$\rightsquigarrow$ Linear combination of features $\Rightarrow$ $\hat{y}_p = \underbrace{X}_{1 \times d+1} . \Theta \rightsquigarrow \in \mathbb{R}$

$d+1 \times 1$

Min $\quad E(\Theta) = L(\Theta) = \frac{1}{N} \sum_{i=1}^{N} (y_a - \hat{y}_p)^2 \quad \Rightarrow \nabla_{\Theta} E(\Theta) = 0$

$$\underset{(d+1 \times 1)}{\Theta} = (X^T X)^{-1} X^T y$$

Polynomial features $\underline{X_1}$ $\quad \hat{y}_p = \Theta_0 + \Theta_1 X_1 \rightsquigarrow X$- space

Z space $\quad X_1, X_1^2, X_1^3, \dots, X_1^d$

$\downarrow \quad \downarrow \quad \downarrow \quad \quad \downarrow$

$Z_1 \quad Z_2 \quad Z_3 \quad \quad Z_d \quad \Rightarrow \hat{y}_p = Z . \Theta$

Regularized Linear Regression

Mahdi Roozbahani
Georgia Tech

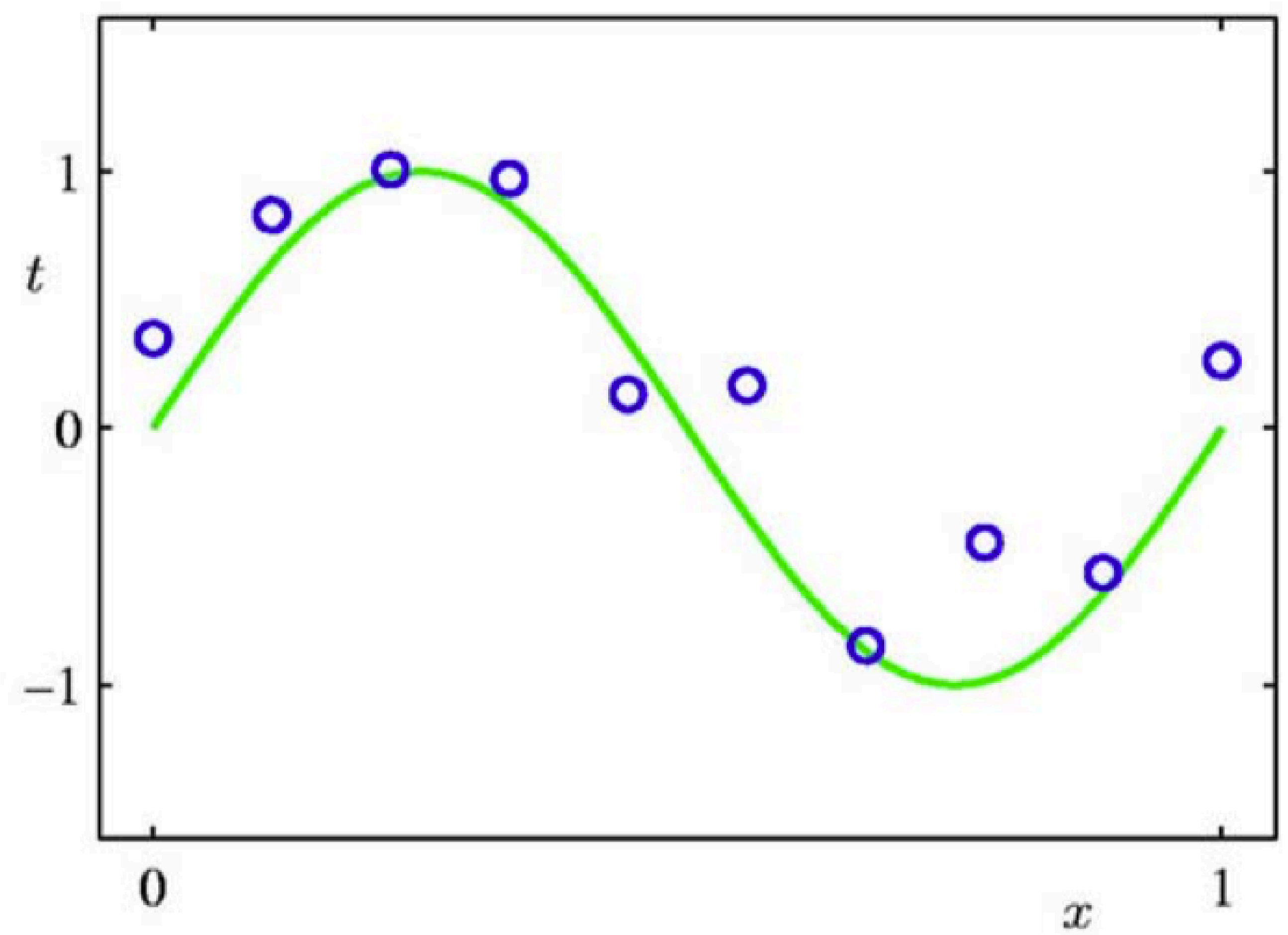
These slides are inspired based on slides from Andrew Zisserman, Jonathan Taylor, Chao Zhang and Yaser Abu-Mostafa.


EVERY GROUP PROJECT
DOES 99% OF THE WORK
HAS NO IDEA WHAT'S GOING ON THE WHOLE TIME
SAYS HE'S GOING TO HELP BUT HE'S NOT
DISAPPEAR AT THE VERY BEGINNING AND DOESN'T SHOW UP AGAIN TIL THE VERY END

Outline

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

Regression: Recap

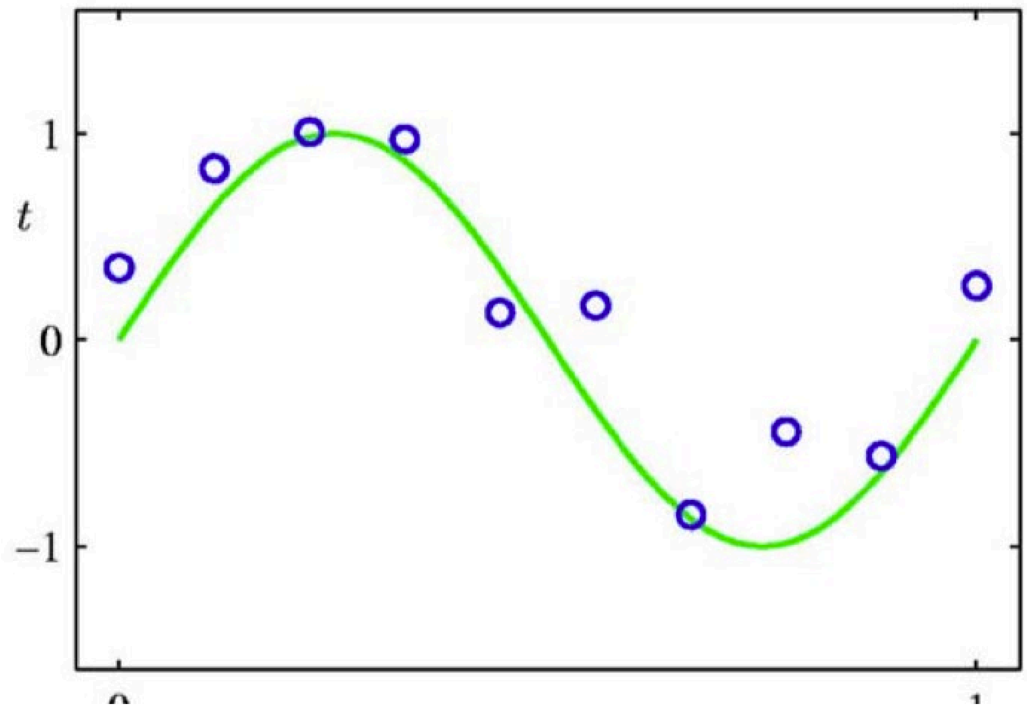


- Suppose we are given a training set of N observations

$$(x_1, \ldots, x_N) \text{ and } (y_1, \ldots, y_N)$$

- Regression problem is to estimate y(x) from this data

Regression: Recap



- Want to fit a polynomial regression model

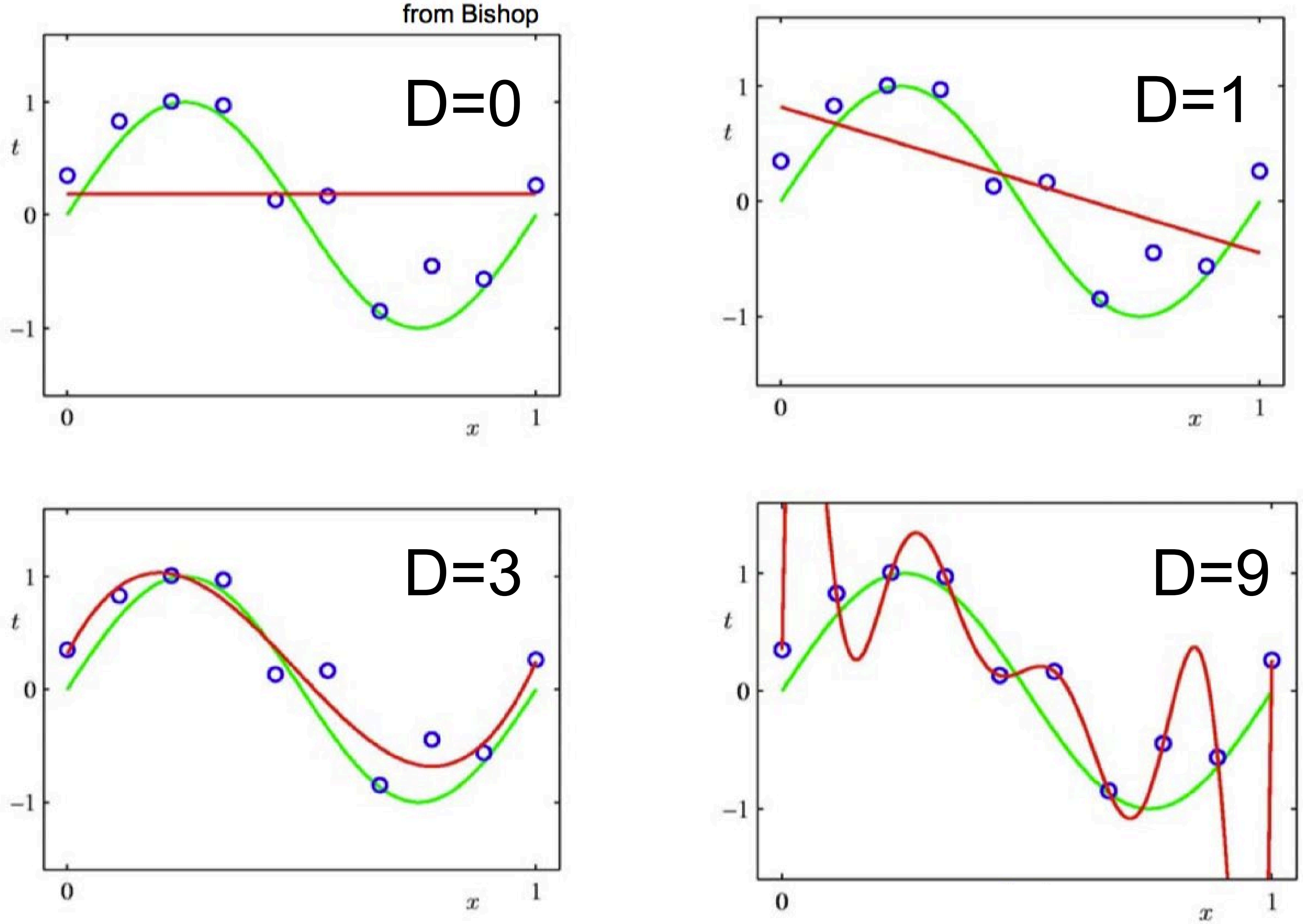
$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{\mathrm{d}} x^{\mathrm{d}} + \epsilon$$

- $z = \{1, x, x^2, \ldots, x^d\} \in R^d$ and $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_{\mathrm{d}})^\top$

$$y = z\theta \rightsquigarrow E(\theta) \neq 0$$

Which One is Better?

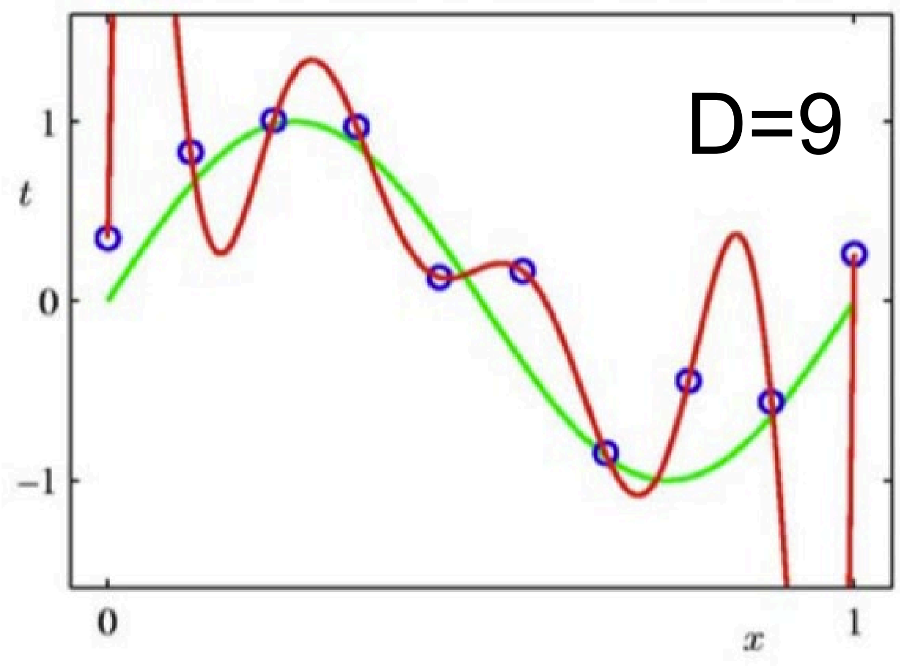
from Bishop

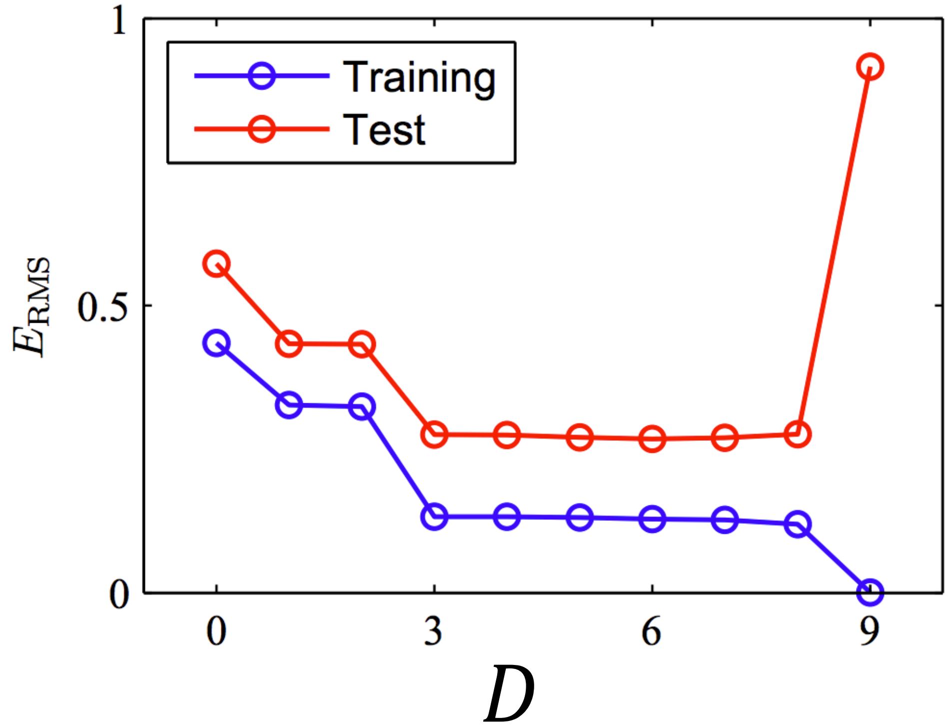


- Can we increase the maximal polynomial degree to very large, such that the curve passes through all training points?

No, this can lead to **overfitting**!

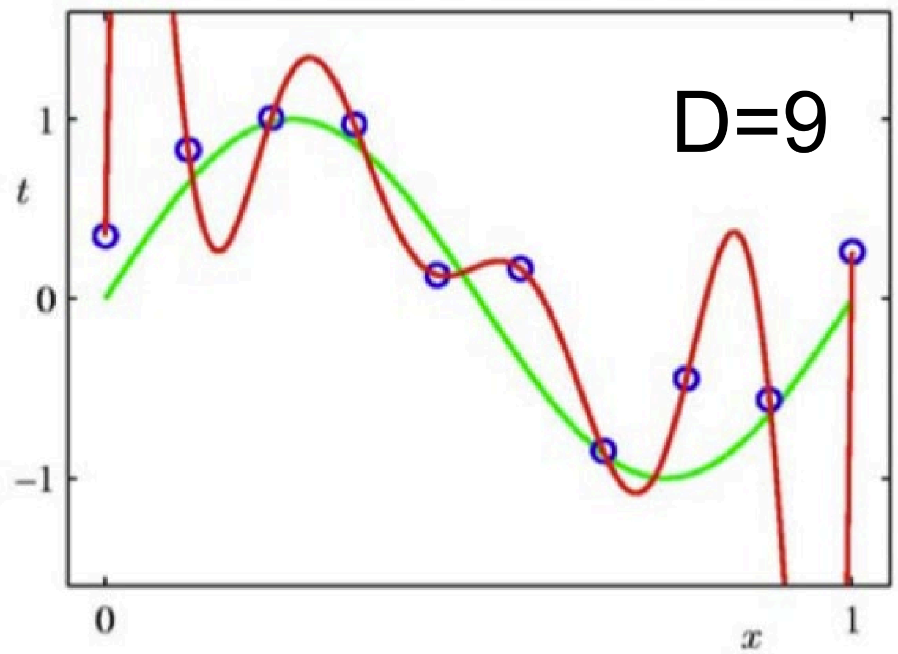
The Overfitting Problem





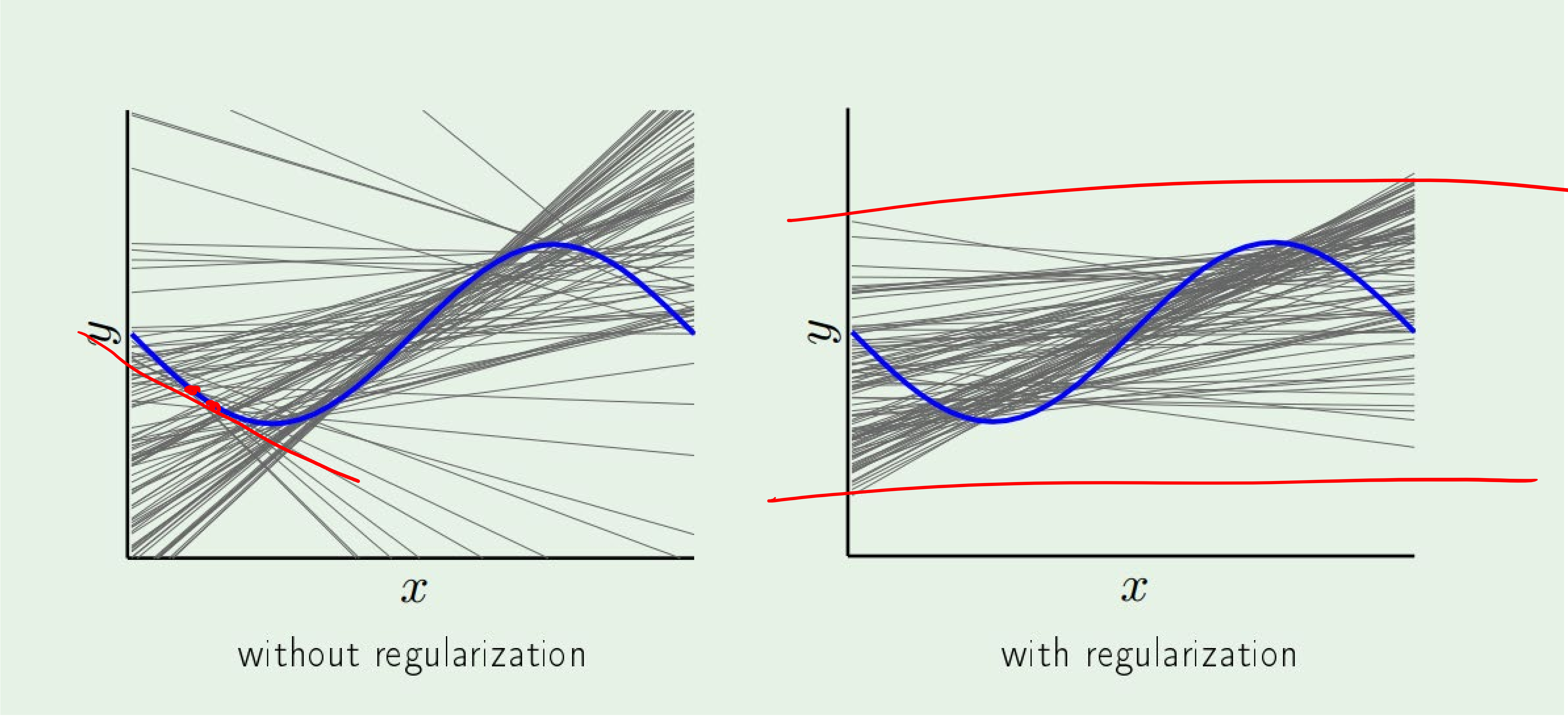
- The training error is very low, but the error on test set is large.
- The model captures not only patterns but also noisy nuisances in the training data.

The Overfitting Problem



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

Regularization
(smart way to cure overfitting disease)



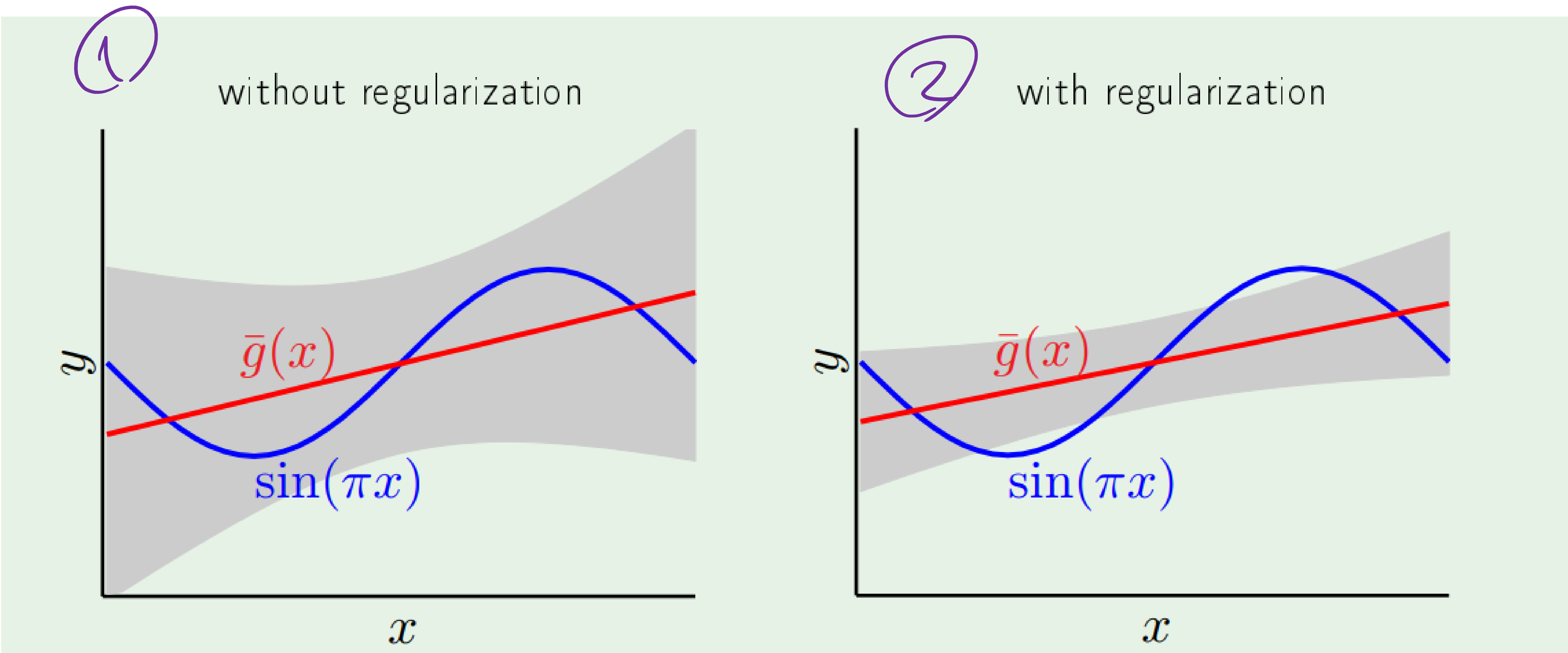
Put a brake on fitting

Fit a linear line on sinusoidal with just two data points

Who is the winner?

$E(\Theta) = \text{bias}^2 + \text{Variance}$

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



bias=0.21; var=1.69

bias=0.23; var=0.33

Polynomial Model

Want to fit a polynomial regression model

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d + \epsilon$$

Let's rewrite it as:

$$y = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \cdots + \theta_d z_\mathrm{d} + \epsilon = \boldsymbol{z\theta}$$

Regularizing is just constraining the weights ($\boldsymbol{\theta}$)

For example: let's do a **hard** constraining

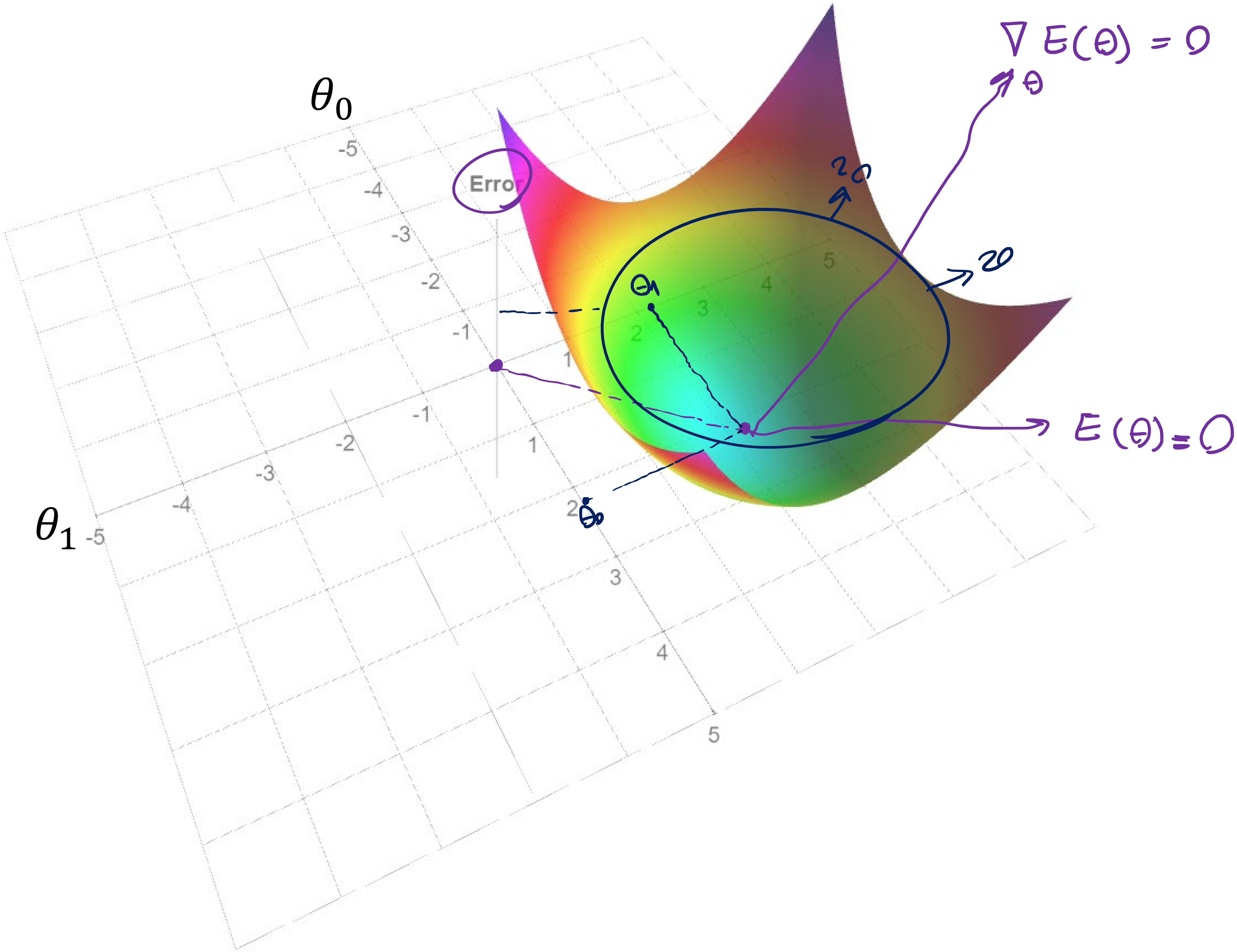
$$y = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \cdots + \theta_d z_{\mathrm{d}}$$

subject to

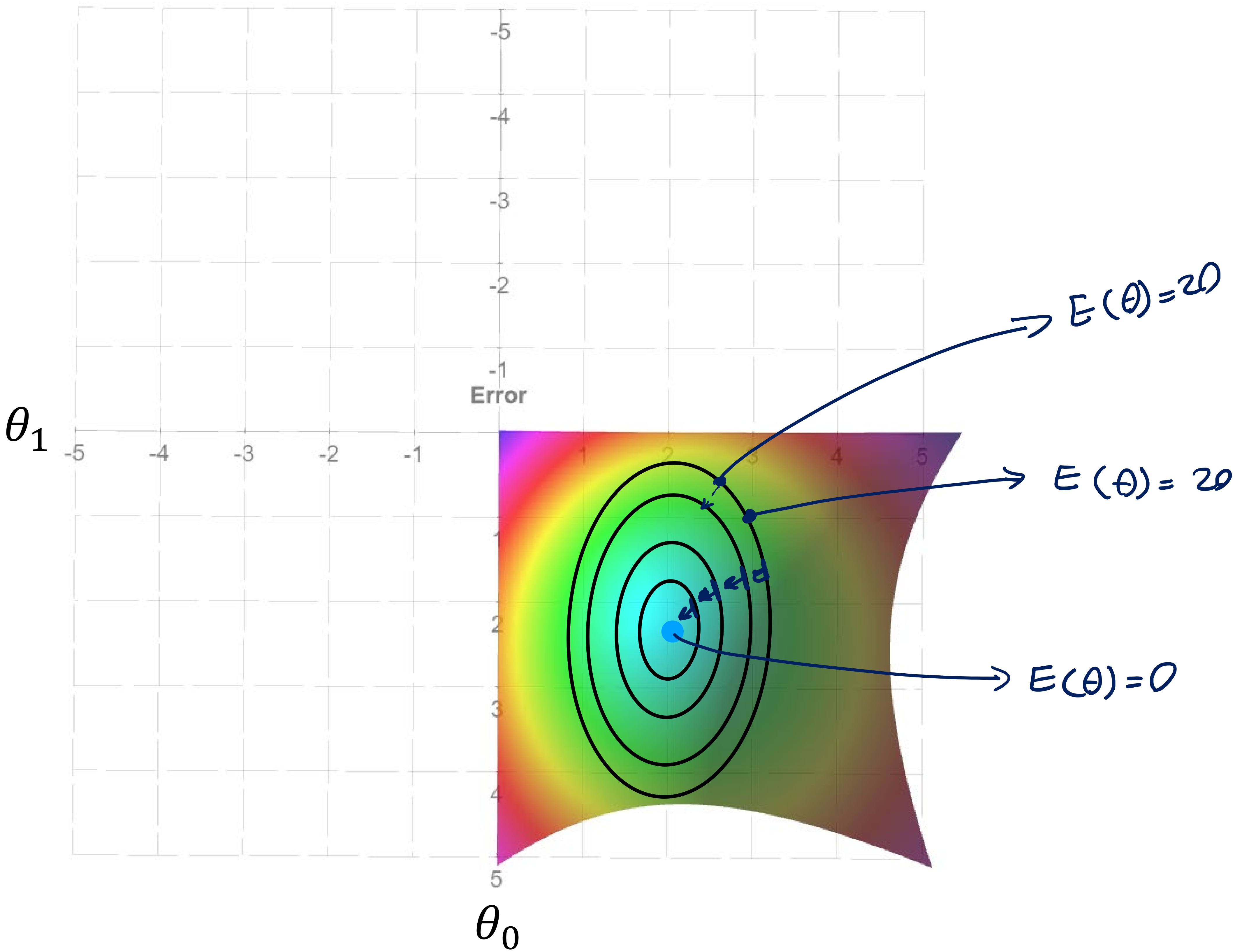
$$\theta_d = 0 \; for \; d > 2$$

$$y = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + 0 + \cdots + 0$$

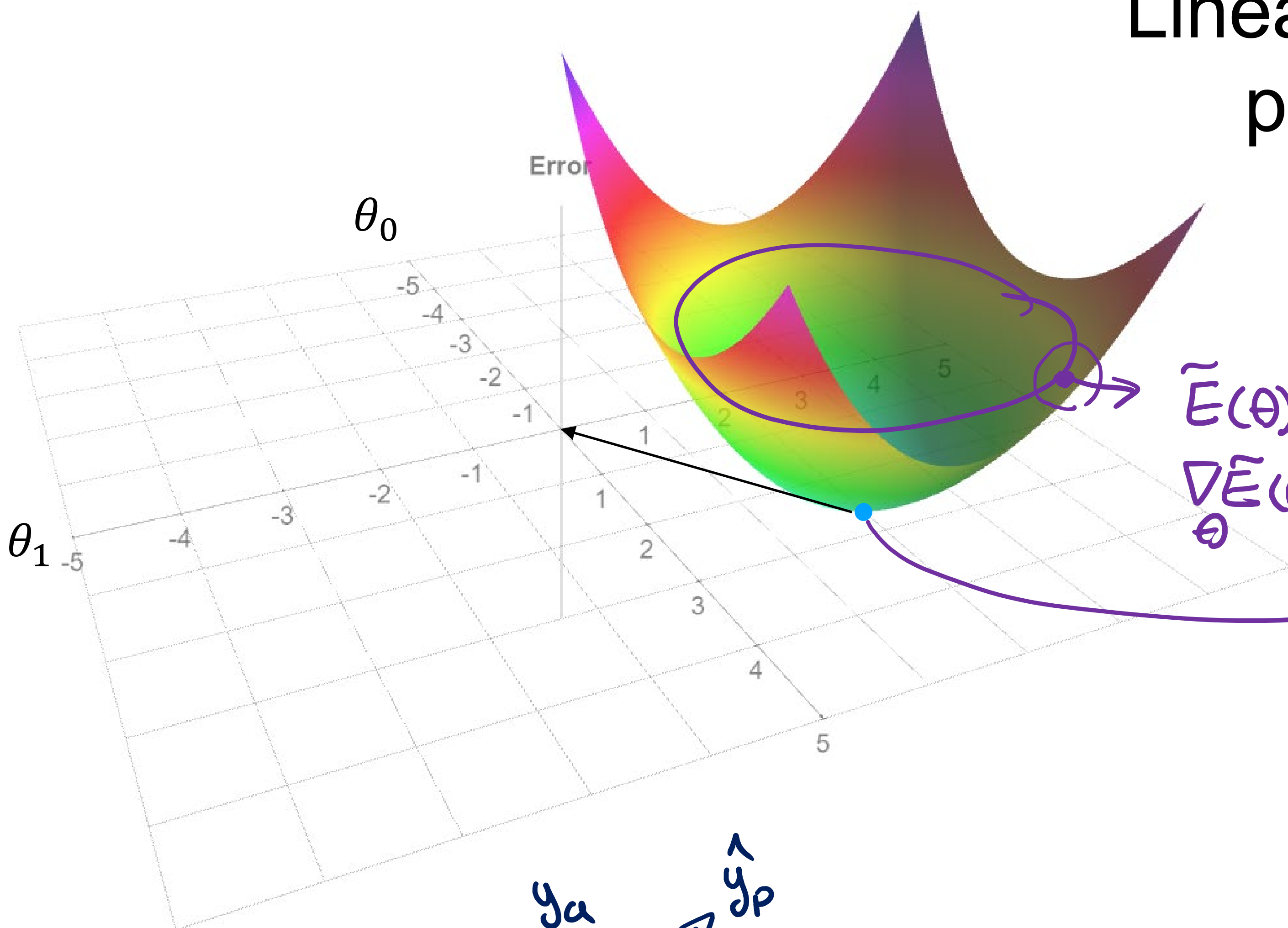
$$E(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y^{i} - z_{i}\theta\right)^{2}$$



Project the same graph on x-y using contour plot



Linear regression with a very high polynomial degree solution

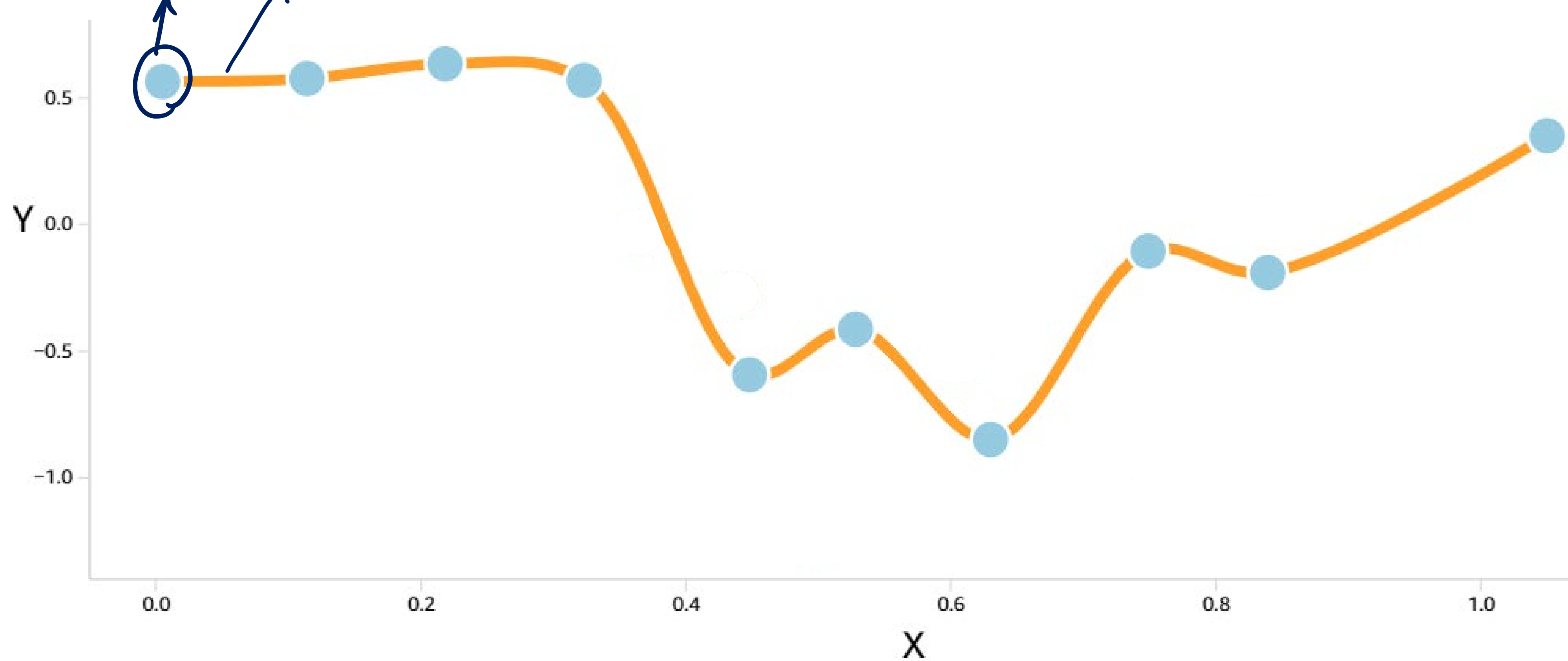


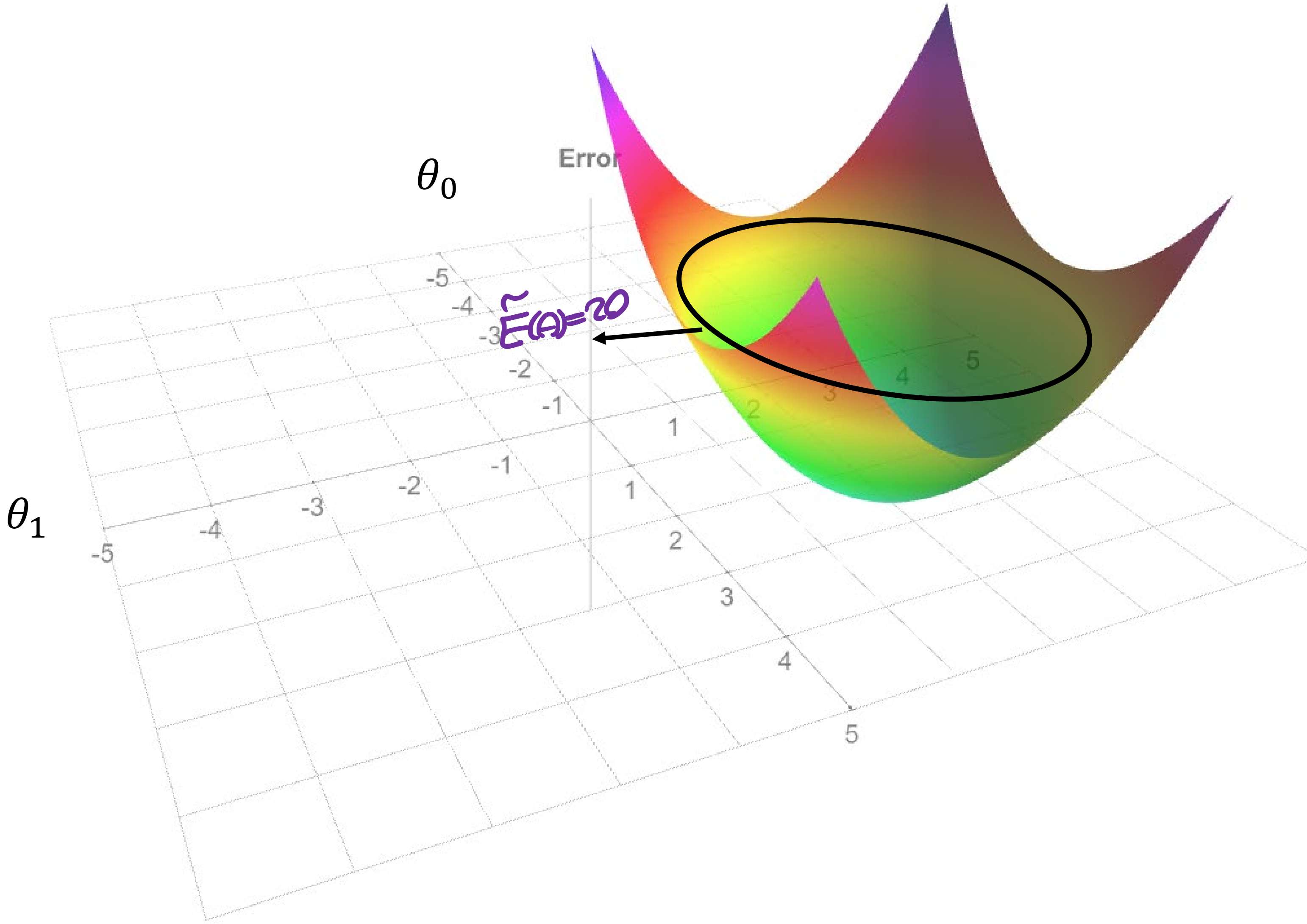
$\Theta_{overfitted}$ $\quad E(\Theta_{overfitted}) = 0$

$\tilde{E}(\Theta) = 20$

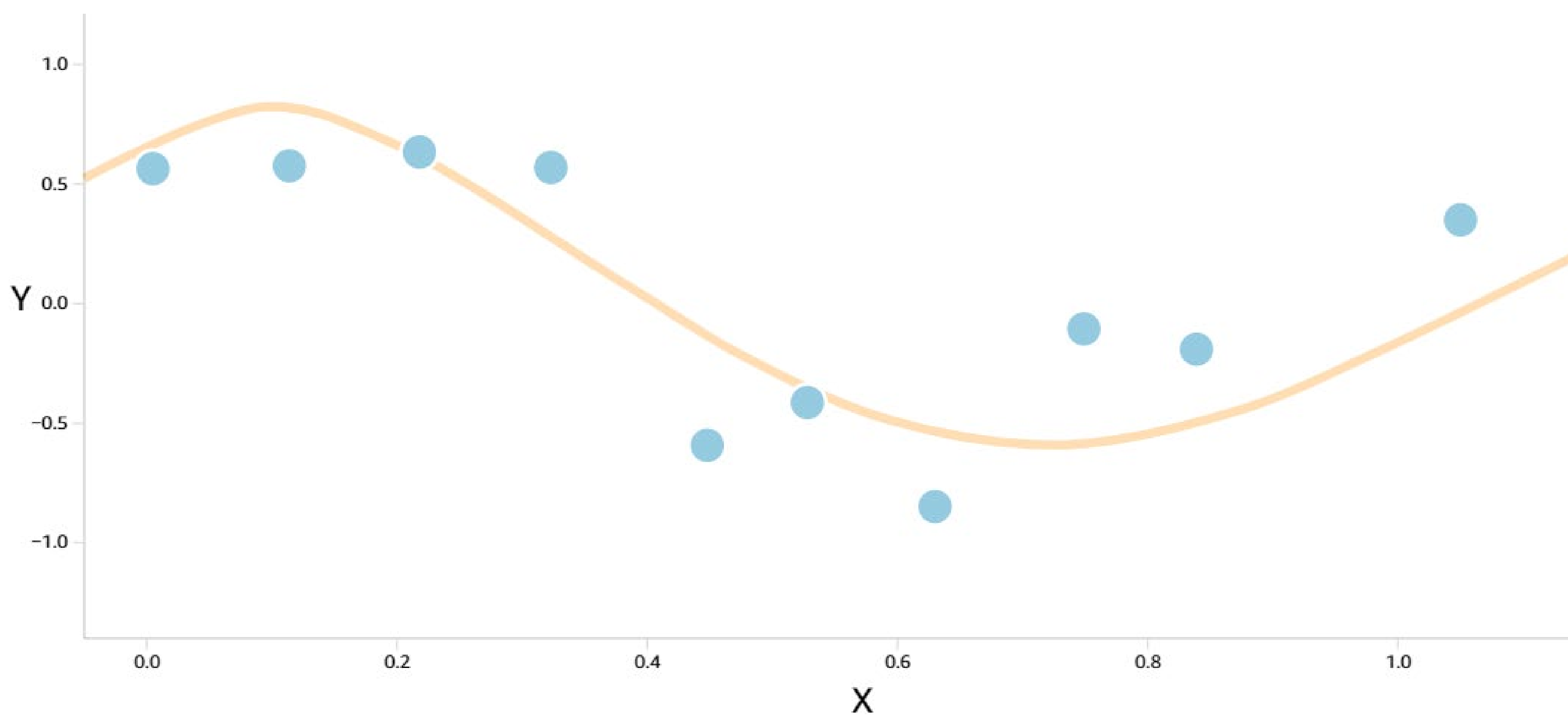
$\nabla_\Theta E(\Theta) = 0$

$\nabla_\Theta E(\Theta) = 0$





θ_0
Error
$\tilde{E}(\theta)=20$
θ_1



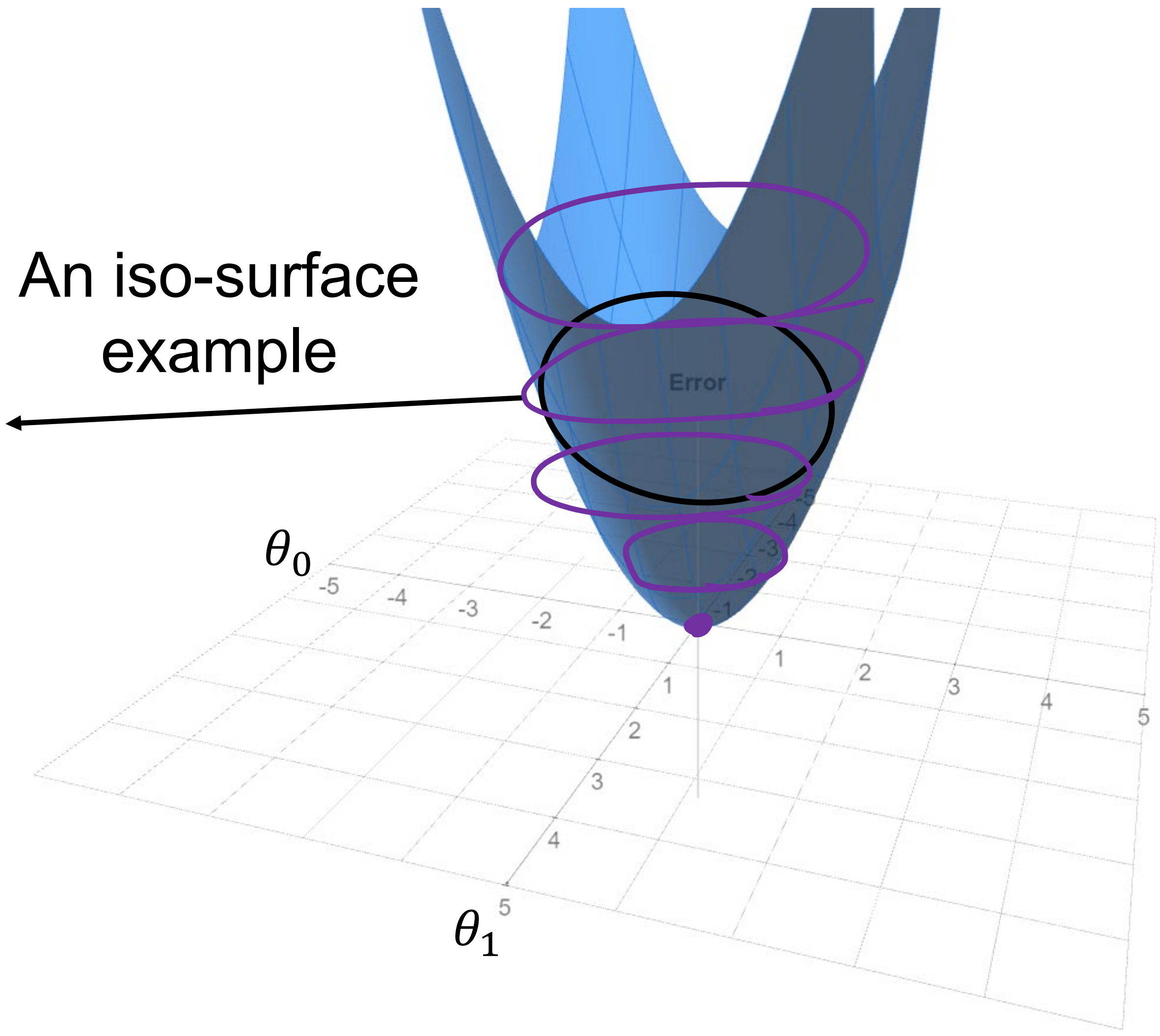
Y
X
1.0
0.5
0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

How can we get an optimal solution with a positive error for a model that overfits?

We need to introduce a constraint

$$g(\theta) = \theta_0^2 + \theta_1^2 = \theta^T\theta = C$$

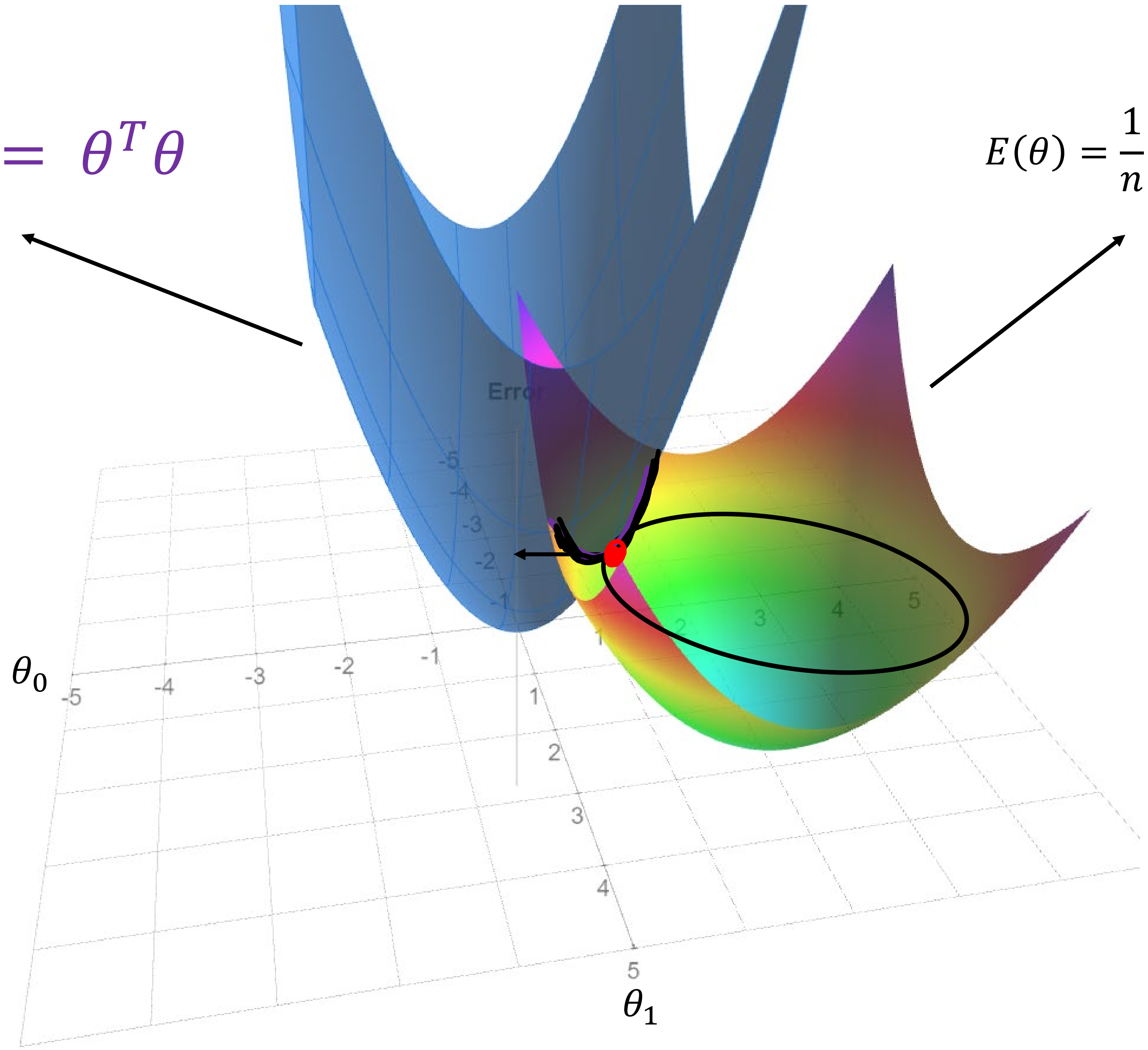
An iso-surface example



Error function together with a new introduced constraint

$$g(\theta) = \theta_0^2 + \theta_1^2 = \theta^T\theta$$

$$E(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2$$



Let's define the Lagrange function

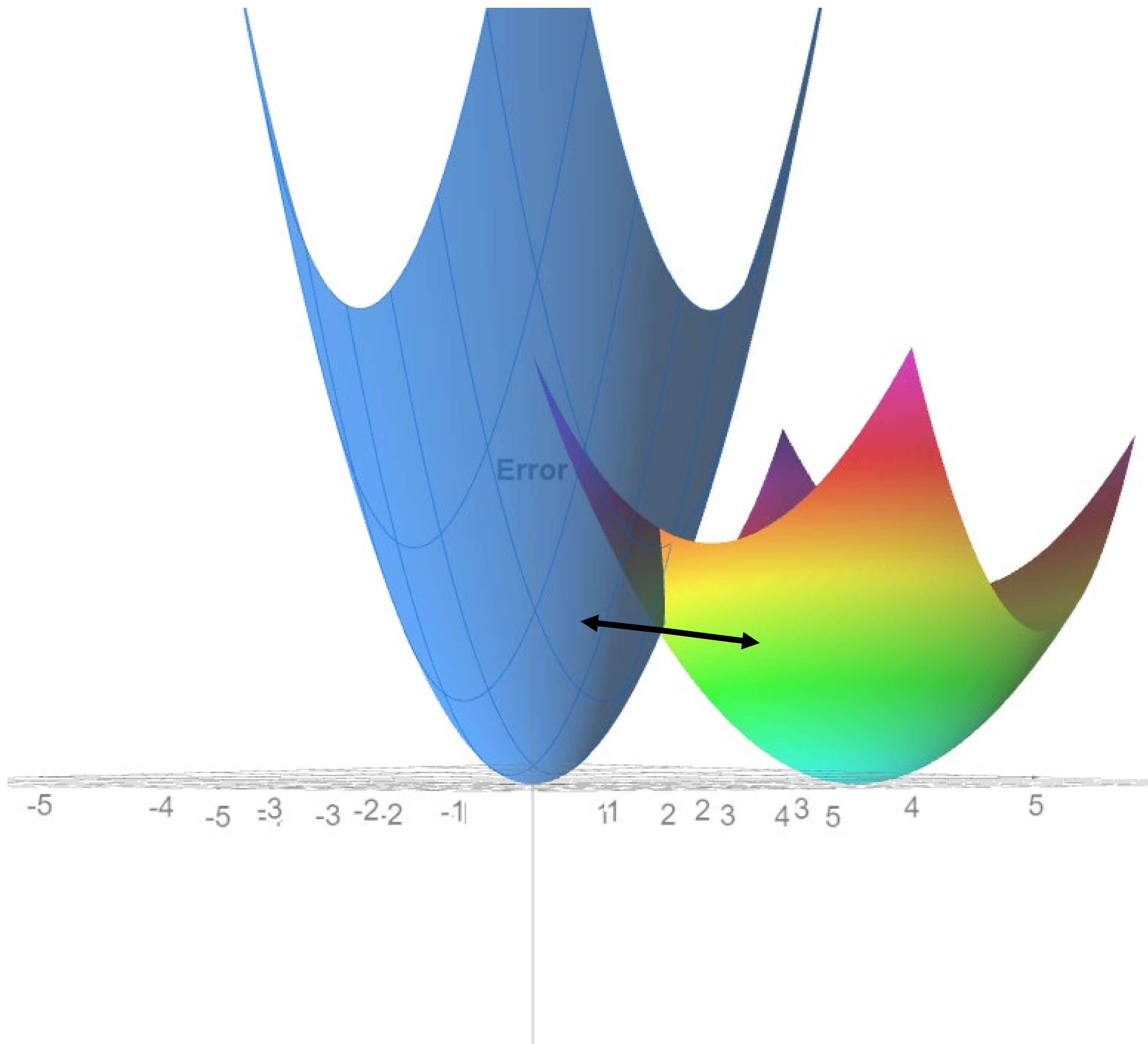
$$L(\theta, \lambda) = E(\theta) + \lambda g(\theta)$$

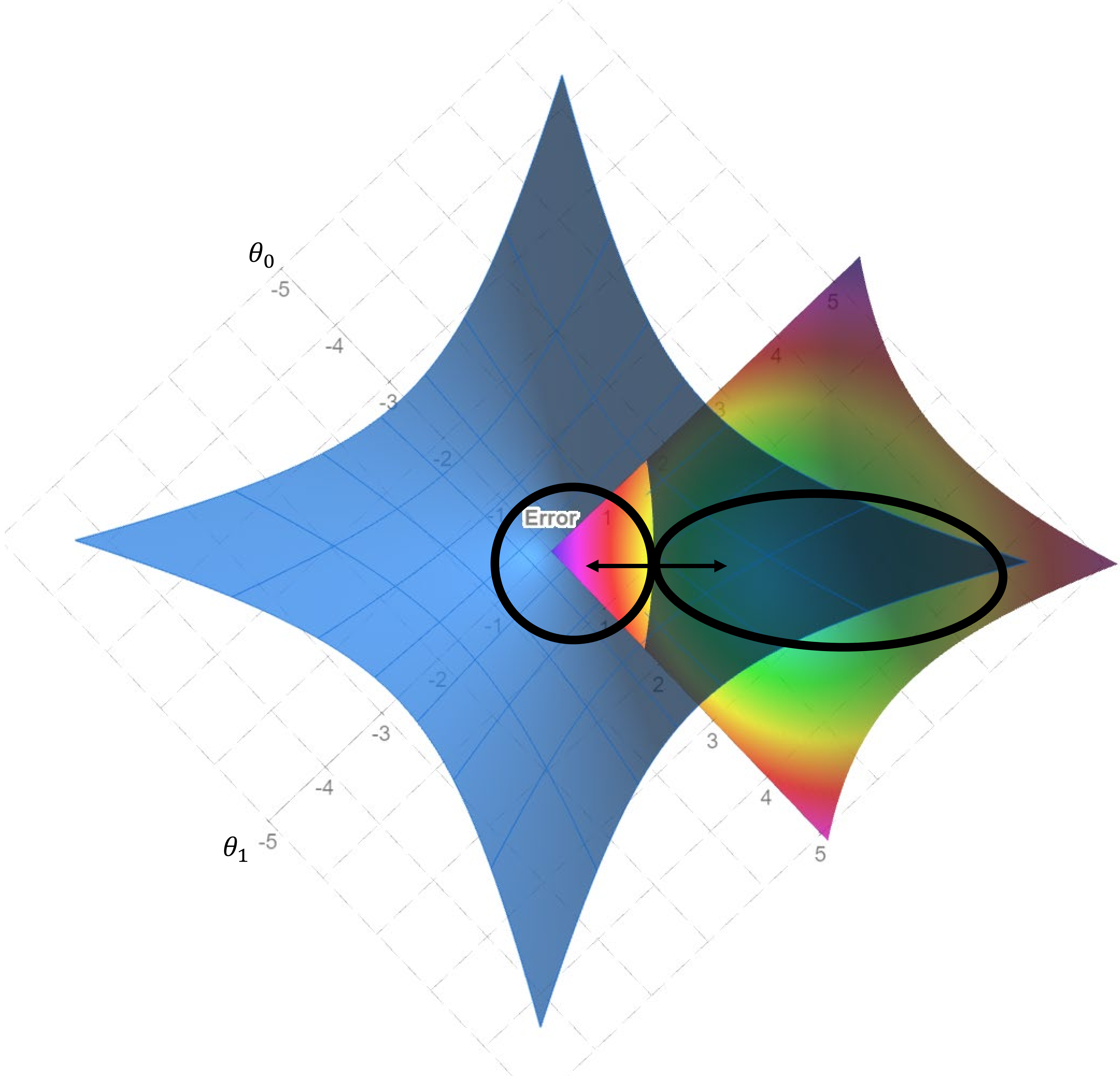
$$L(\theta, \lambda) = E(\theta) + \lambda \theta^T \theta$$

$$\nabla L(\theta, \lambda) = 0 \qquad \nabla[E(\theta) + \lambda \theta^T \theta] = 0$$

$$\nabla[E(\theta)] + \lambda \nabla[\theta^T \theta] = 0$$

How to enforce the gradient of Lagrange function to be zero





θ_0
θ_1
Error
-5
-4
-3
-2
-1
1
2
3
4
5

Let's calculate the gradients

Gradient of constraint $g(\theta)$ $\qquad \nabla[\theta^T\theta] = 2\theta$

$$\nabla[E(\theta)] + \lambda\nabla[\theta^T\theta] = 0$$

$$\nabla[E(\theta)] = -\lambda\nabla[\theta^T\theta]$$

$$\nabla E(\theta) = -2\lambda\theta$$

$$\text{Min} \rightarrow \boxed{\tilde{E}(\theta) = \overbrace{E(\theta)}^{\text{Objective function}} + \overbrace{\lambda\theta^T\theta}^{\text{Regularization term / Penalty term}}} = E(\theta) \quad \text{s.t. } \theta^T\theta = C$$

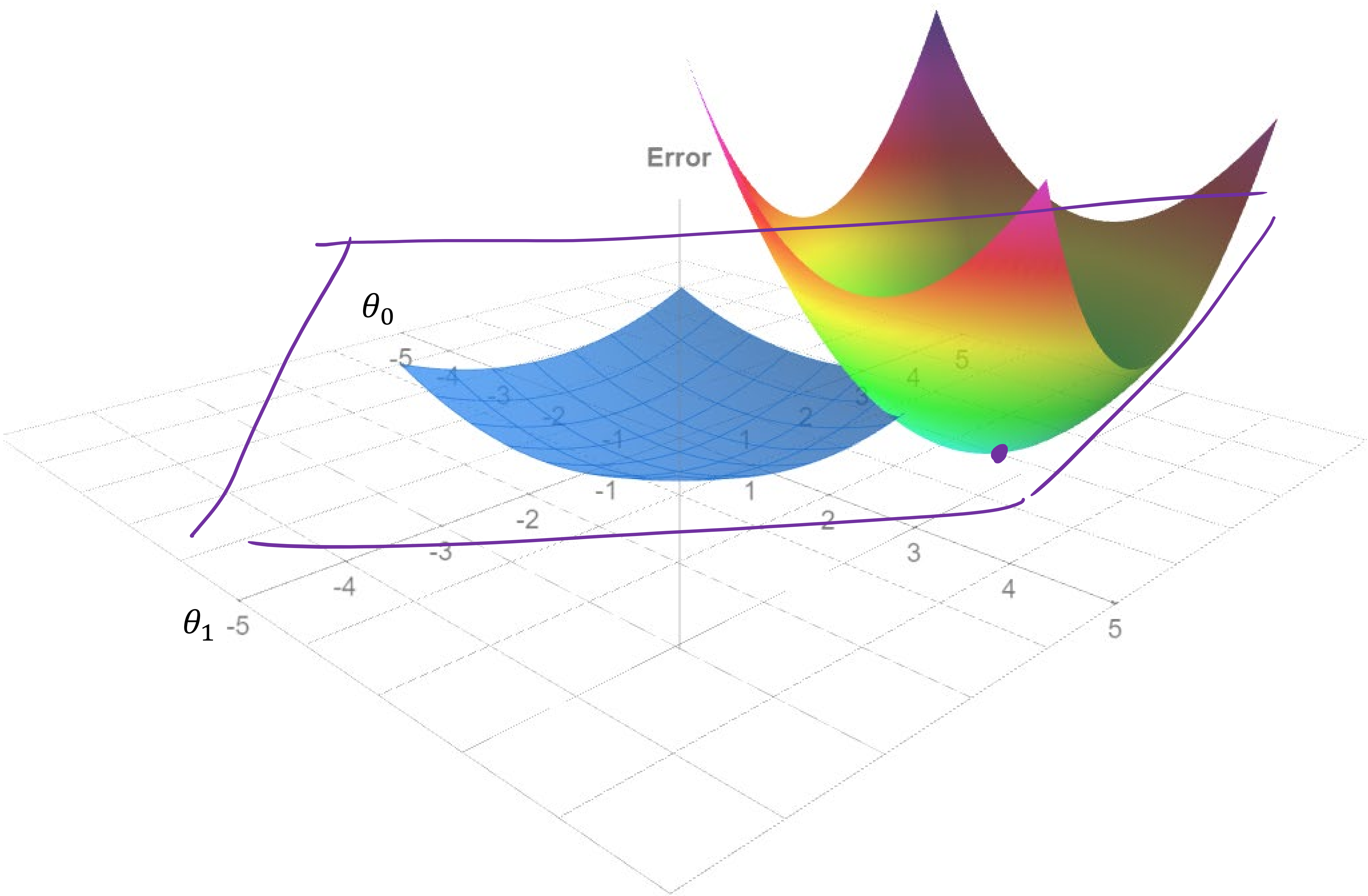
$$\nabla E(\theta) + 2\lambda\theta = 0$$

Let's do integration $\quad E(\theta) + \lambda\theta^T\theta$

The effect of low Lambda

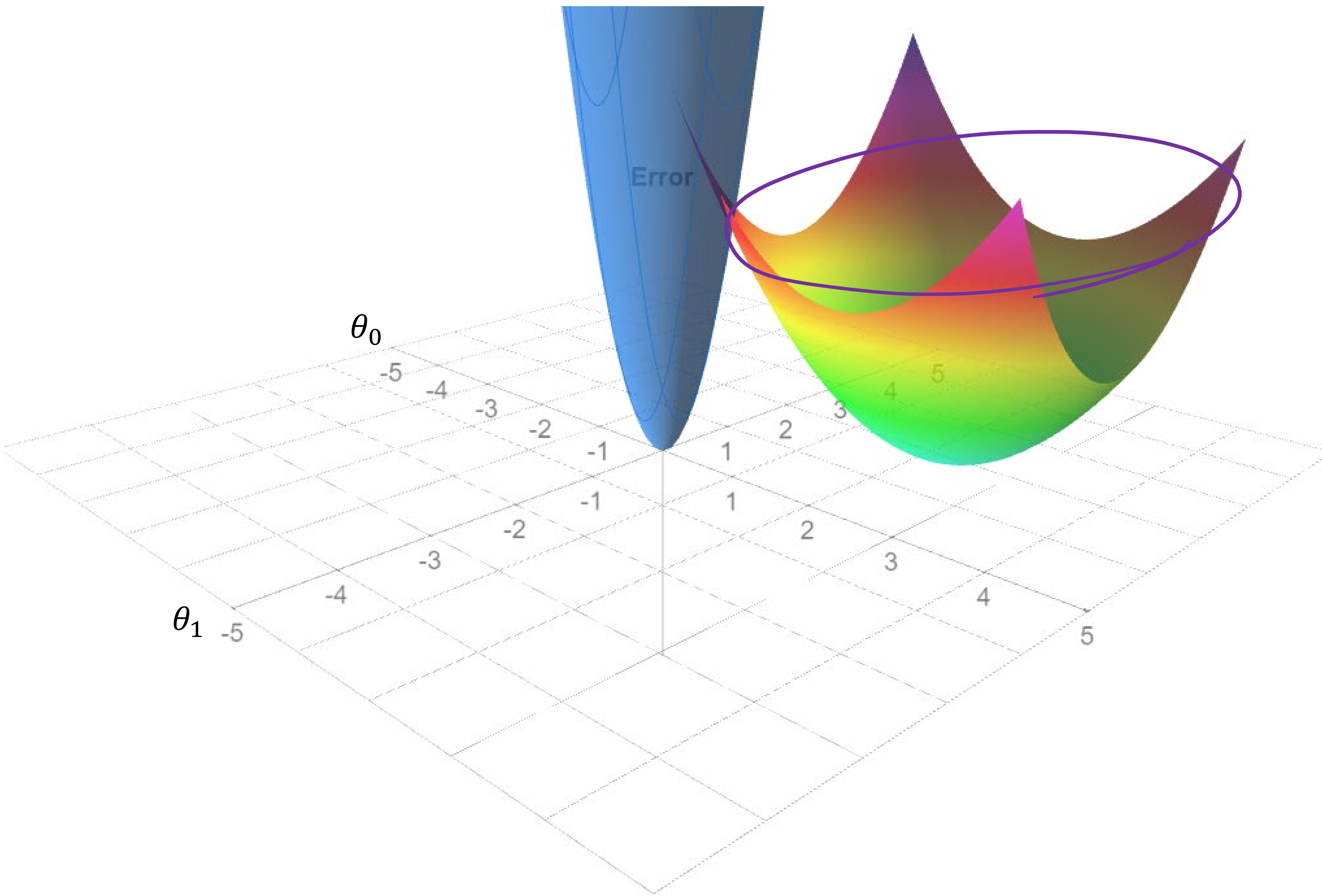
$$E(\theta) + \frac{\lambda}{2N}\theta^T\theta$$

number of datapoints



The effect of high Lambda

$$E(\theta) + \frac{\lambda}{N}\theta^T\theta$$



Regularized Learning

an outsider parameter = Hyper Parameter

Now we know Why this term leads to the regularization of parameters

Minimize $E(\theta) + \lambda\theta^T\theta$

Regularized Error

$$\tilde{E}(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2 + \frac{\lambda}{2N}\|\theta\|_2^2$$

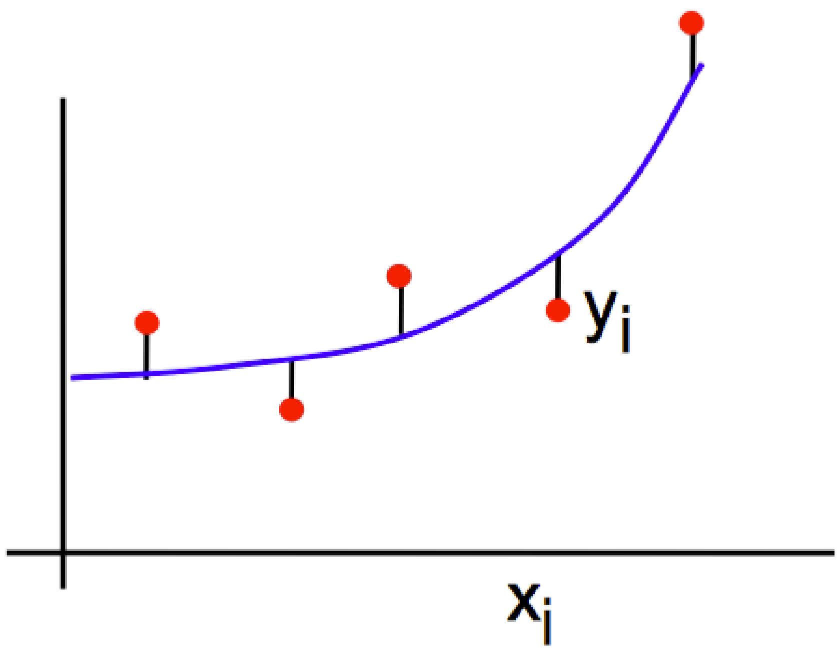
L2 Regularization term

Outline

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

Ridge Regression

$$\tilde{E}(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2 + \lambda\|\theta\|_2^2$$



$$\theta_0 + \theta_1 z_1 + \theta_2 z_2 + \cdots + \theta_d z_d + \epsilon = \boldsymbol{z\theta}$$

General form

$$\tilde{E}(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2 + \lambda\|\theta\|_2^2$$

Matrix form

$$\tilde{E}(\theta) = \frac{1}{N}(y - z\theta)^{\mathrm{T}}(\mathrm{y} - z\theta) + \lambda\|\theta\|_2^2$$

$$\frac{\partial\tilde{E}(\theta)}{\partial\theta} = -z^T(y - z\theta) + \lambda\theta$$

$$(z^T z + \lambda I)\theta = z^T y$$

$$\boxed{\theta = (z^T z + \lambda I)^{-1} z^T y}$$

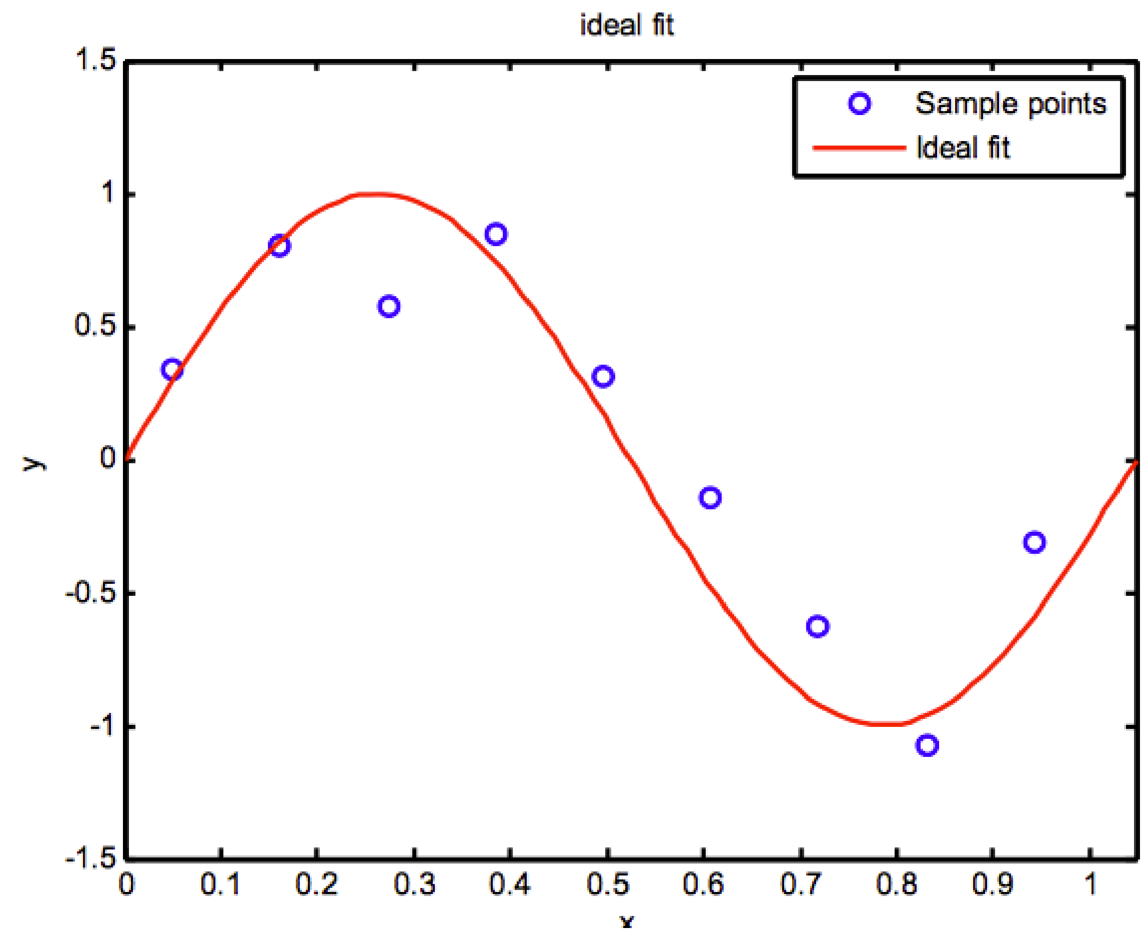
$\theta_{overfitted} = (z^T z)^{-1} z^T y$

Ridge Regression Example

- The red curve is the true function (which is not a polynomial)

- The data points are samples from the curve with added noise in y.

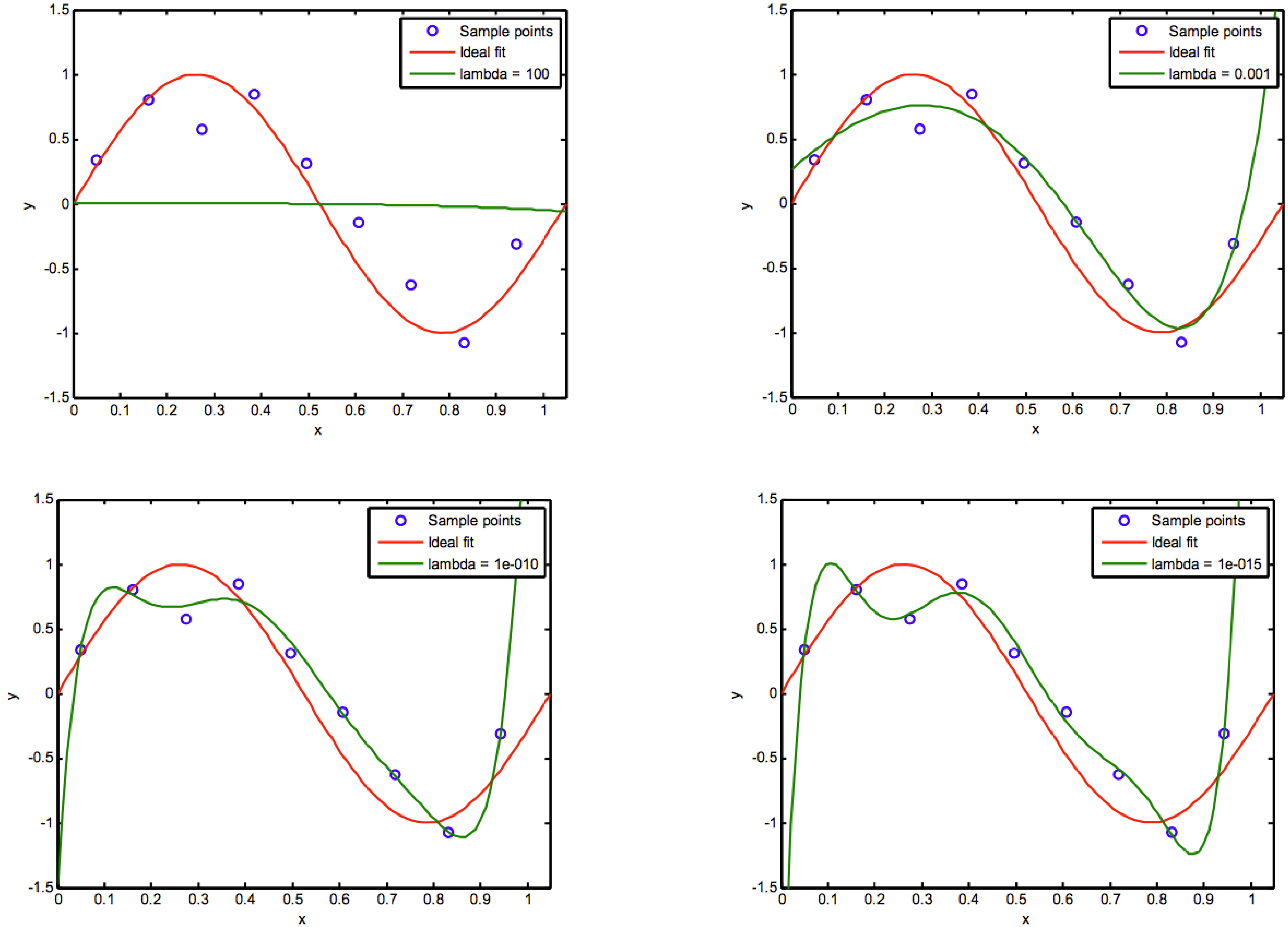
- There is a choice in both the degree, D, of the basis functions used, and in the strength of the regularization



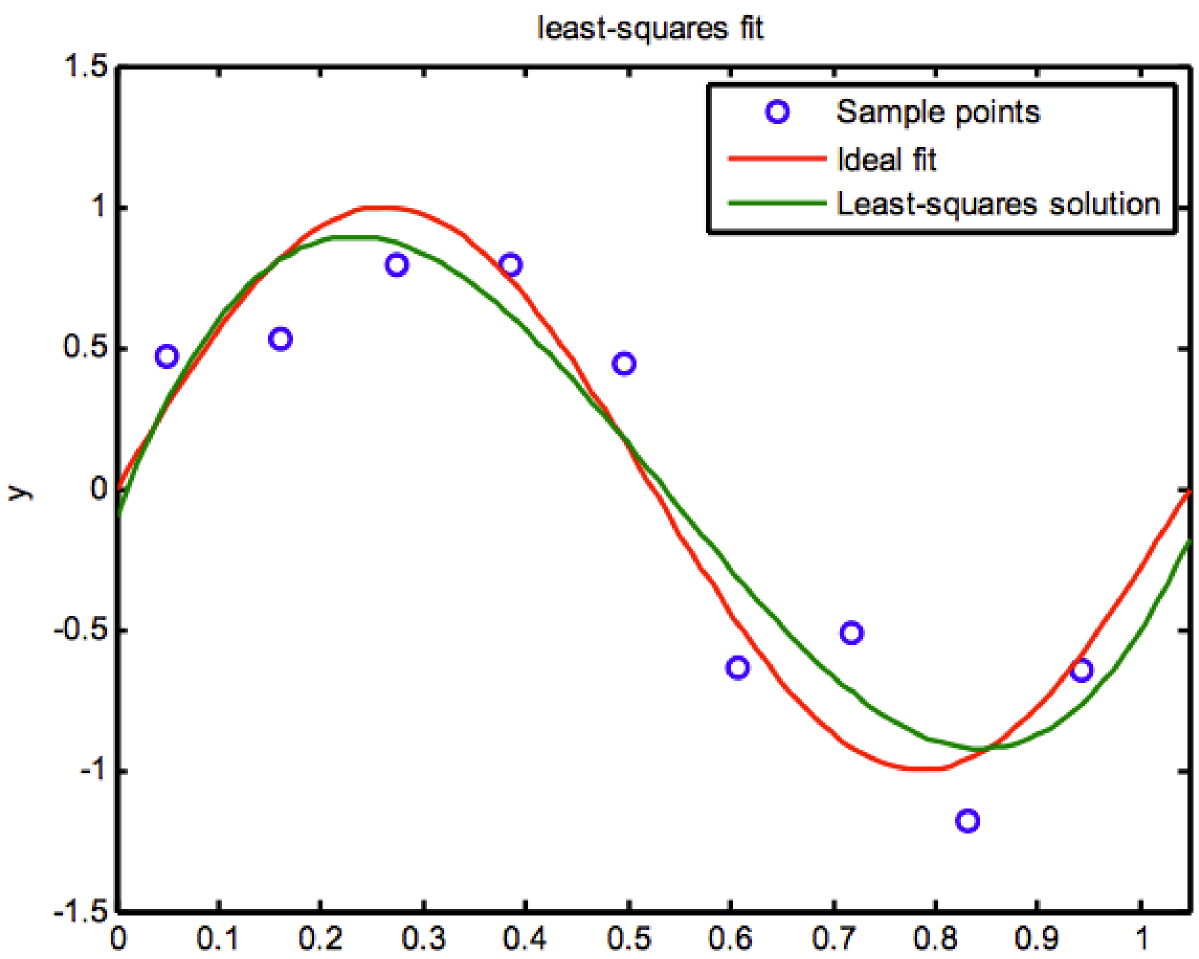
$$f(x, \theta) = z\theta \qquad z: x \to z$$

$$\tilde{E}(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2 + \lambda\|\theta\|_2^2 \qquad \theta \in \mathbb{R}^{D+1}$$

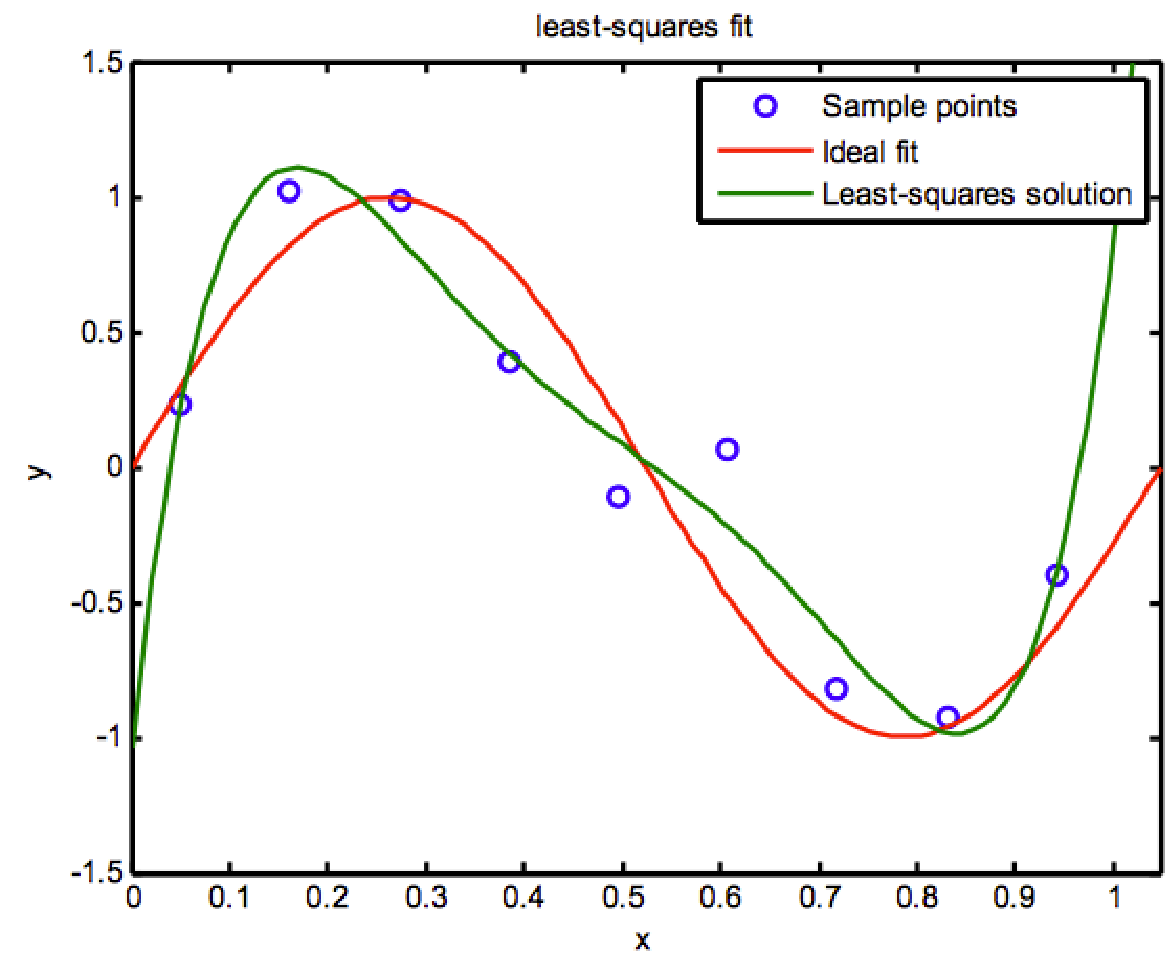
N = 9 samples, D = 7



D = 3


least-squares fit
Sample points
Ideal fit
Least-squares solution
y
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

D = 5


least-squares fit
Sample points
Ideal fit
Least-squares solution
y
x
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Outline

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

Regularized Regression

$$\tilde{E}(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2 + \lambda\|\theta\|_2^2$$

Squared loss\Error $\frac{1}{N}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2$

L2 Regularizer $\lambda\|\theta\|_2^2$

Now let's look at another regularization choice.

The Lasso Regularization (L1 norm) and sparsity

Lasso = **L**east Absolute **S**hrinkage and **S**election **O**perator

$\|\theta\|_2^2$

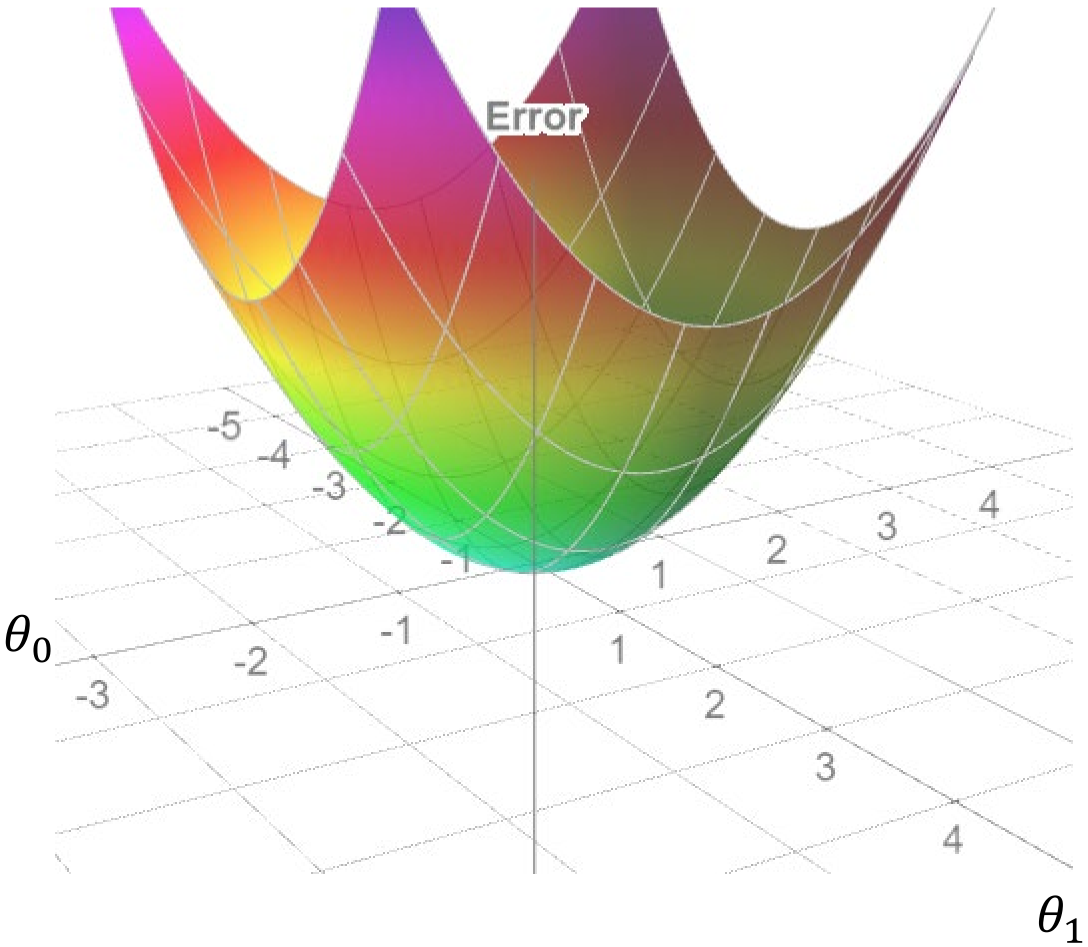
$\|\theta\|$

$$\tilde{E}(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y^i - z_i\theta\right)^2 + \lambda\|\theta\|_1$$

L1 norm induces sparsity. This means that some of the weights become zero, and the feature contribution will be completely removed. L1 Regularizer could be used for feature selection

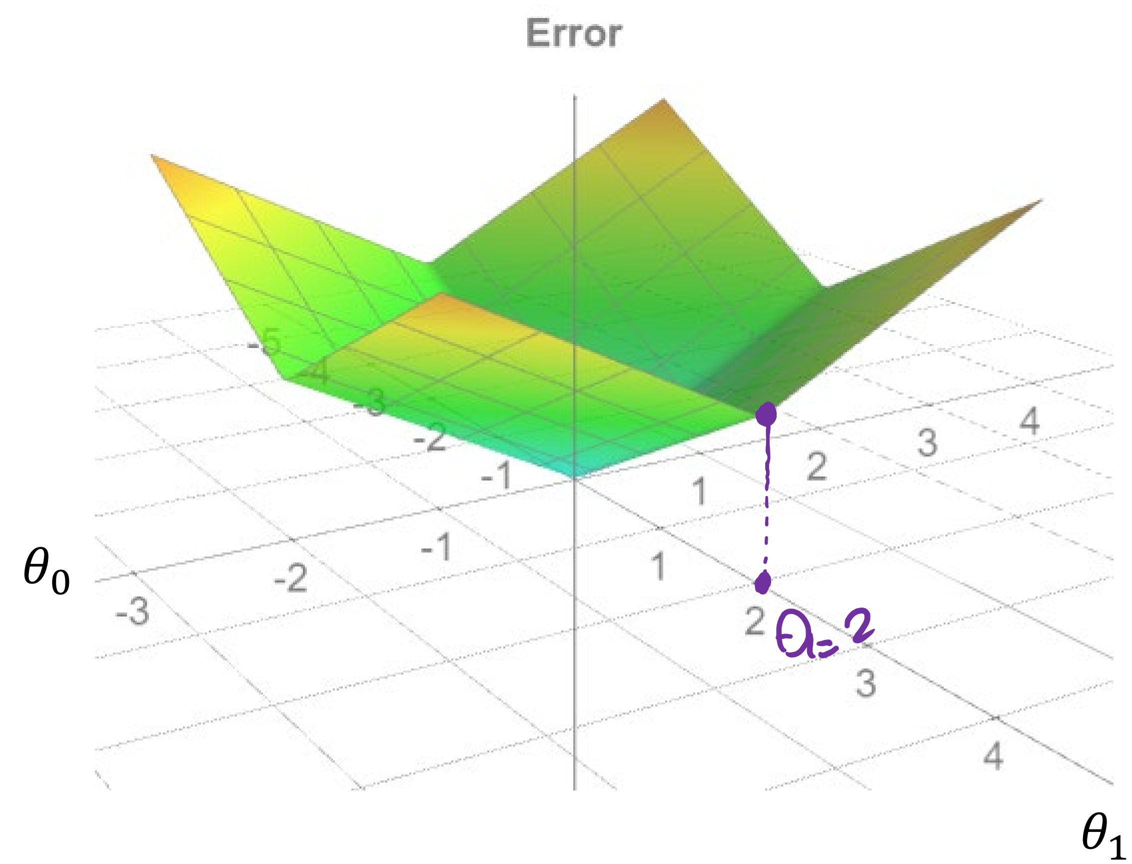
Ridge Regularizer

$$g(\theta) = \theta_0^2 + \theta_1^2 = \theta^T\theta$$



Lasso Regularizer

$$g(\theta) = \theta_0 + \theta_1 = \theta$$

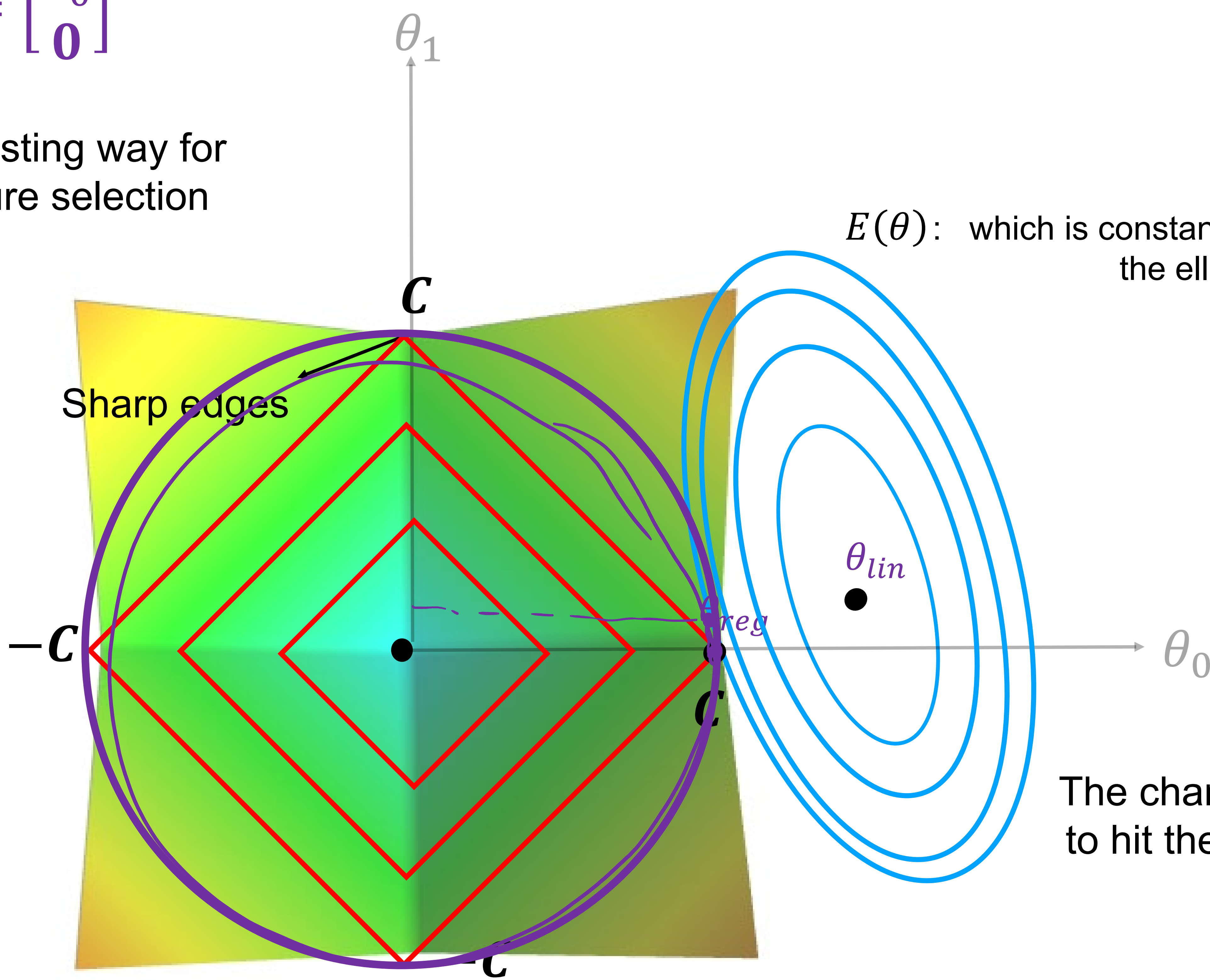


Let's say we have two parameters (θ_0 *and* θ_1)

$$Min\ E(\theta) = \frac{1}{N}(\mathrm{z}w - y)^T(\mathrm{z}\theta - y) + \lambda\|\theta\|_1$$

$$\theta = \begin{bmatrix} \theta_0 \\ \mathbf{0} \end{bmatrix}$$

Interesting way for feature selection



$E(\theta)$: which is constant on the surface of the ellipsoid

The chance is very high to hit the sharp corners first

Ridge versus Lasso

Ridge

$$\tilde{E}(\theta) = \frac{1}{N}(y - z\theta)^{\mathrm{T}}(\mathrm{y} - z\theta) + \lambda\|\theta\|_2^2$$

It is a convex model

Both mean squared error and L2 regularizer are differentiable.

We can get a closed form solution

Lasso

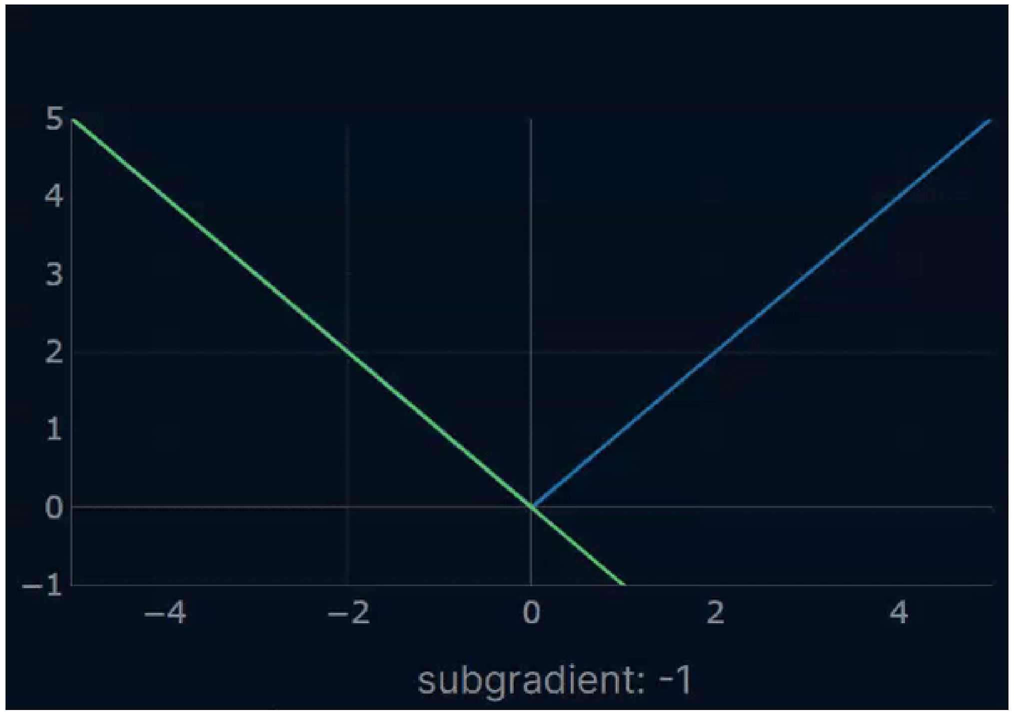
$$\tilde{E}(\theta) = \frac{1}{N}(y - z\theta)^{\mathrm{T}}(\mathrm{y} - z\theta) + \lambda\|\theta\|_1$$

It is a convex model

L1 regularizer is NOT differentiable.

We can **NOT** get a closed form solution

Sub-gradient Descend in Lasso

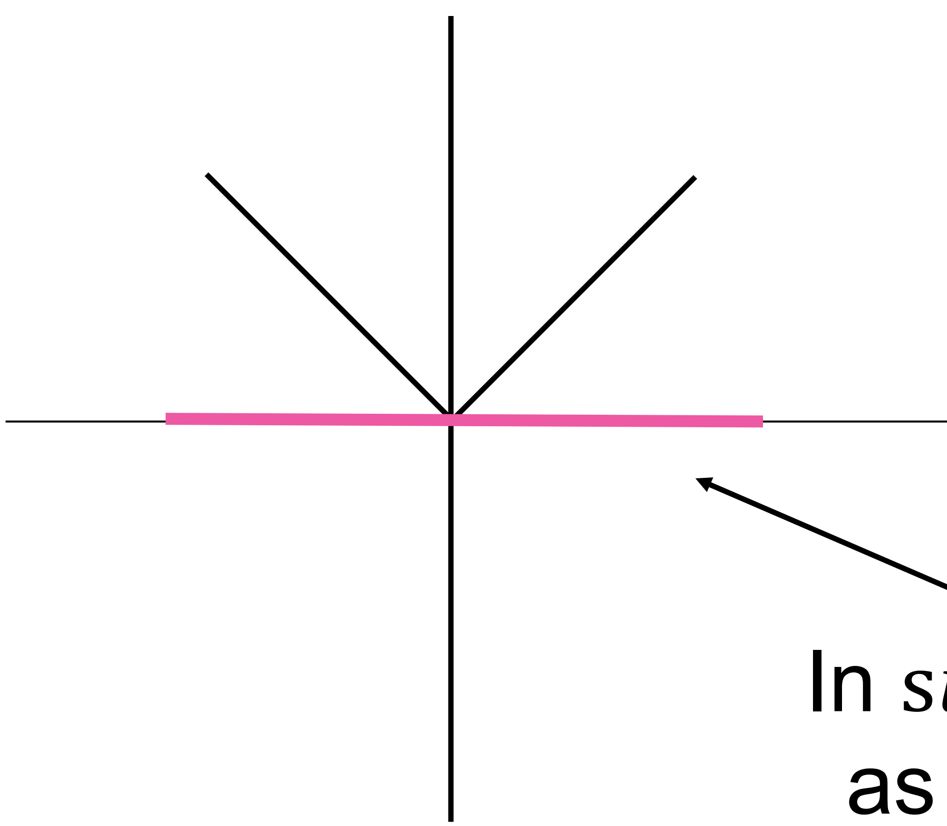


$$\tilde{E}(\theta) = \frac{1}{N}(y - z\theta)^{\mathrm{T}}(\mathrm{y} - z\theta) + \lambda\|\theta\|_1$$

$$\frac{\partial \tilde{E}(\theta)}{\partial \theta} = -z^T(y - z\theta) + \frac{\partial(\lambda\|\theta\|_1)}{\partial \theta}$$

Using Sub-gradient

$$\boxed{\frac{\partial \tilde{E}(\theta)}{\partial \theta} = -z^T(y - z\theta) + \lambda sign(\theta)}$$



In $sign$ function, we use this sub-gradient line as our under-estimator (below our function)

Outline

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

Leave-One-Out Cross Validation

LOOCV

① $\delta = 0.001$

② $\delta = 0.01$

For every $i = 1, \ldots, n$:

- train the model on every point except i, ③ $= \delta\, 0.01$
- compute the test error on the held out point.

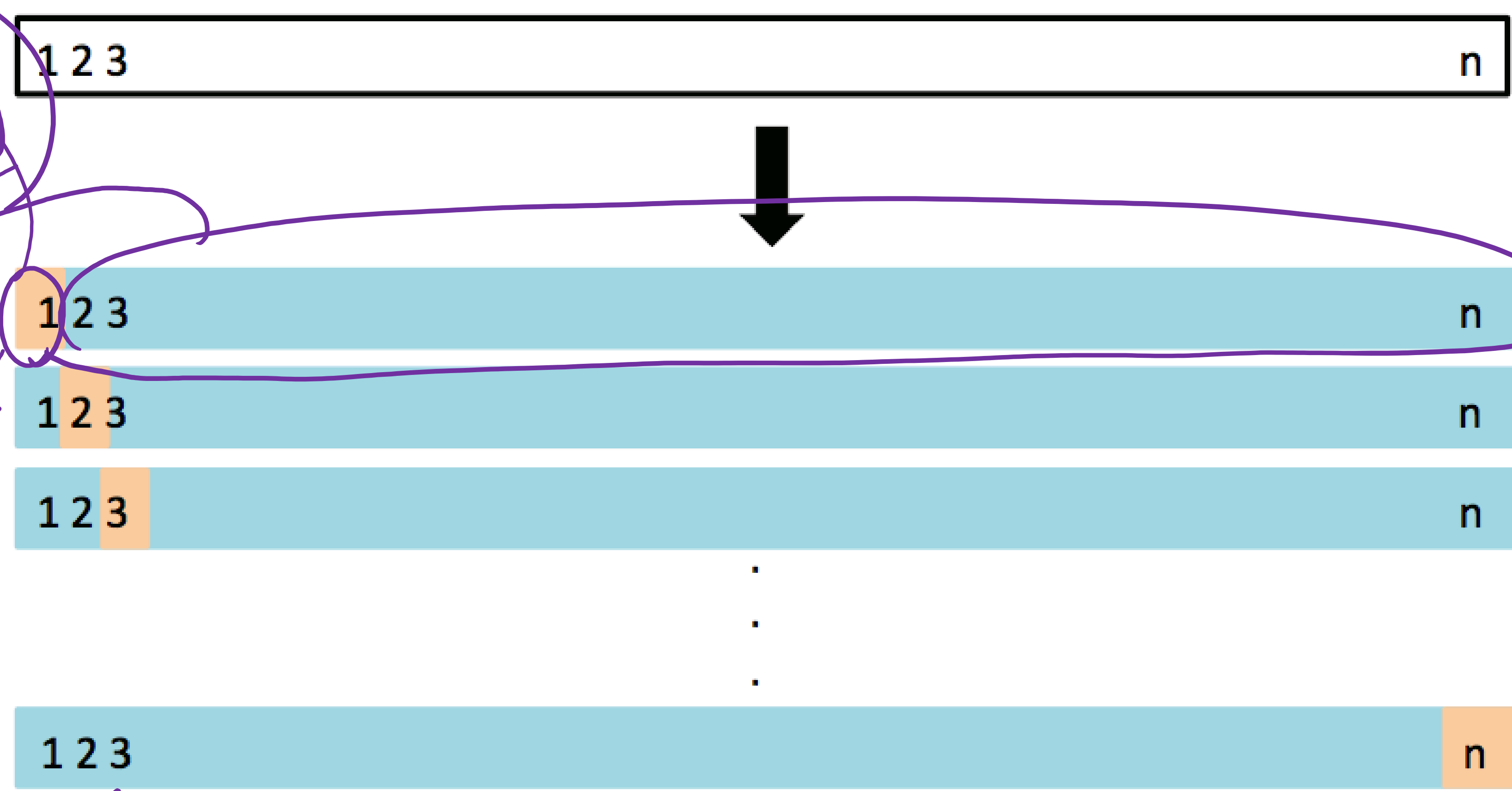
Average the test errors. $$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{(-i)})^2$$

$\hat{y}_p = Z \cdot \Theta$

$\Theta = (X^TX + \delta I)X^Ty$

$E(\Theta) = \frac{1}{N}\sum_{i=1}^{N}(y_a - \hat{y}_p)$

$E_2(\Theta)$



$(E_1(\Theta) + E_2(\Theta) + \dots)/100 = E(\Theta)$ for $\delta = 0.001$

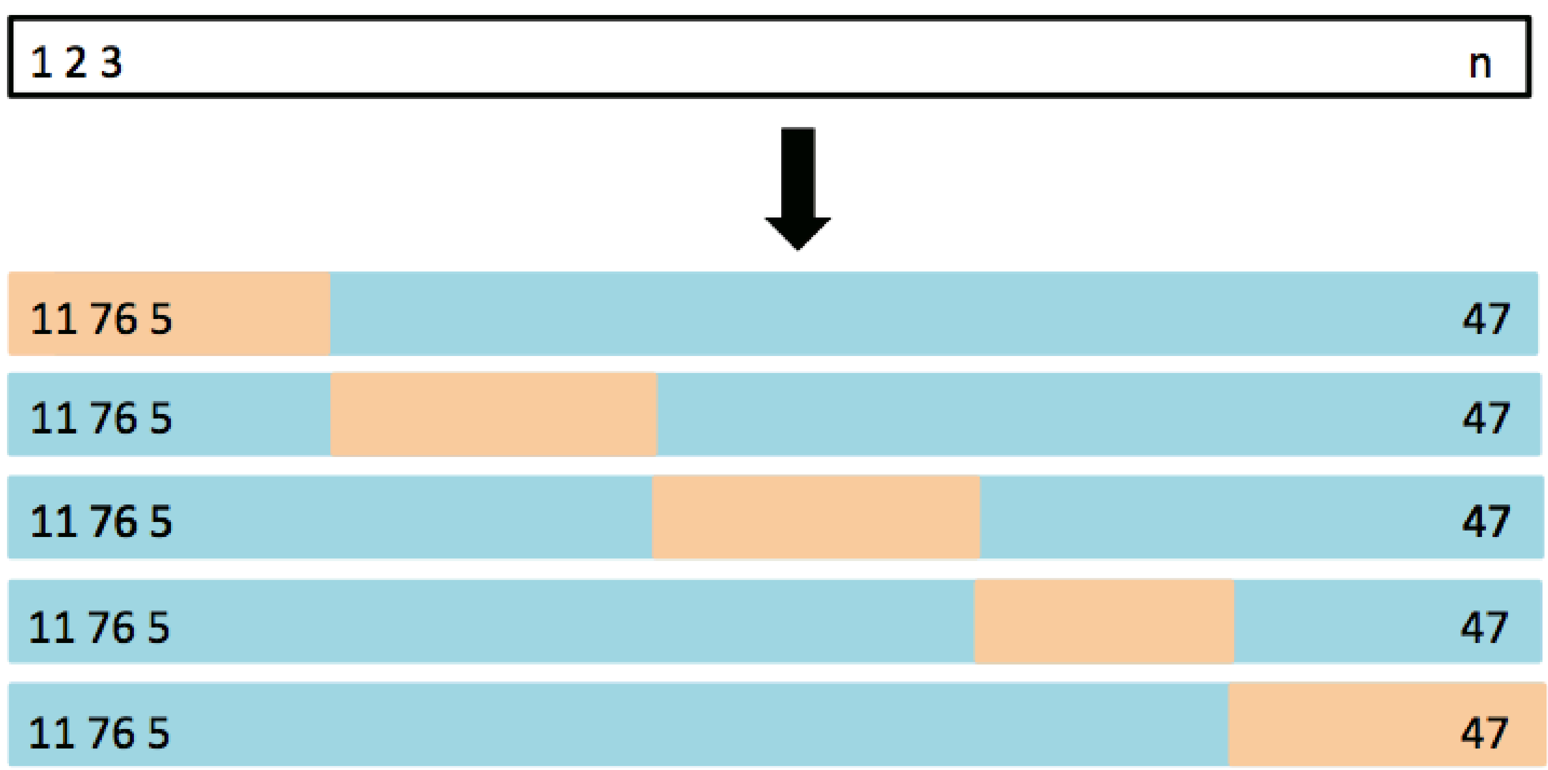
K-Fold Cross Validation

Split the data into k subsets or *folds*.

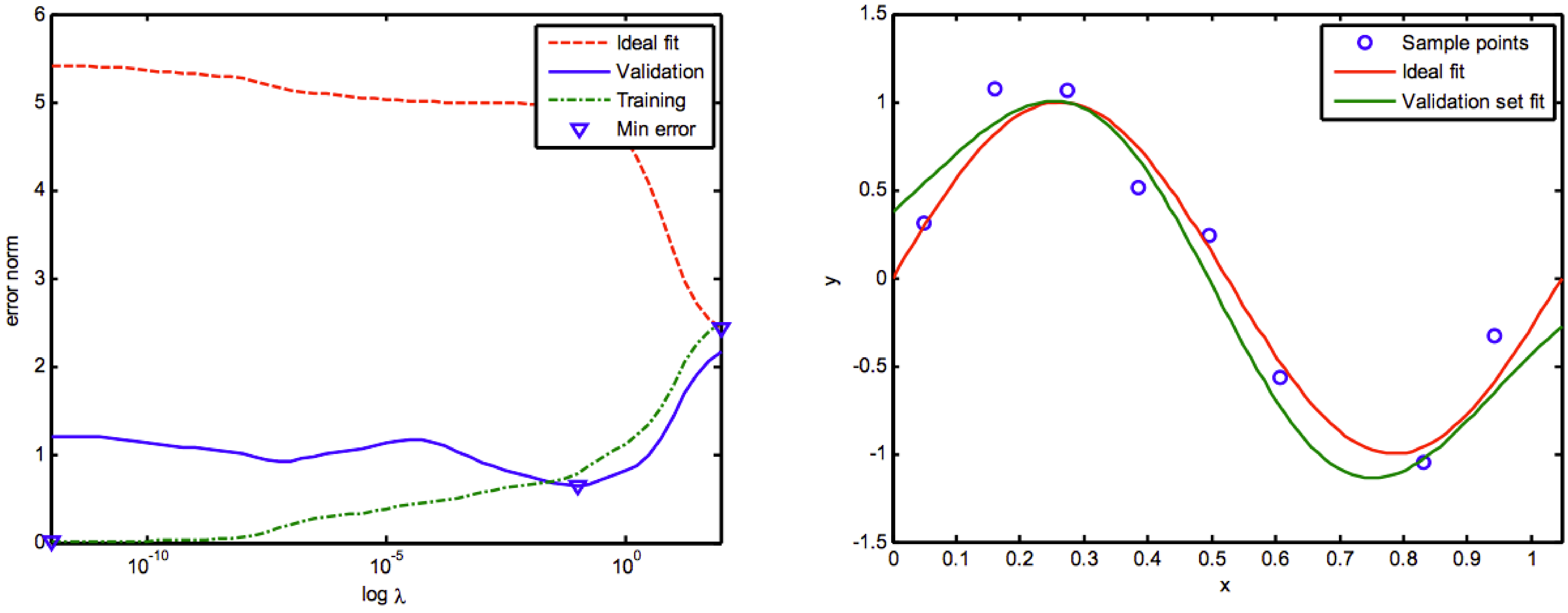
For every $i = 1, \ldots, k$:

- train the model on every fold except the ith fold,
- compute the test error on the ith fold.

Average the test errors.



Choosing λ Using Validation Dataset



Pick up the lambda with the lowest mean value of rmse calculated by Cross Validation approach

Take-Home Messages

- What is overfitting
- What is regularization
- How does Ridge regression work
- Sparsity properties of Lasso regression
- How to choose the regularization coefficient λ