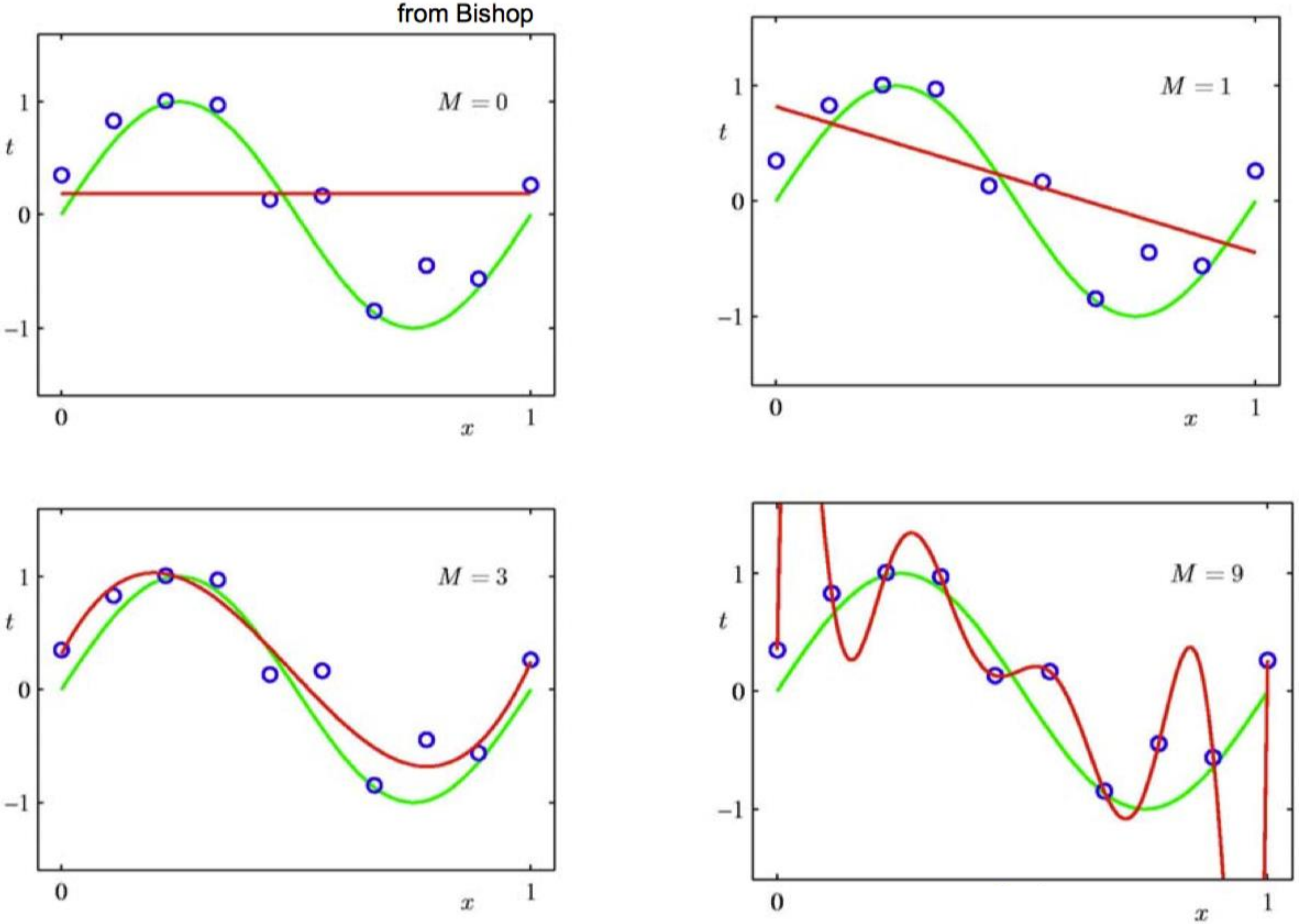
- Example, polynomial basis function of degree $M-1$:

$$y(x,\mathbf{w}) = w_0x^0 + w_1x^1 + \cdots + w_{M-1}x^{M-1}$$

Which one is better?

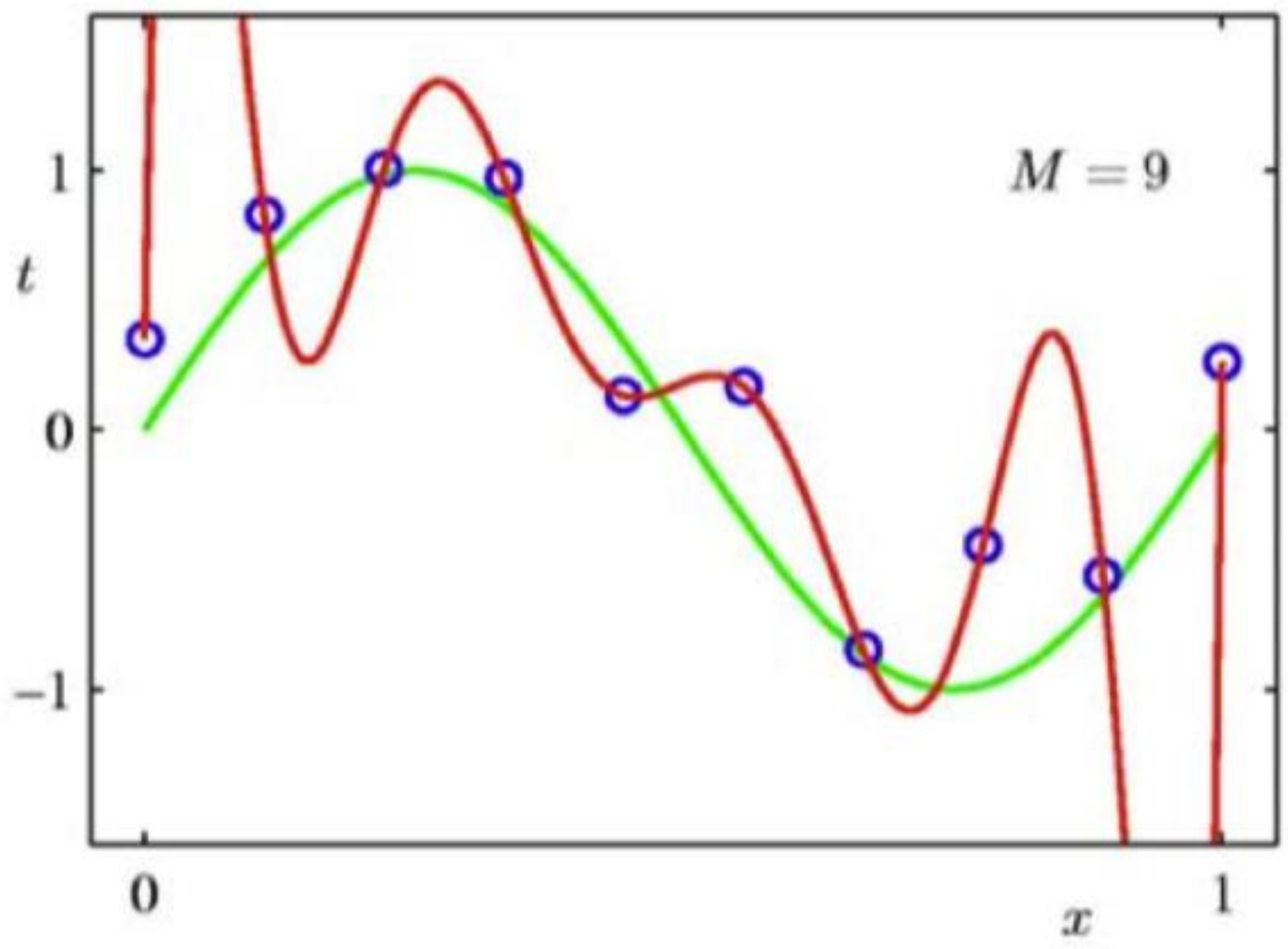
- Can we increase the maximal polynomial degree such that the curve passes through all training points?

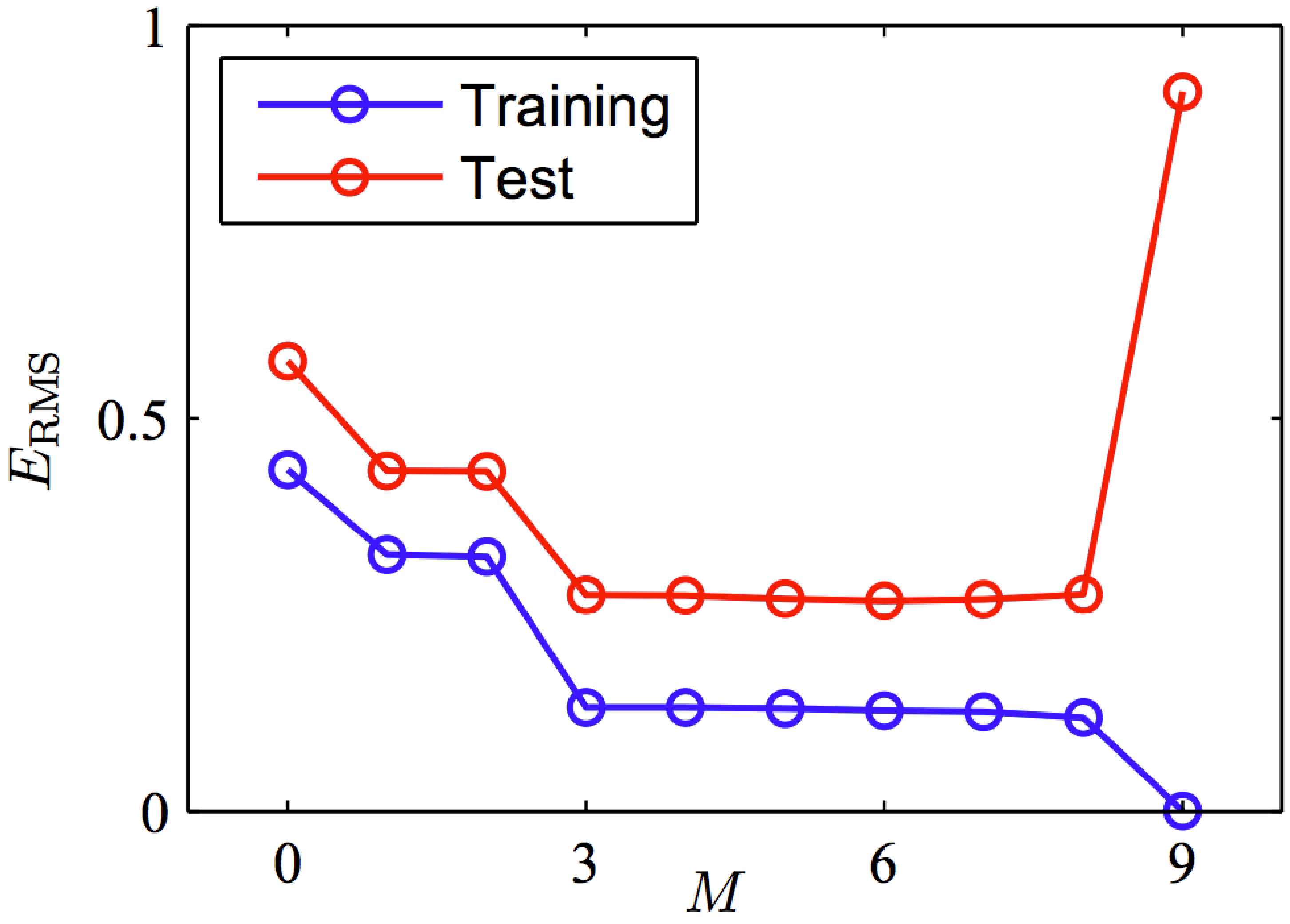
from Bishop



The overfitting problem

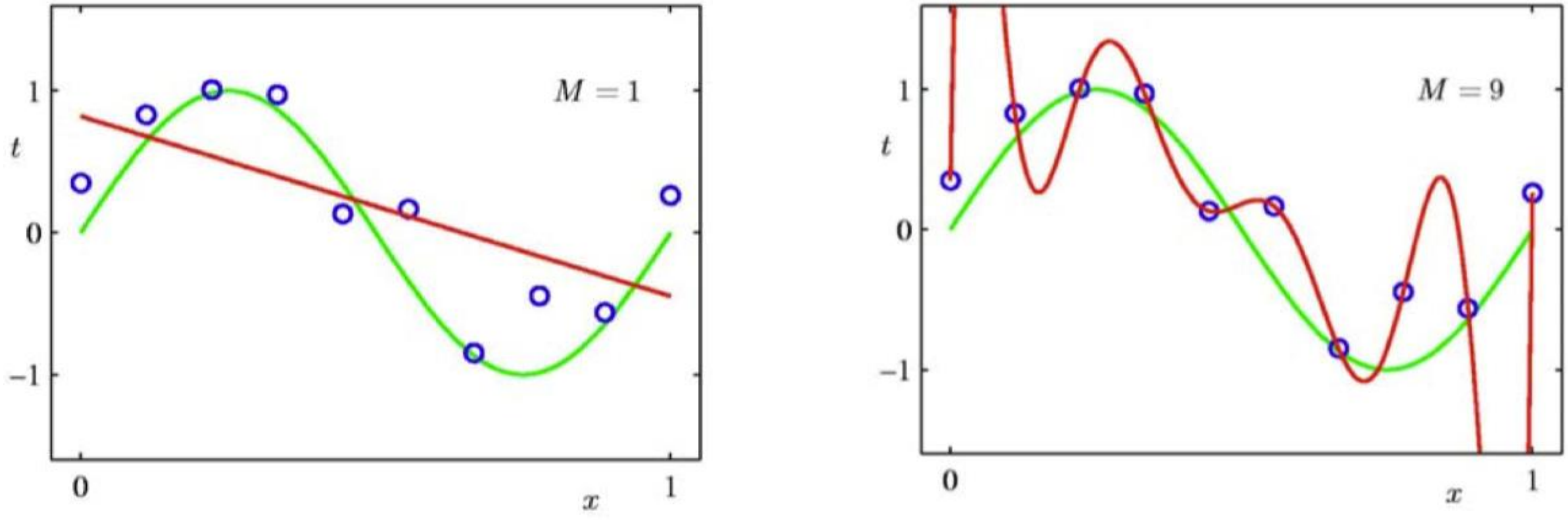
$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}$$





- The training error is very low, but the error on test set is large.
- The model captures not only patterns but also noisy nuisances in the training data.

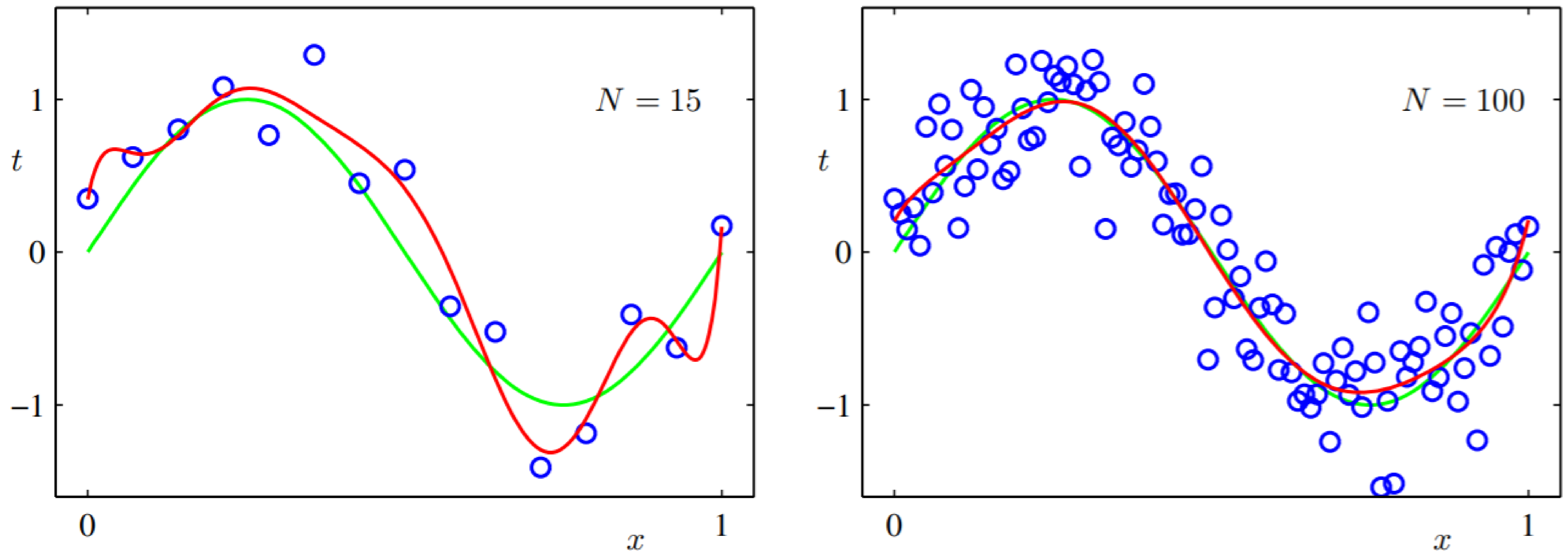
The overfitting problem



$\mathbf{w} = [0.82, -1.27]^T$ $\mathbf{w} = [0.35, 232.37, -5321.83, \dots, 125201.43]^T$

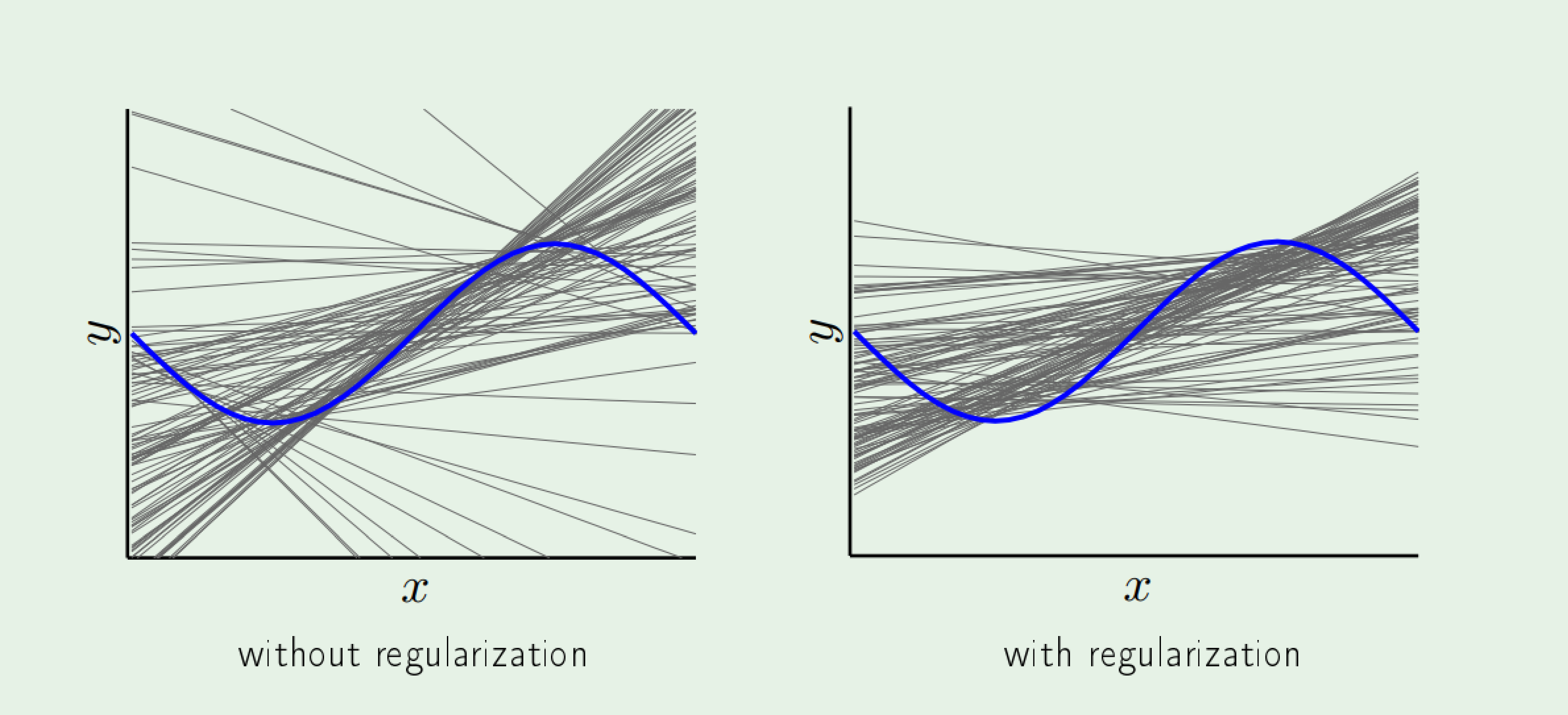
- In regression, overfitting is often associated with large weights (severe oscillation)
- How can we address overfitting?

The overfitting problem



- But what if I don't have more data?

Regularization
(smart way to cure overfitting disease)

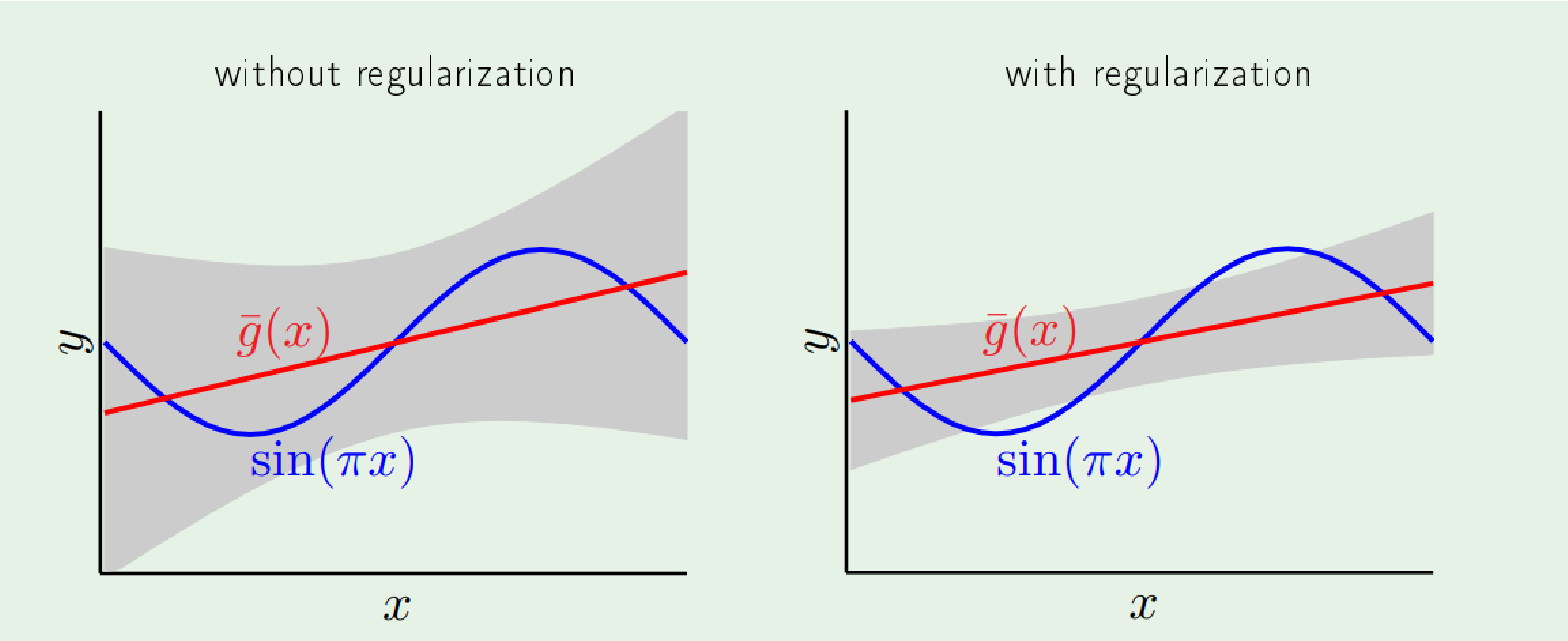


Put a brake on fitting

Fit a linear line on sinusoidal with just two data points

Who is the winner?

$\bar{g}(x)$ = average over all lines



$bias = 0.21; var = 1.69$

$bias = 0.23; var = 0.33$

Converting expressions to linear algebra notation

- Model: $y(\mathbf{x}, \mathbf{w}) = \sum_{m=0}^{M-1} w_m \phi_m(\mathbf{x}) \ = \mathbf{w}^{\boldsymbol{T}} \boldsymbol{\phi}(\mathbf{x})$, where $\mathbf{w} \in \mathbb{R}^M$
- Design matrix for a polynomial of degree $M - 1$:

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}_{N\times M} = \boldsymbol{\Phi} \begin{pmatrix} 1 & x_1 & \cdots & x_1^{M-1} \\ 1 & x_2 & \cdots & x_2^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \cdots & x_N^{M-1} \end{pmatrix}_{N\times M}$$

- Target value and weight vectors:

$$\mathbf{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}_{N\times 1} \quad \text{and } \mathbf{t} = \begin{bmatrix} w_0 \\ \vdots \\ w_{M-1} \end{bmatrix}_{M\times 1}$$

- Sum-of-squares error

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(t_n - \mathbf{w}^{\boldsymbol{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right)^2 = \frac{1}{2} (\mathbf{t} - \boldsymbol{\Phi}\mathbf{w})^T (\mathbf{t} - \boldsymbol{\Phi}\mathbf{w})$$

Regularized learning

- Minimize:

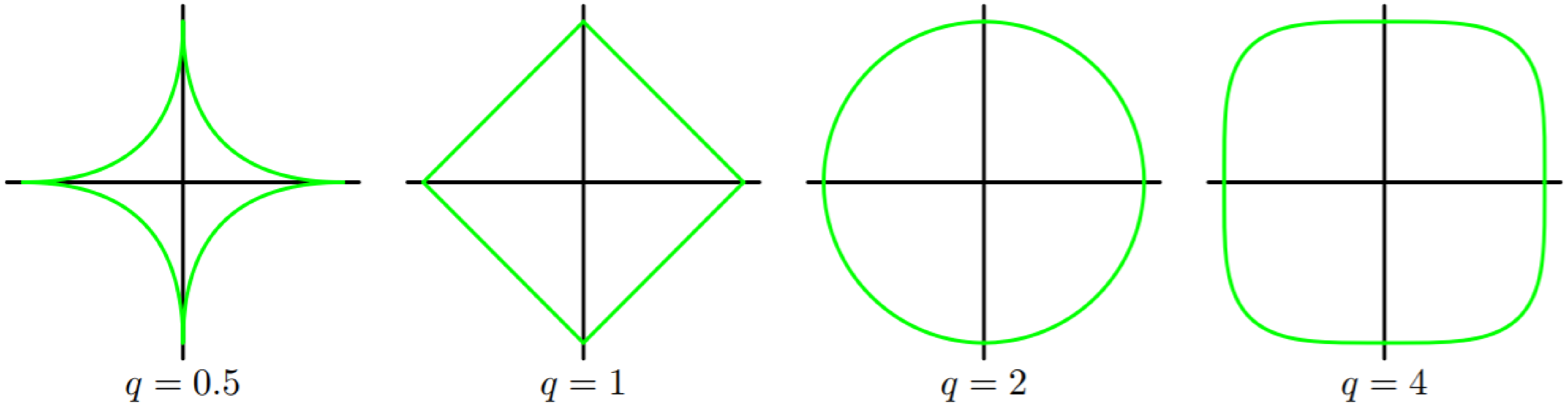
$$\tilde{E}(\mathbf{w}) = E_D(\mathbf{w}) + \lambda E_w(\mathbf{w})$$

- Data-dependent error

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 = \frac{1}{2}(\mathbf{t} - \mathbf{\Phi}\mathbf{w})^T(\mathbf{t} - \mathbf{\Phi}\mathbf{w})$$

- Regularization term

$$E_w(\mathbf{w}) = \sum_{j=1}^{M}\left|w_j\right|^q$$



Regularized learning

- Ridge regression ($q = 2$):

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^{\boldsymbol{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 + \frac{\lambda}{2}\mathbf{w}^{\boldsymbol{T}}\mathbf{w}$$

- Lasso regression ($q = 1$):

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^{\boldsymbol{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 + \lambda\sum_{j=1}^{M}\left|w_j\right|$$

Outline

- Overfitting and regularized learning
- Ridge regression
- Lasso regression
- Determining regularization strength

Regularized learning

- Minimize

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^{\boldsymbol{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 \quad + \quad \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

$$\Downarrow$$

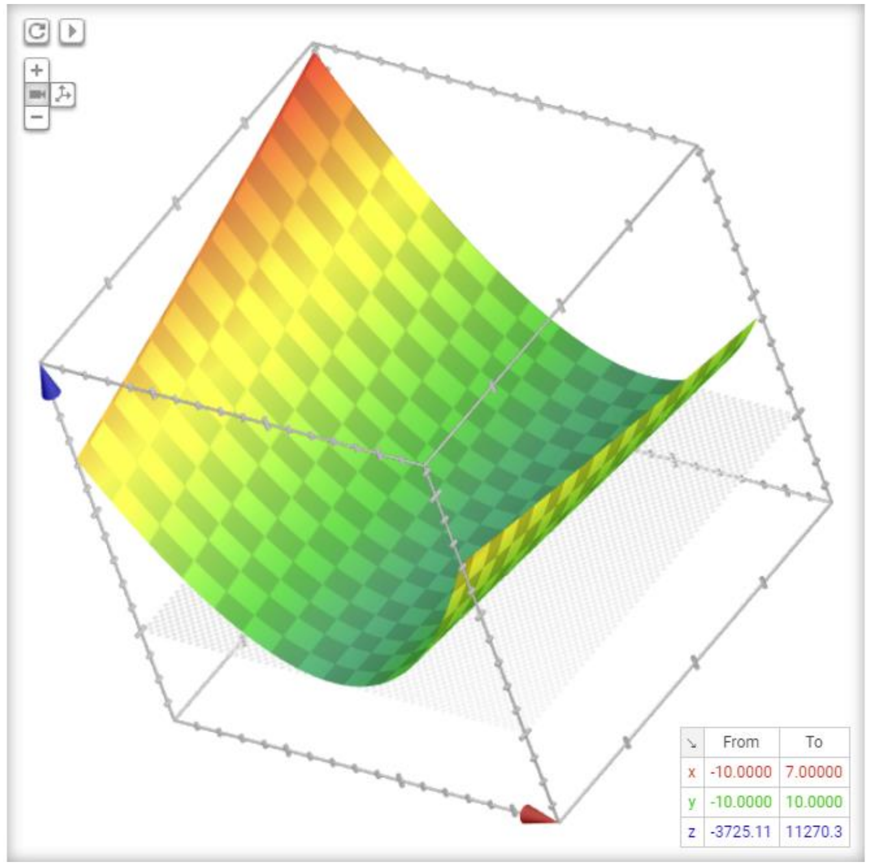
$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{t} - \mathbf{\Phi}\mathbf{w})^T(\mathbf{t} - \mathbf{\Phi}\mathbf{w}) \quad + \quad \frac{\lambda}{2}\mathbf{w}^{\boldsymbol{T}}\mathbf{w}$$

Example: regularized learning

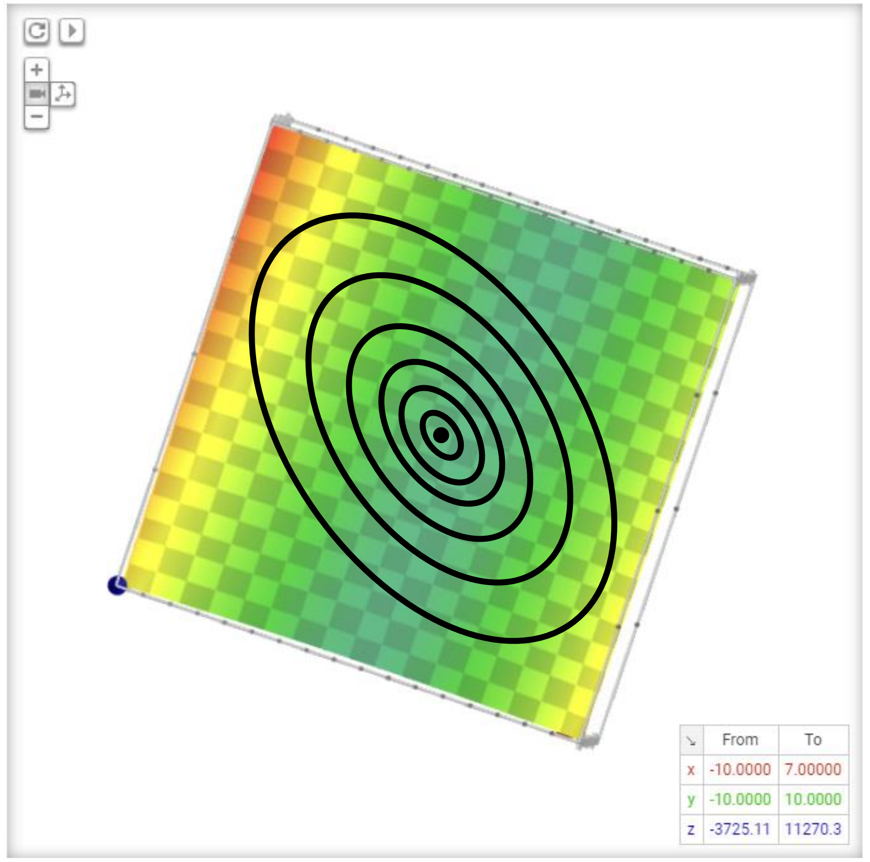
- Given a dataset $\{(x_1, t_1\ ), (x_2, t_2), \dots, (x_N, t_N)\}$ (one single feature)
- We fit a polynomial of degree 1 (line) on the data: $y(x, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(x_n) = w_0 + w_1 x_n$
- We can then write the data-dependent error (in this case the mean of the square error)

$$E_D(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 = \frac{1}{N}\sum_{n=1}^{N}(t_n - w_0 + w_1 x_n)^2$$

and plot it with respect to the two parameters w_0 and w_1:



3D view



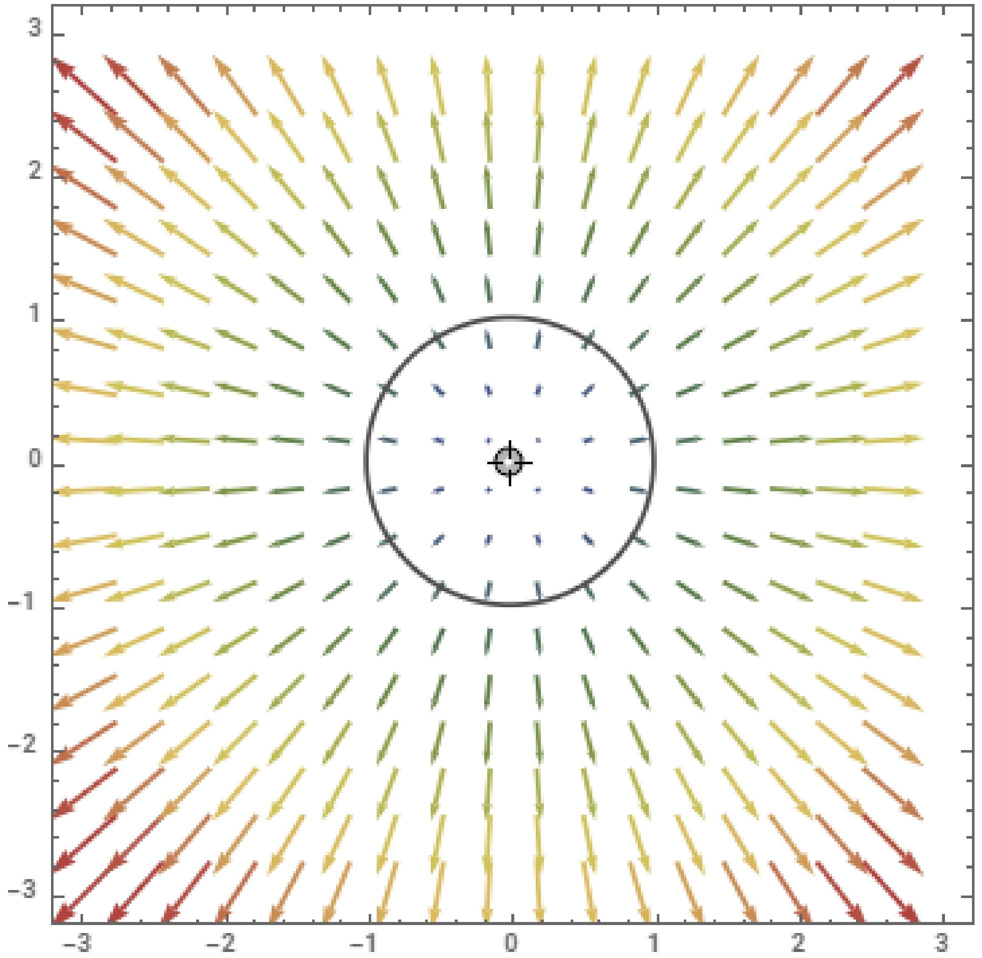
Top view

Example: regularized learning

- Let us plot the gradient of $\mathbf{w}^T\mathbf{w} = [w_0 \quad w_1]\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = w_0^2 + w_1^2$
- If you imagine standing at a point $[w_0, w_1]^T$, $\nabla(\mathbf{w}^T\mathbf{w})$ tells you which direction you should travel to increase the value of $\mathbf{w}^T\mathbf{w}$ most rapidly:

$$\nabla(\mathbf{w}^T\mathbf{w}) = \begin{bmatrix} \dfrac{\partial}{\partial(w_0)}\mathbf{w}^T\mathbf{w} \\ \dfrac{\partial}{\partial(w_1)}\mathbf{w}^T\mathbf{w} \end{bmatrix} = \begin{bmatrix} 2w_0 \\ 2w_1 \end{bmatrix} \approx \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

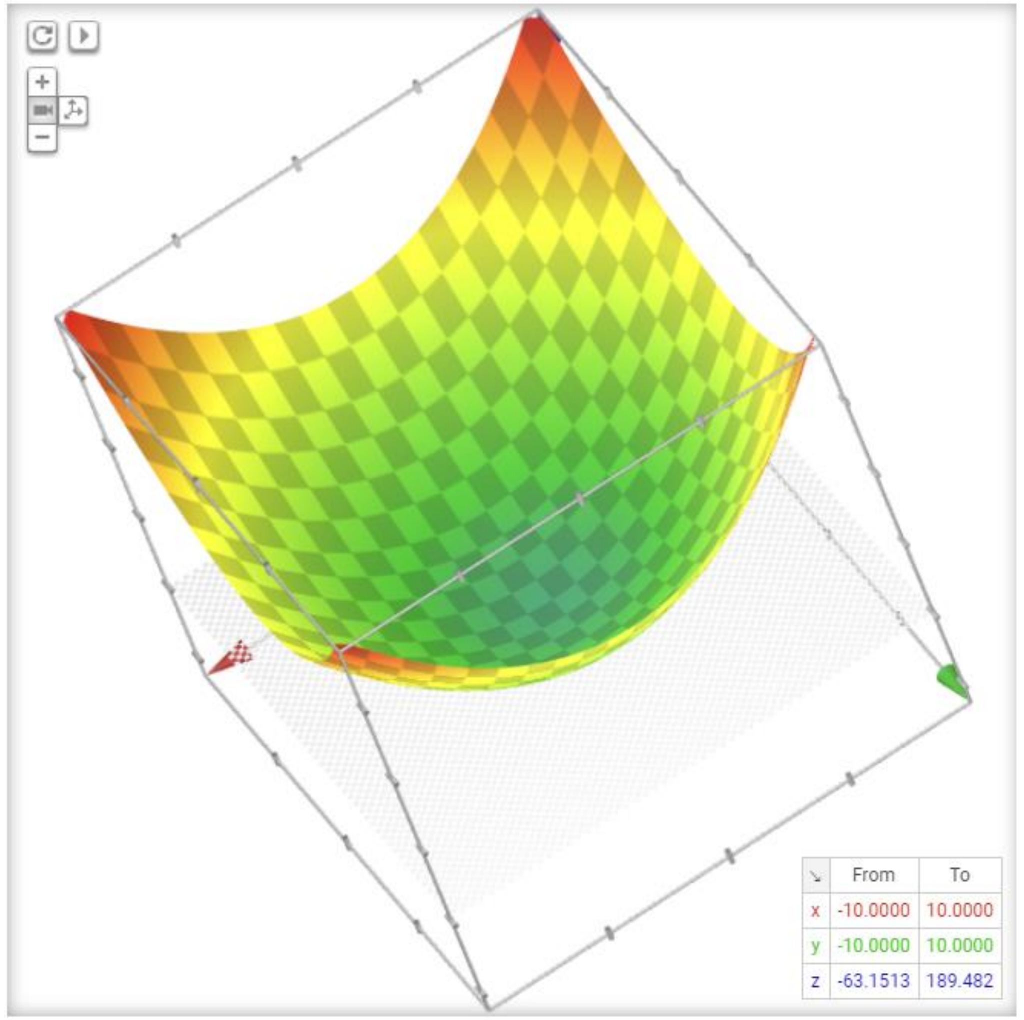
$\nabla(\mathbf{w}^T\mathbf{w})$ is a vector field



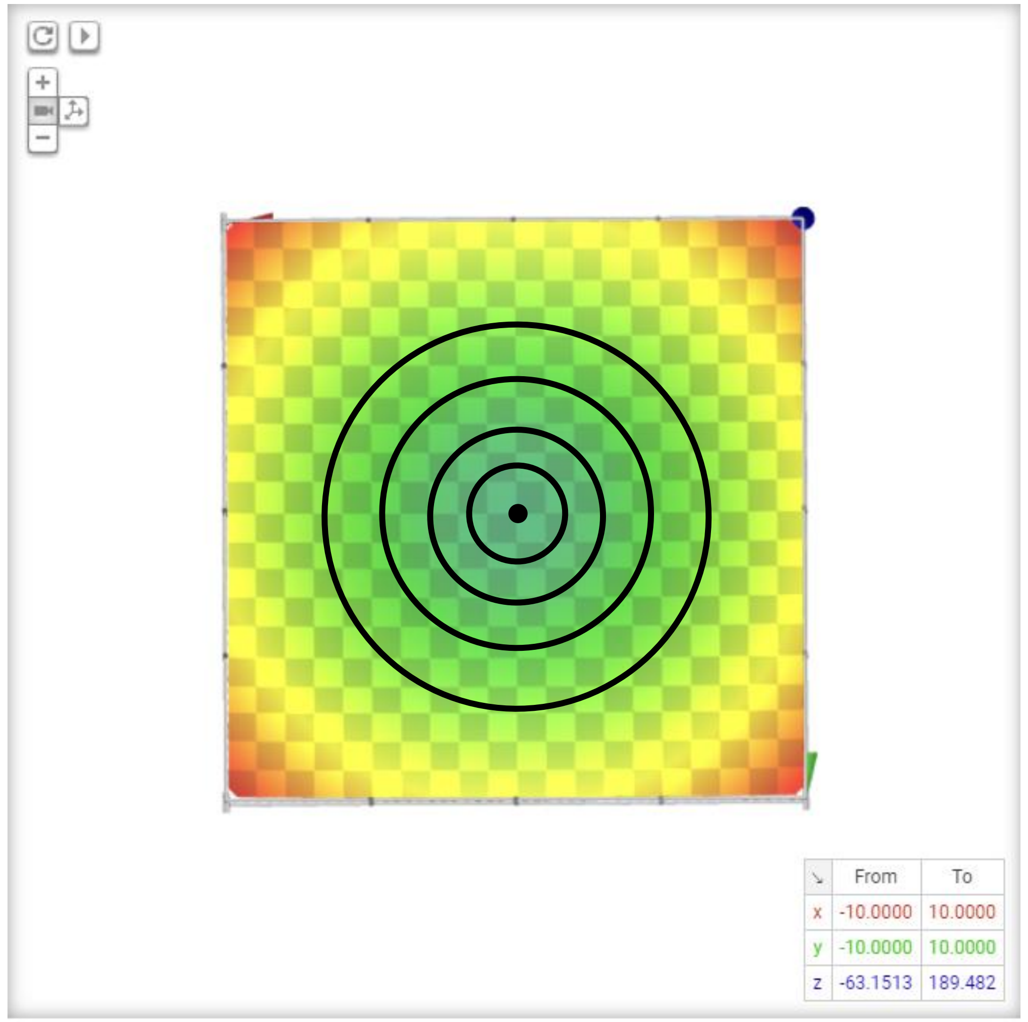
- Any line passing through the center of the circle → most rapid increase

Example: regularized learning

- Plotting the regularization portion: $E_{\mathbf{w}}(\mathbf{w}) = \mathbf{w}^T\mathbf{w} = [w_0 \quad w_1]\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = w_0^2 + w_1^2$ with respect to the two parameters w_0 and w_1:



3D view



Top view

Regularized learning: constrained optimization

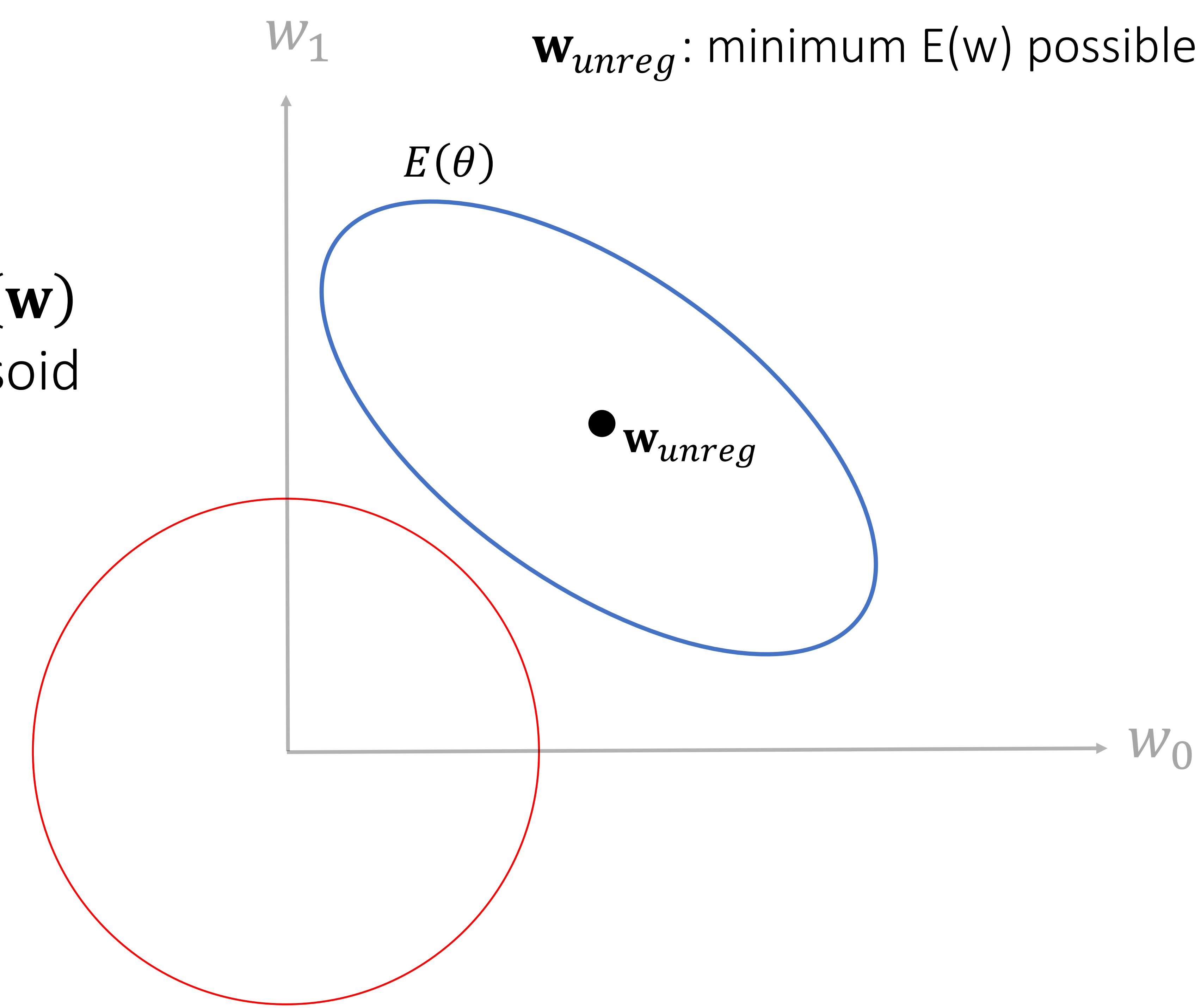
- We can treat this as an optimization problem, where we are trying to minimize:

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{t} - \boldsymbol{\Phi}\mathbf{w})^T(\mathbf{t} - \boldsymbol{\Phi}\mathbf{w})$$

$$subject\ to\ \mathbf{w}^T\mathbf{w} \leq \mathrm{C}$$

- Find a solution in $\mathbf{w}^T\mathbf{w}$ that minimizes $E(\mathbf{w})$ which is constant on the surface of the ellipsoid

$\mathbf{w}_{unreg}$: minimum E(w) possible



Regularized learning: constrained optimization

- Considering the $E(\mathbf{w})$ and C what is a $\mathbf{w}$ candidate?

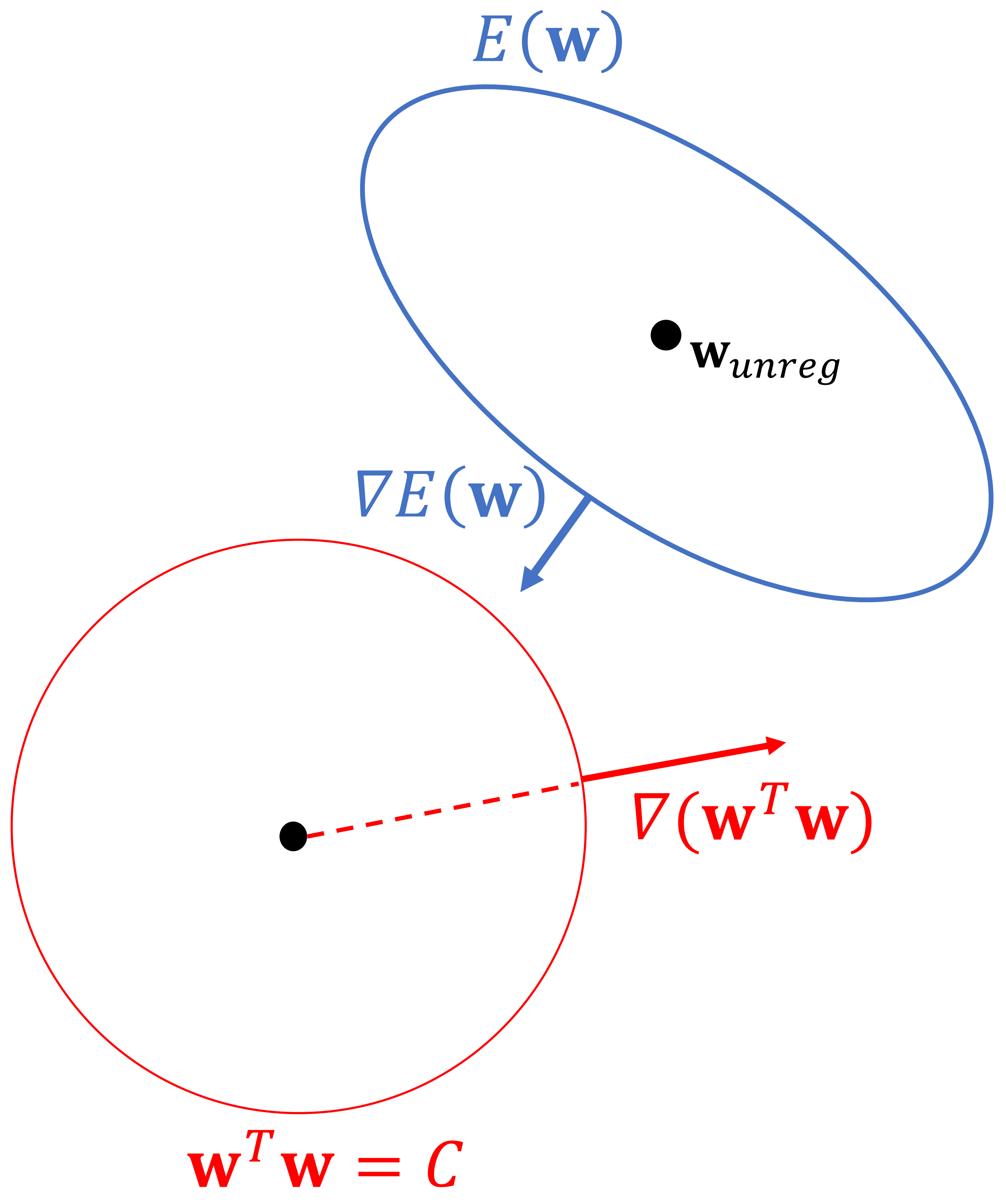
- The gradient $\nabla E(\mathbf{w})$ in objective function which minimizes error (orthogonal to ellipse). Changes happen in orthogonal direction

- **What is the orthogonal direction on the other surface?**

It is a $\mathbf{w}$ along a direction passing through center of the circle

- **Applying the $\mathbf{w}^T\mathbf{w}$, where is the best solution located?**

On the boundary of the circle, as it is the closest one to the minimum absolute



Regularized learning: constrained optimization

- At the solution point we have:

$$\nabla E(\mathbf{w}) \propto -\nabla(\mathbf{w}^T\mathbf{w})$$

- To obtain equality we multiply the gradient by a constant

$$\nabla E(\mathbf{w}) = -\lambda\mathbf{w}$$

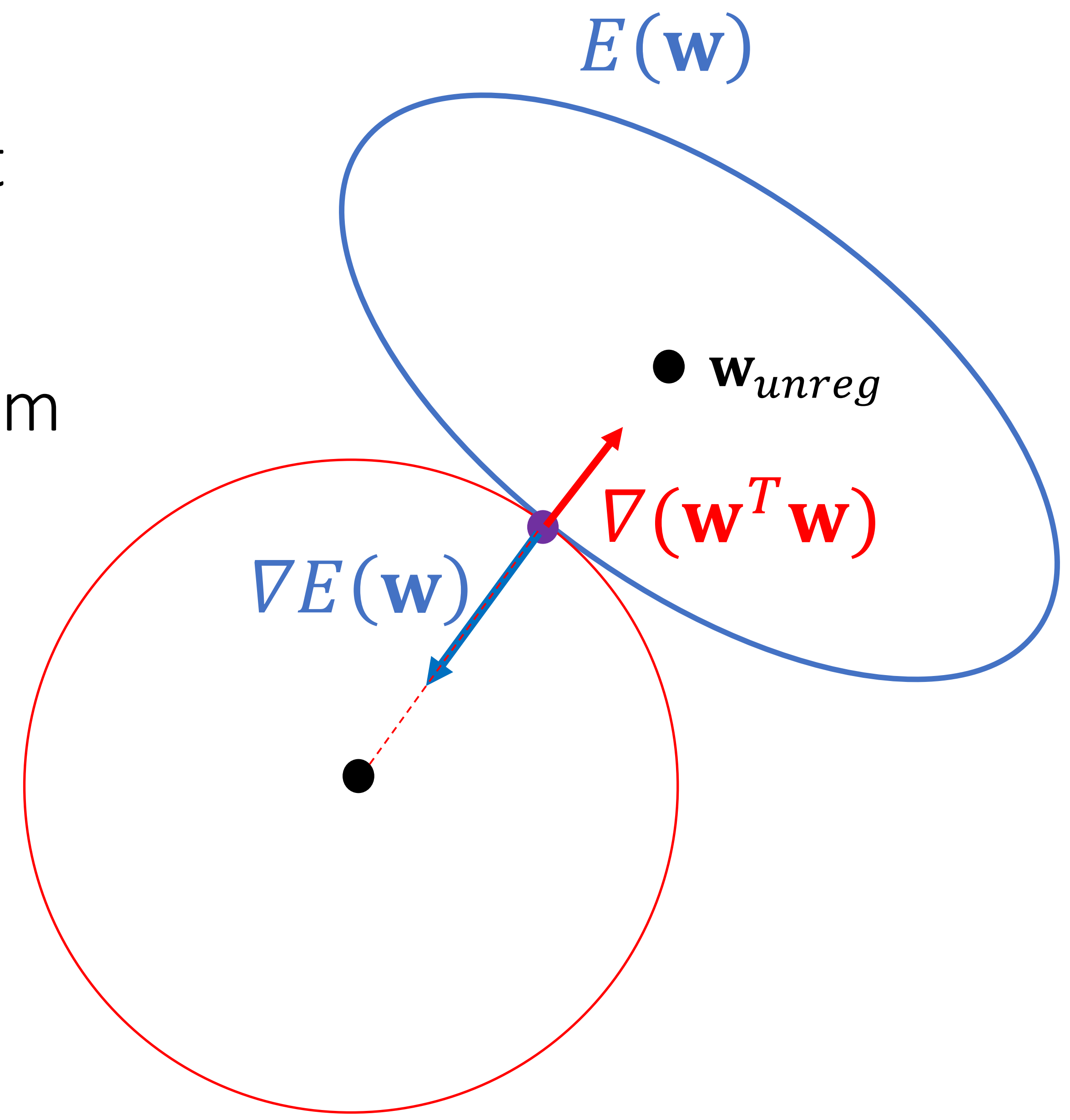
- We then recover our previous expression for the minimum

$$\nabla E(\mathbf{w}) + \lambda\mathbf{w} = 0$$

- And we can rewrite the optimization problem as calculating the minimum of:

$$\tilde{E} = \frac{1}{2}(\mathbf{t} - \mathbf{\Phi}\mathbf{w})^T(\mathbf{t} - \mathbf{\Phi}\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

$$C \uparrow \quad \lambda \downarrow$$



Ridge regression

- Minimize

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}(\mathbf{t} - \mathbf{\Phi w})^T(\mathbf{t} - \mathbf{\Phi w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- Compute the derivative with respect to w and set it to zero

$$\frac{\partial \tilde{E}(\mathbf{w})}{\partial \mathbf{w}} = -\mathbf{\Phi}^T(\mathbf{t} - \mathbf{\Phi w}) + \lambda\mathbf{w} = 0$$

$$-\mathbf{\Phi}^T\mathbf{t} + \mathbf{\Phi}^T\mathbf{\Phi w} + \lambda\mathbf{I w} = \mathbf{0}$$

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

Ridge regression

- Closed form solution

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

This shows that there is a unique solution.

- If $\lambda = 0$ (no regularization), then:

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

where $\mathbf{\Phi}^{+}$ is the pseudo-inverse of $\mathbf{\Phi}$

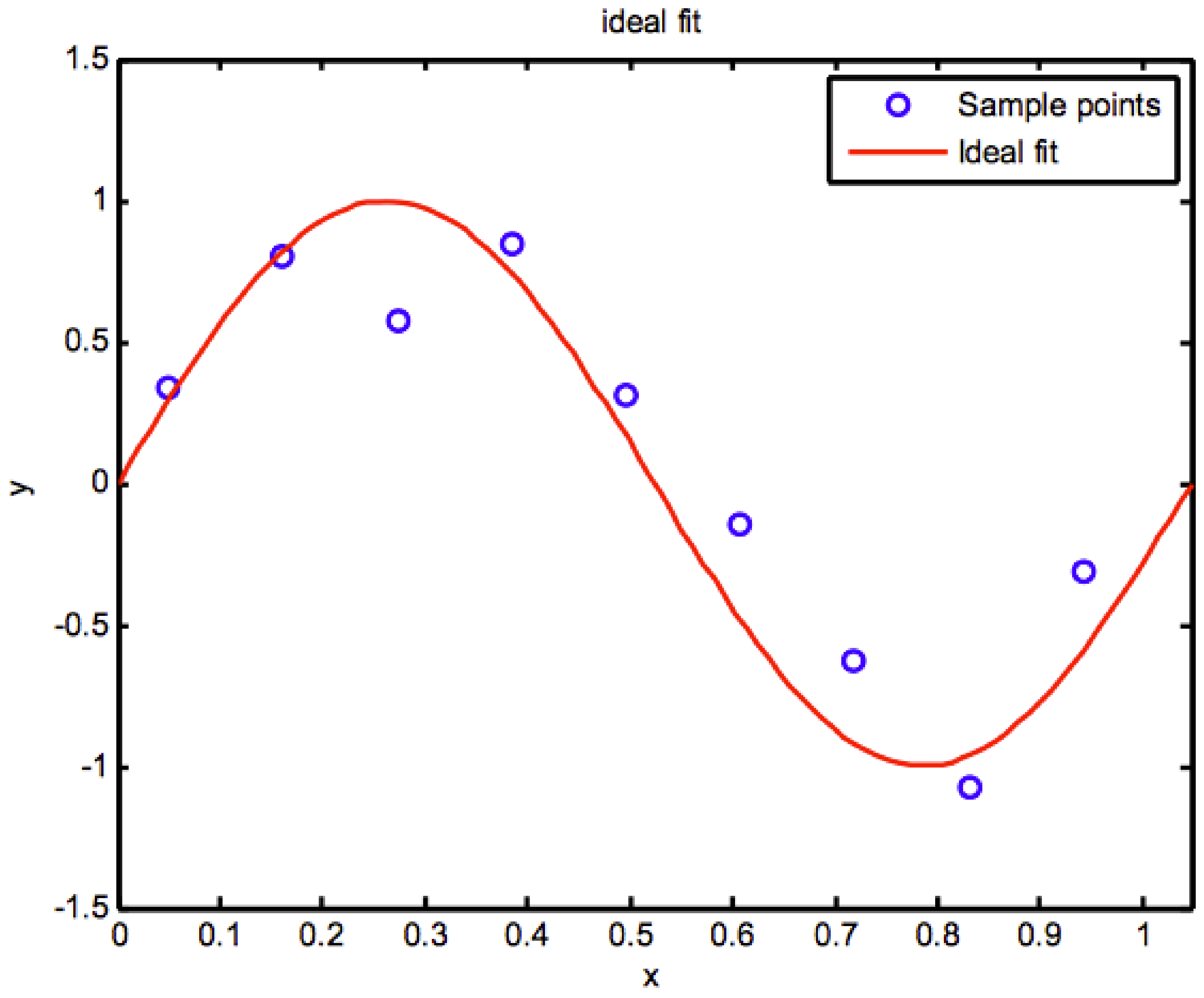
- Adding the term $\lambda\mathbf{I}$ improves the conditioning of the inverse, since if $\mathbf{\Phi}$ is not full rank, then $(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I})$ will be (provided λ is sufficiently large)
- As $\lambda \to \infty$, $\mathbf{w} \to \frac{1}{\lambda}\mathbf{\Phi}^T\mathbf{t} \to \mathbf{0}$

Ridge regression: example

- The red curve is the true function, which is not polynomial.
- The data points are samples from the curve with added noise
- There is a choice in both the degree, M, of the basis functions used and the strength of the regularization

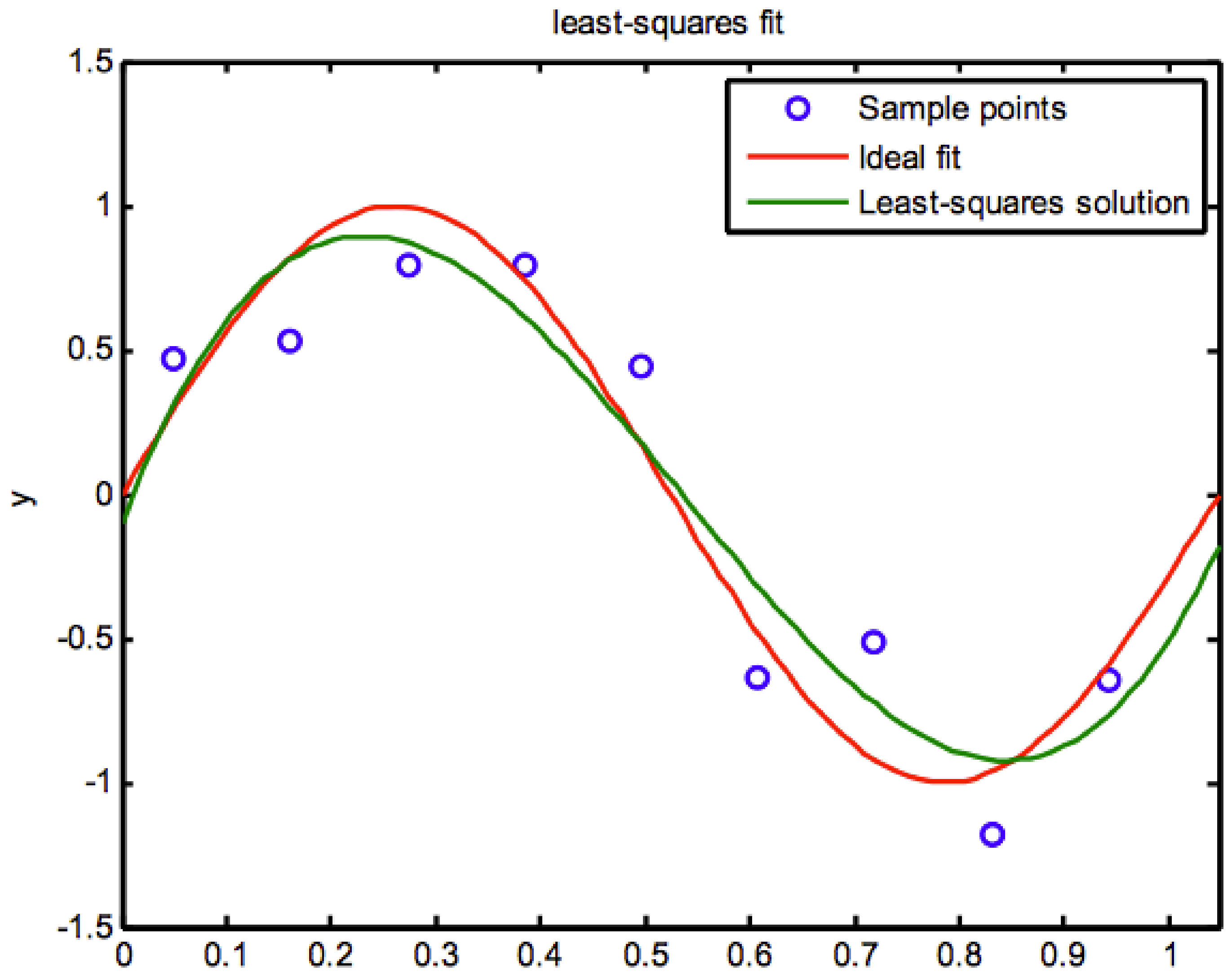
$$y(\mathrm{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathrm{x})$$

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n) \right)^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

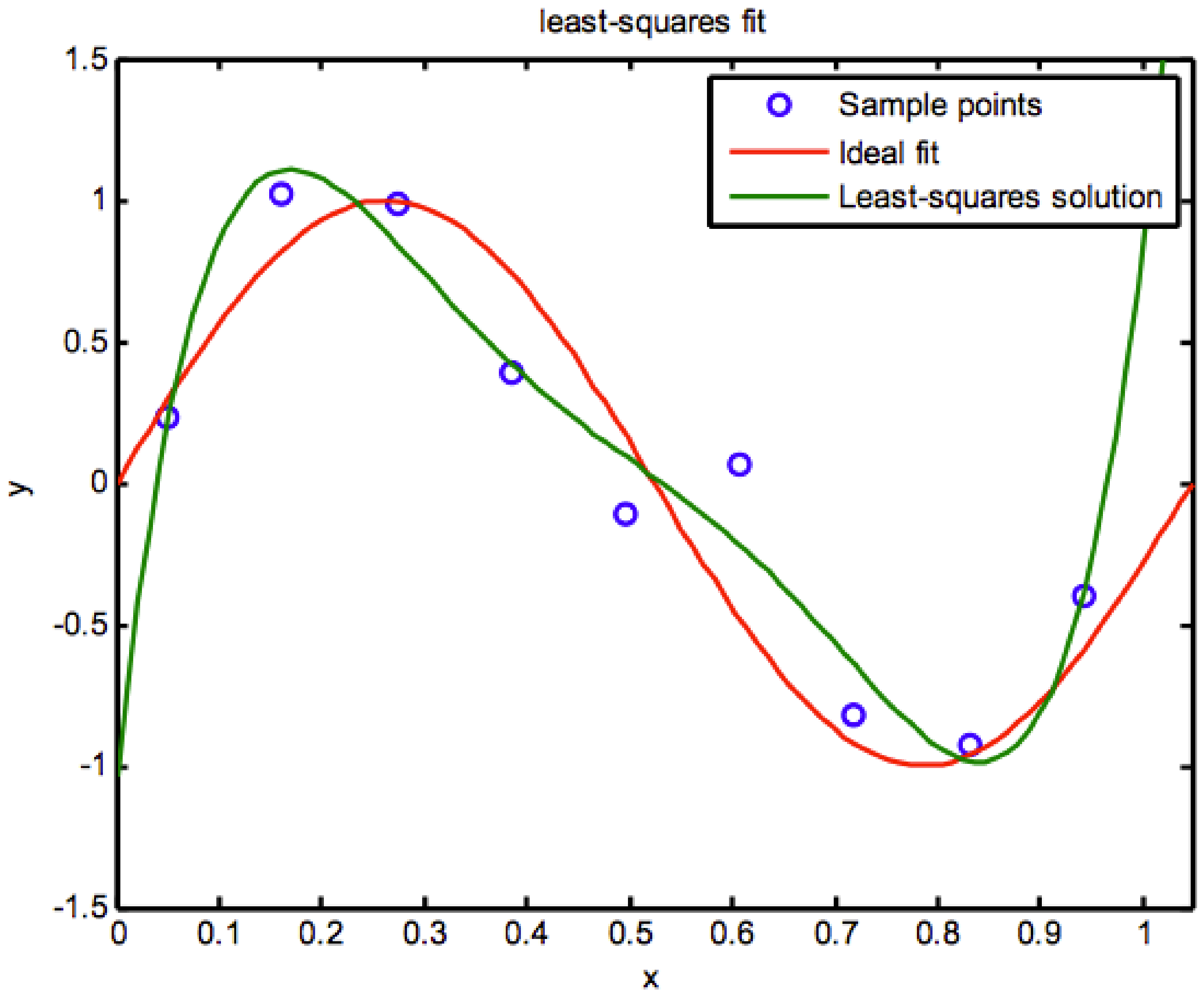


Ridge regression: example

$M-1\ =3$ (polynomial of degree 3)



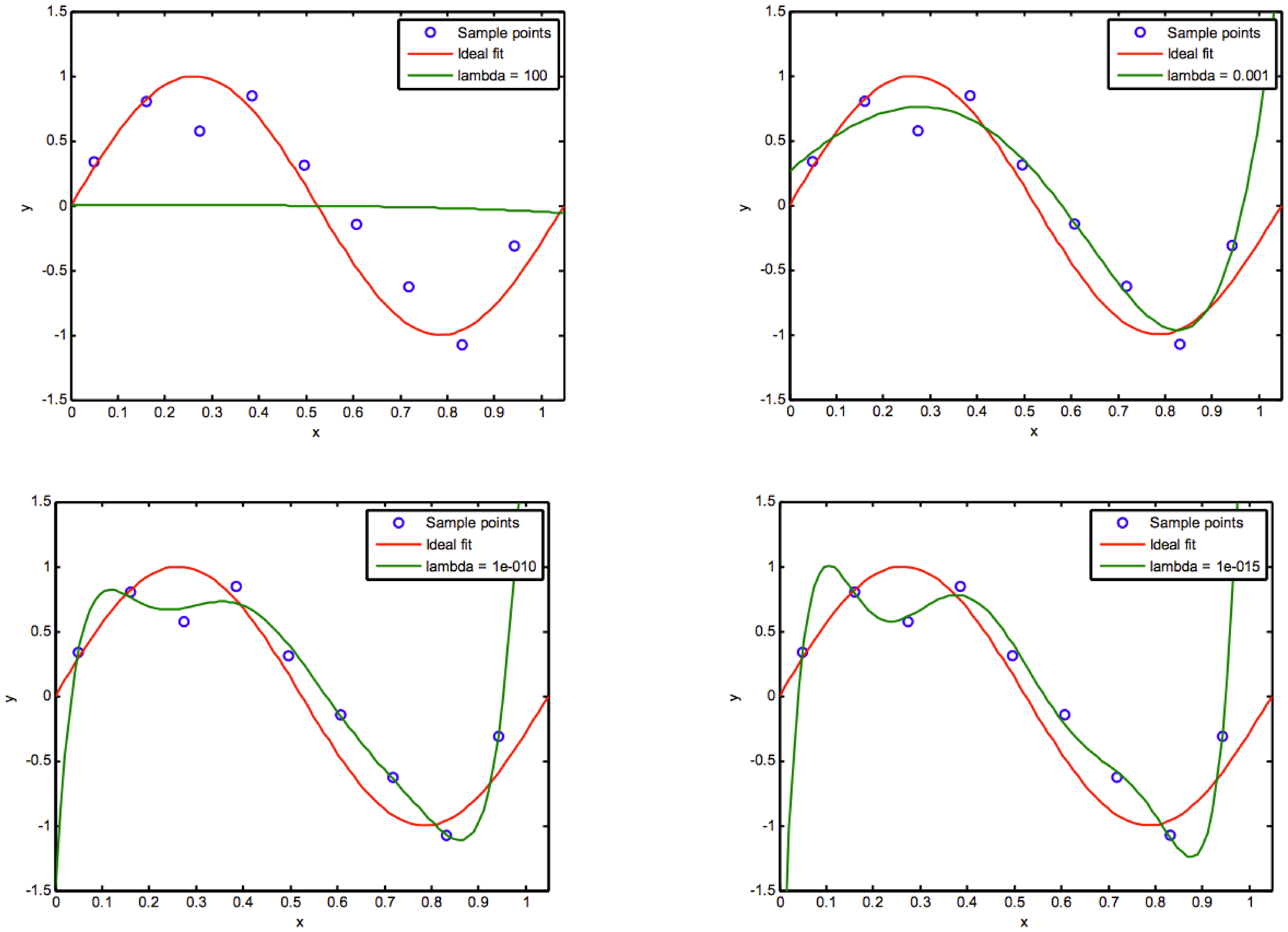
$M-1\ =5$ (polynomial of degree 5)



Example

$N = 9$ datapoints

$M - 1 = 7$



Outline

- Overfitting and regularized learning
- Ridge regression
- Lasso regression
- Determining regularization strength

Lasso regularization

- *LASSO = Least absolute shrinkage and selection*
- Minimize with respect to $\mathbf{w} \in \mathbb{R}^M$

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^{\boldsymbol{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 \quad + \quad \lambda\sum_{m=0}^{M-1}|w_m|$$

- This is a quadratic optimization problem
- There is a unique solution

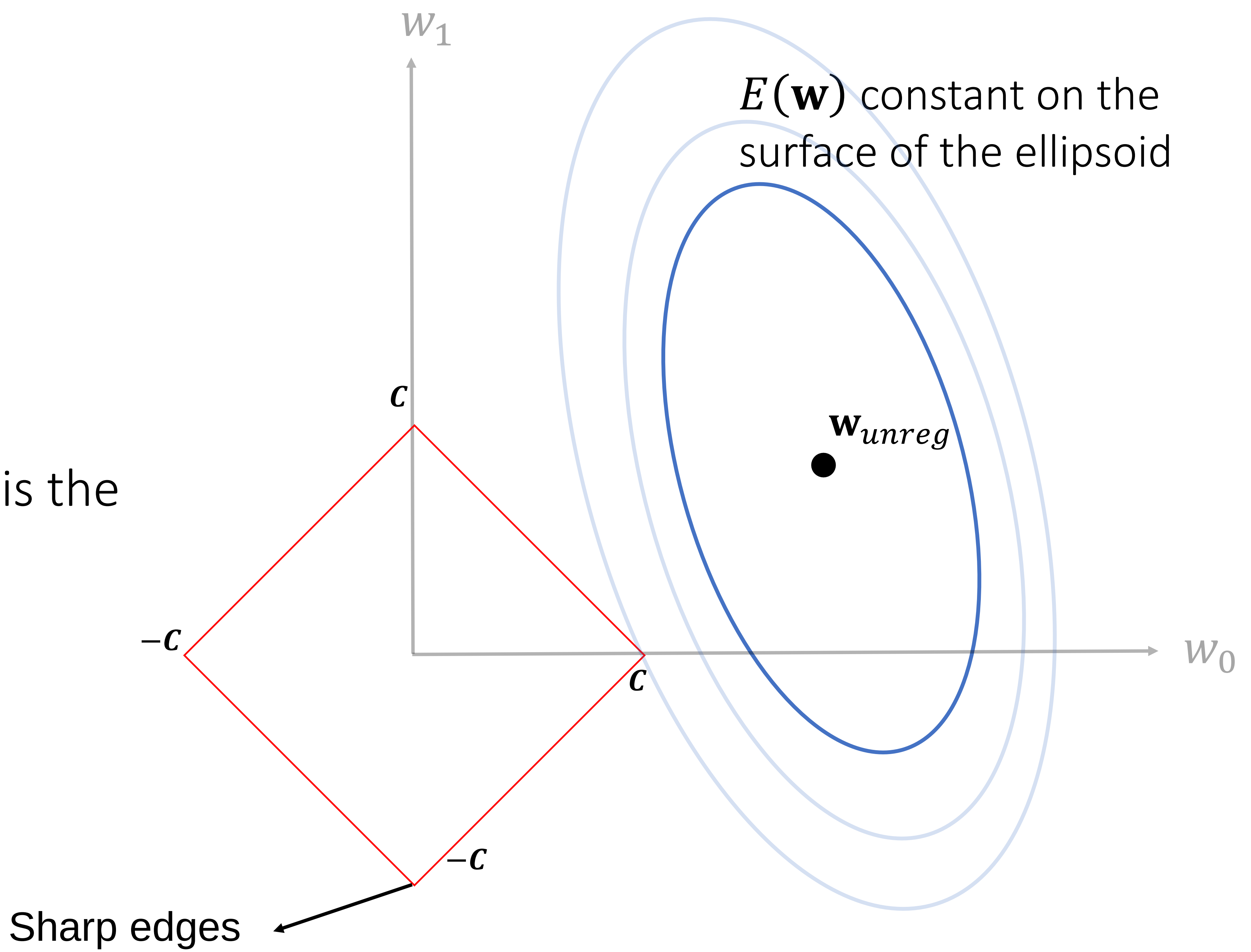
Lasso regression

- Minimize:

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{t} - \mathbf{\Phi w})^T(\mathbf{t} - \mathbf{\Phi w})$$

$$subject\ to \sum_{m=0}^{M-1} |w_m| \leq C$$

- One advantage of Lasso regression is the potential to obtain sparse solutions
- Application to feature selection



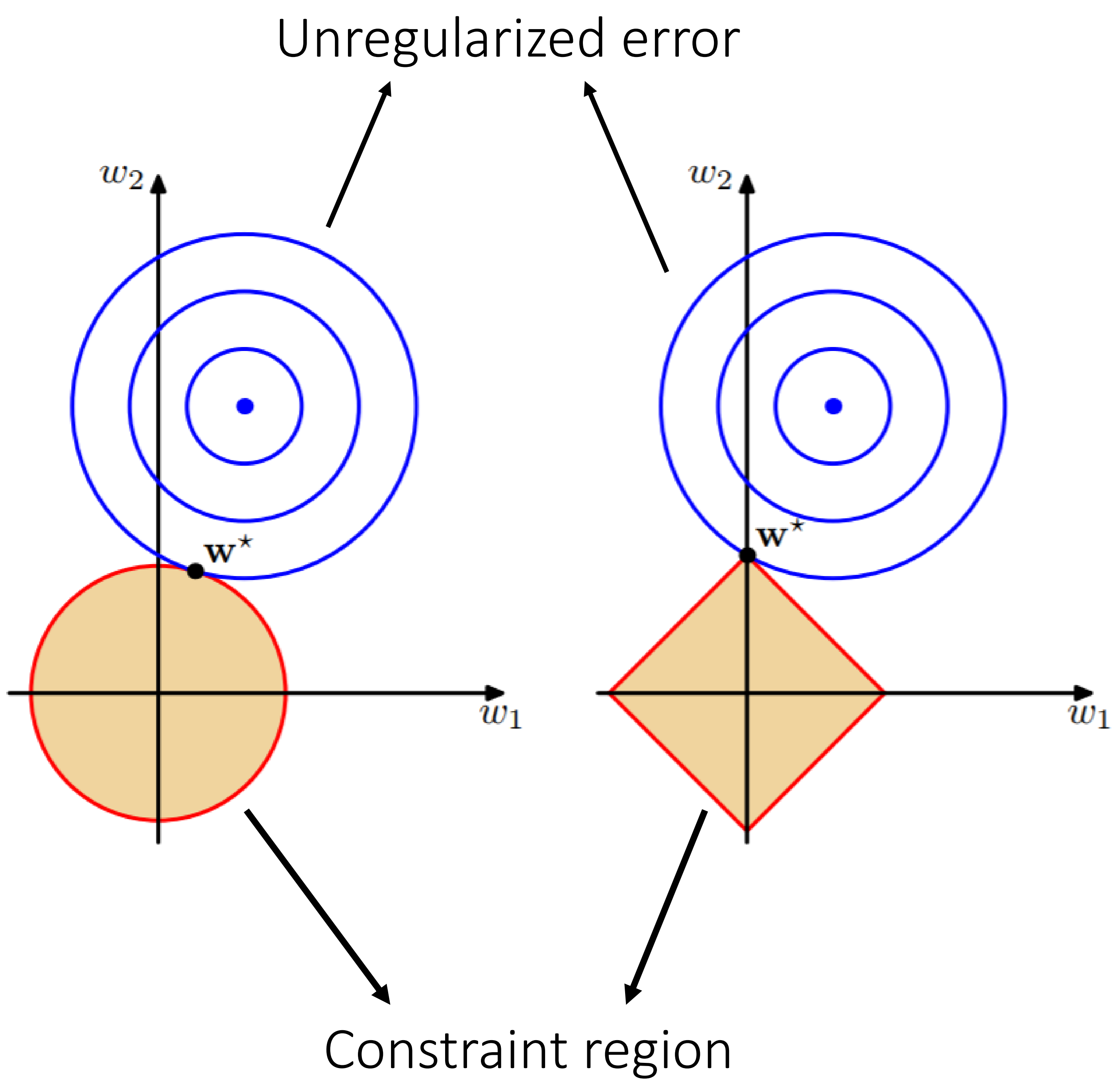
Regularized learning

- Ridge regression ($q = 2$):

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

- Lasso regression ($q = 1$):

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 + \lambda\sum_{j=1}^{M}|w_j|$$

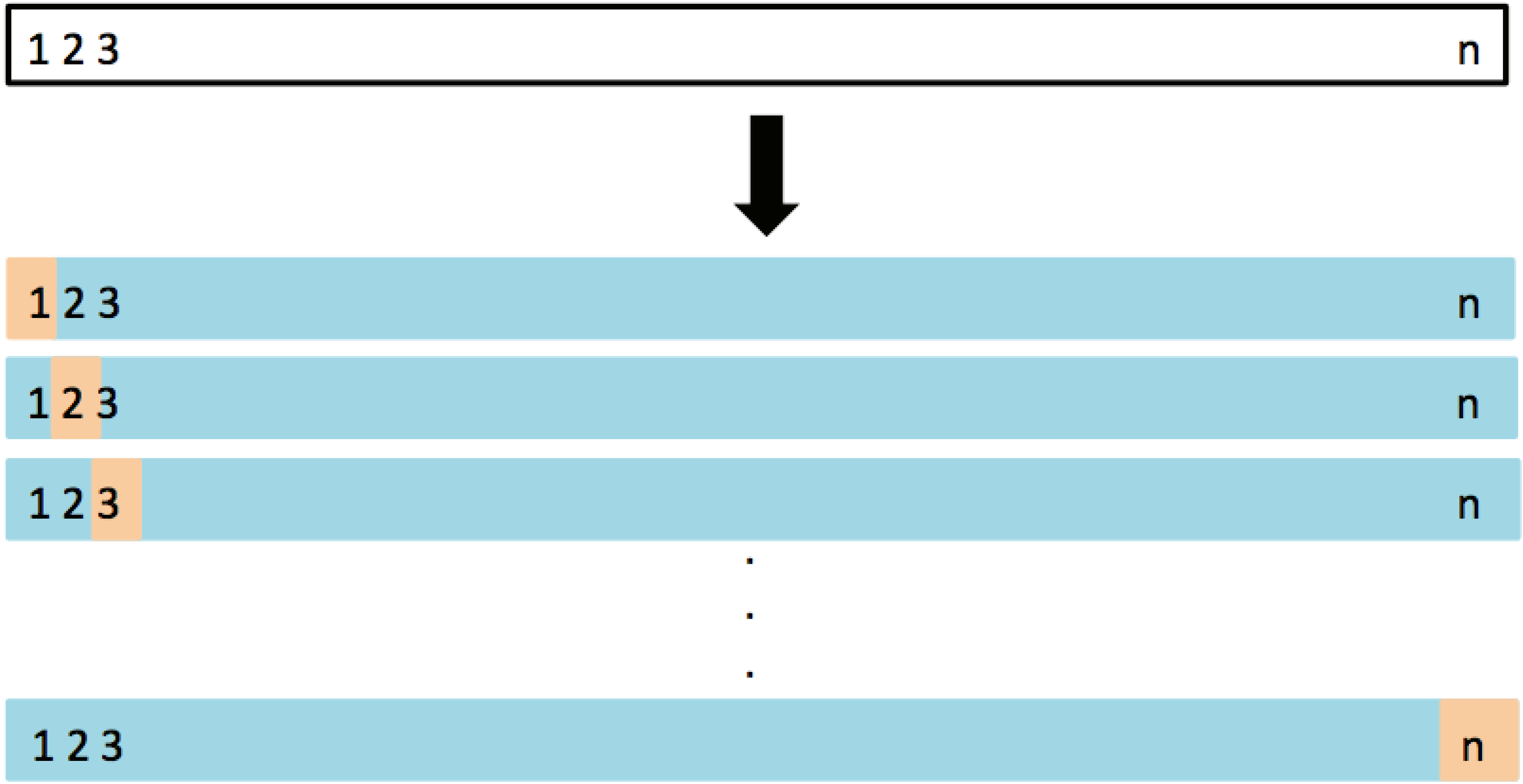


Outline

- Overfitting and regularized learning
- Ridge regression
- Lasso regression
- Determining regularization strength

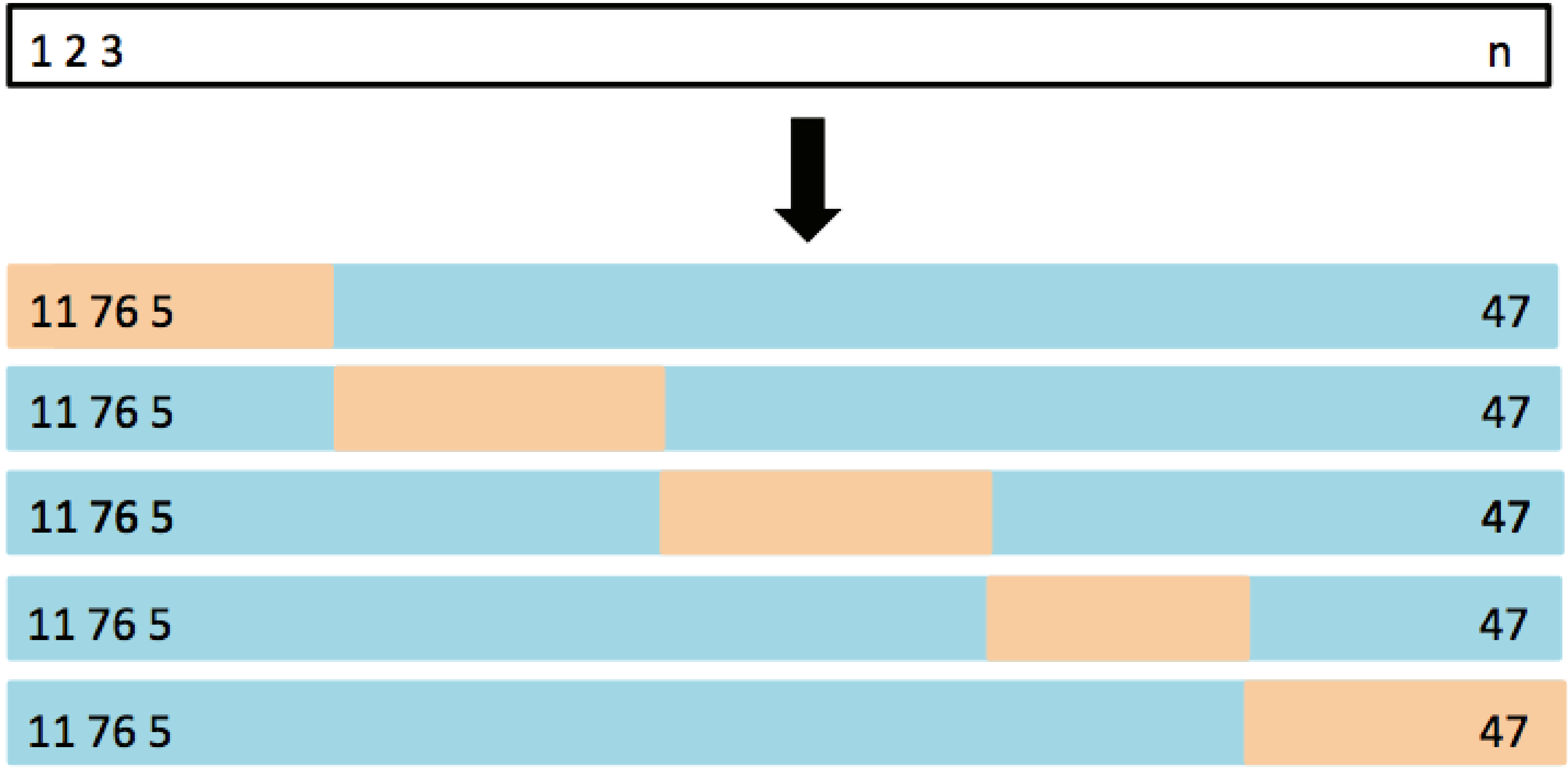
Leave-one-out cross-validation

- For every $\mathrm{n} = 1, \ldots, N$:
  - Train the model on every datapoint except the n-th
  - Compute the test error on the held-out point
- Average the test errors

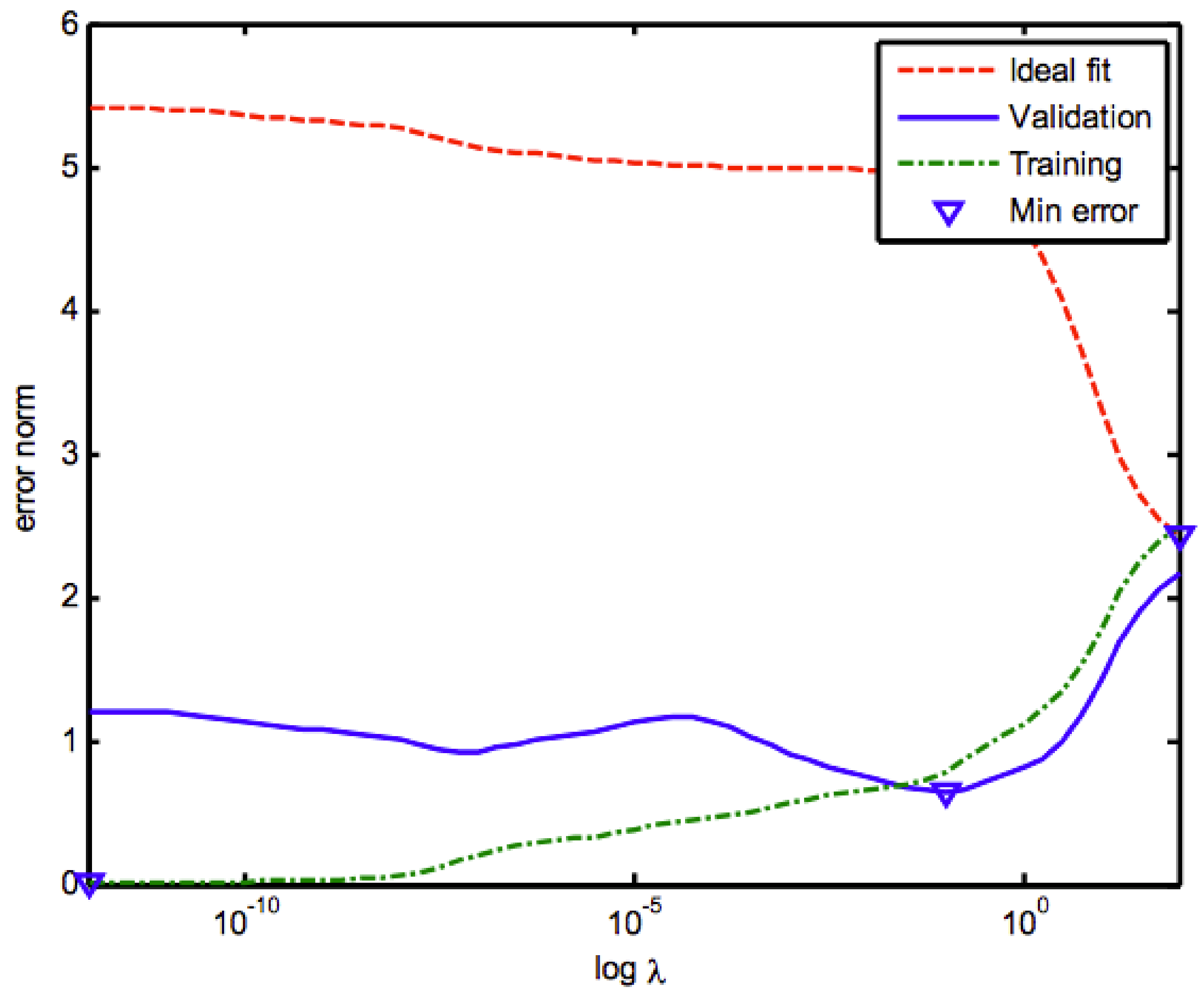


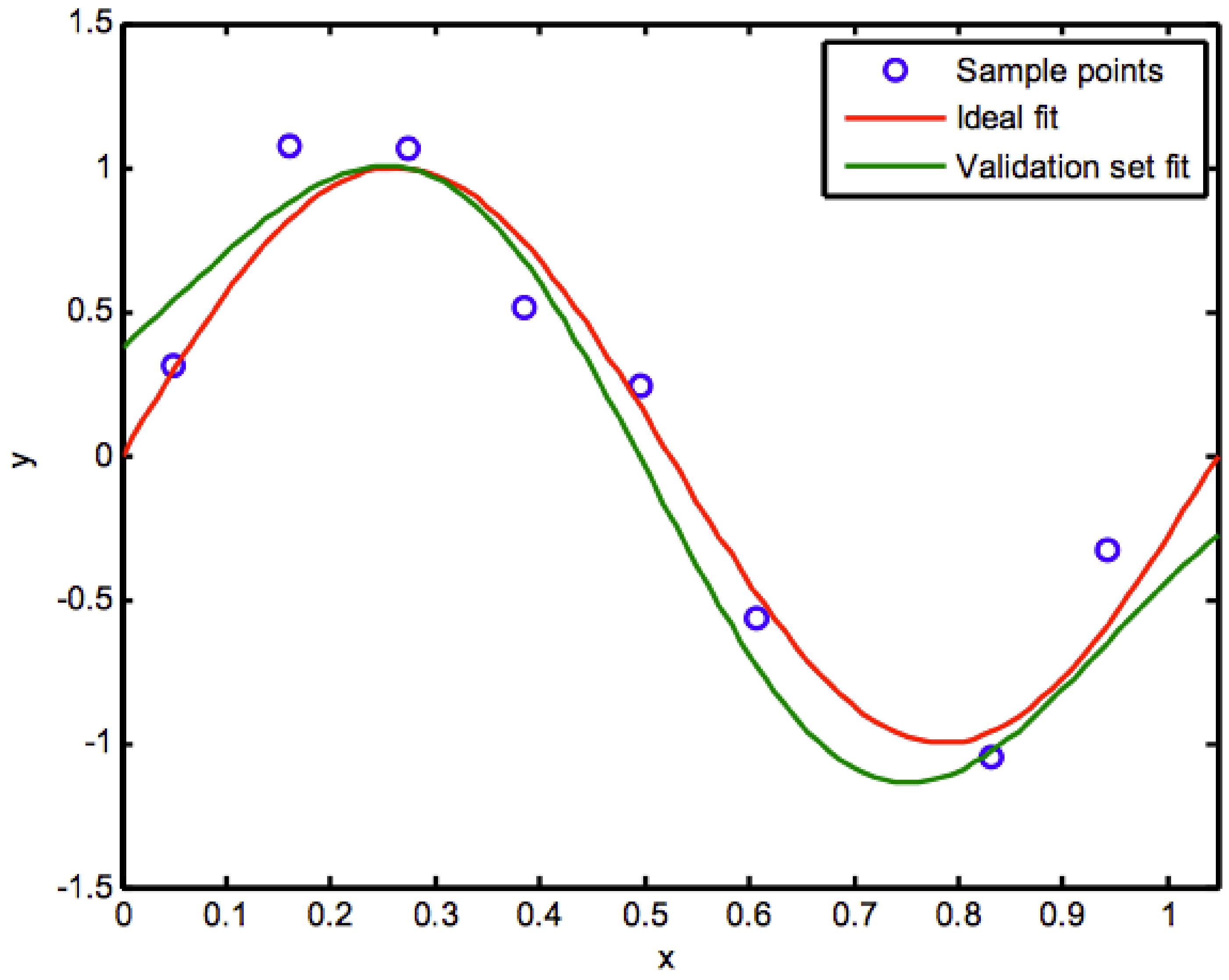
K-fold cross-validation

- Split the data into K subsets or folds
- For every $k = 1, \dots, K$:
  - Train the model on every fold except the k-th fold
  - Compute the test error
- Average the test errors



Choosing λ using validation dataset





Pick up the λ with the lowest mean value of E_{RMS} obtained from cross-validation