Georgia Tech

Machine Learning CS 4641-7641

Regularized Linear Regression

Dr. Nimisha Roy
Georgia Tech

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

Georgia Tech

Outline

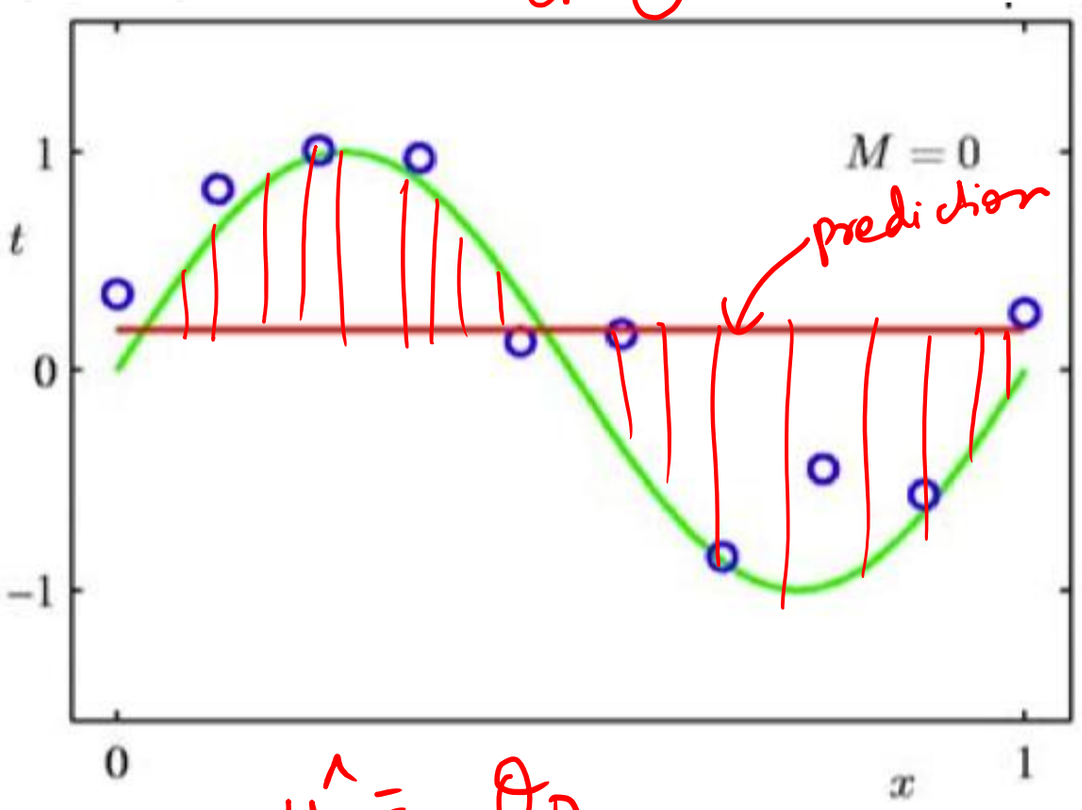
- Overfitting and regularized learning
- Ridge regression
- Lasso regression
- Determining regularization strength

Recap: Overfitting vs Underfitting

z domain

$$\hat{y}_p = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \cdots + \theta_d x_1^d$$

d=0

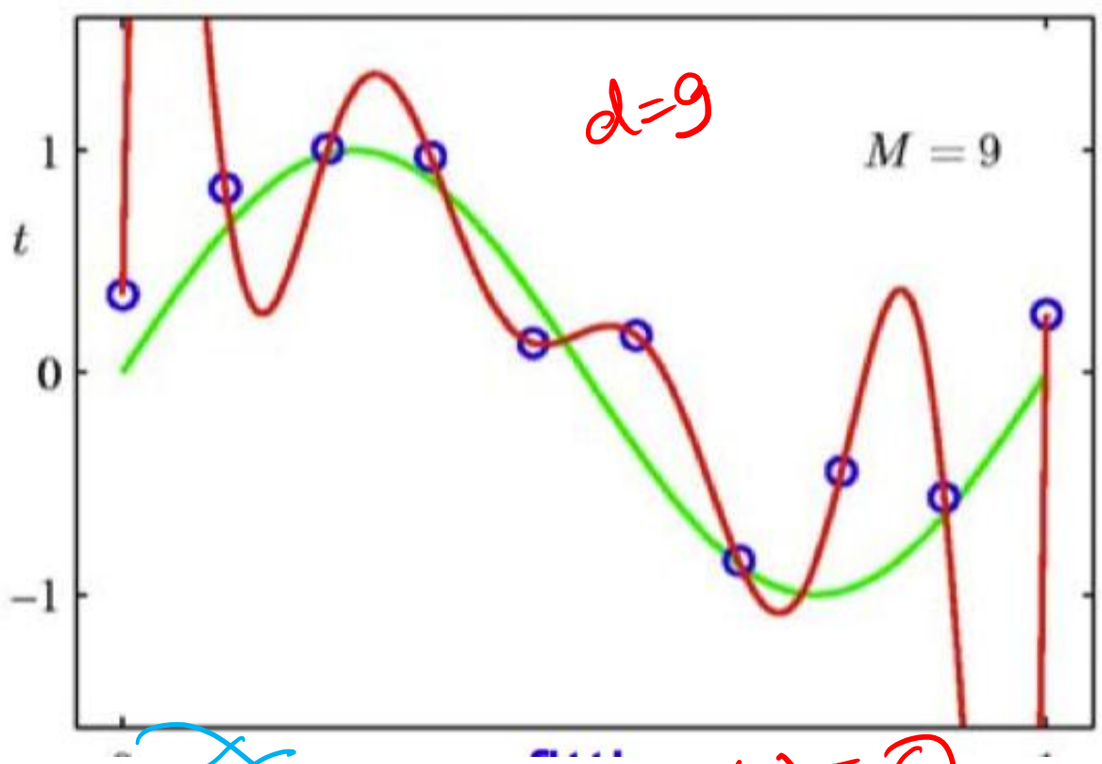


$\hat{y}_p = \theta_0$

High Bias

Bias: $\left(E[\hat{y}_p] - y_a\right)$

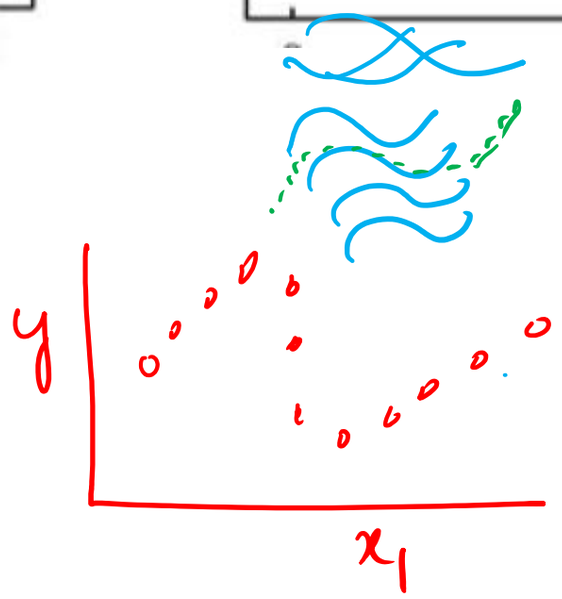
d=9



$E(\theta) = 0$ (training)

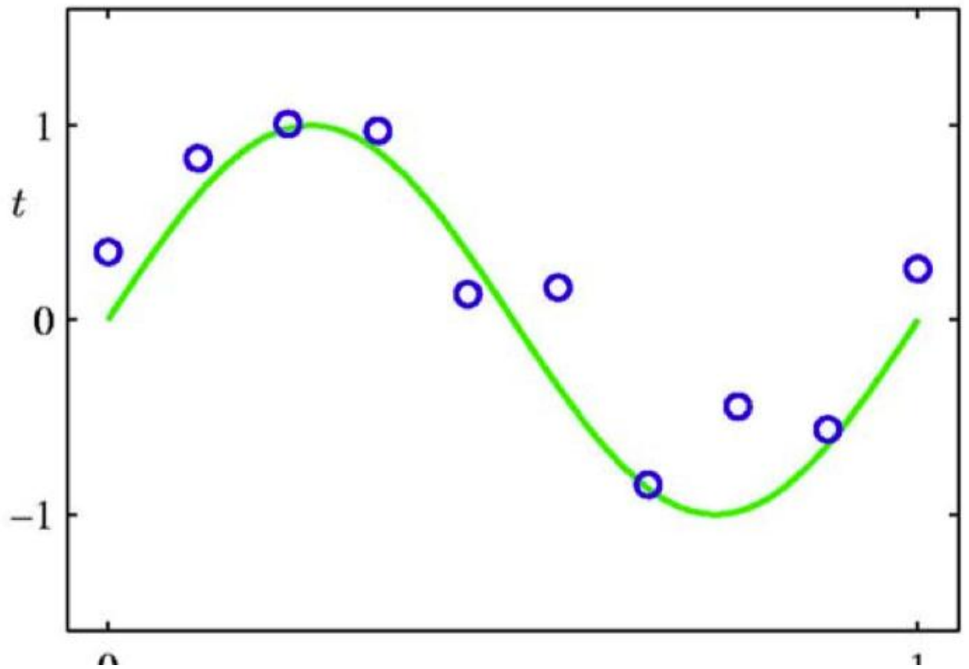
High Variance

$E\left[\left(\hat{y}_p - E[\hat{y}_p]\right)^2\right]$



$$E(\theta) = \text{bias}^2 + \text{variance}$$

Regression: Why is overfitting more common?



Free of Cost

Engineering Features to get to more features in z space

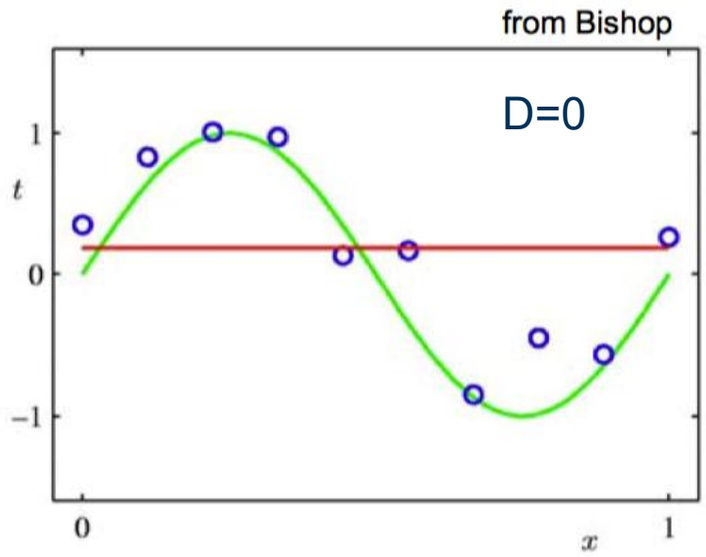
- Want to fit a polynomial regression model

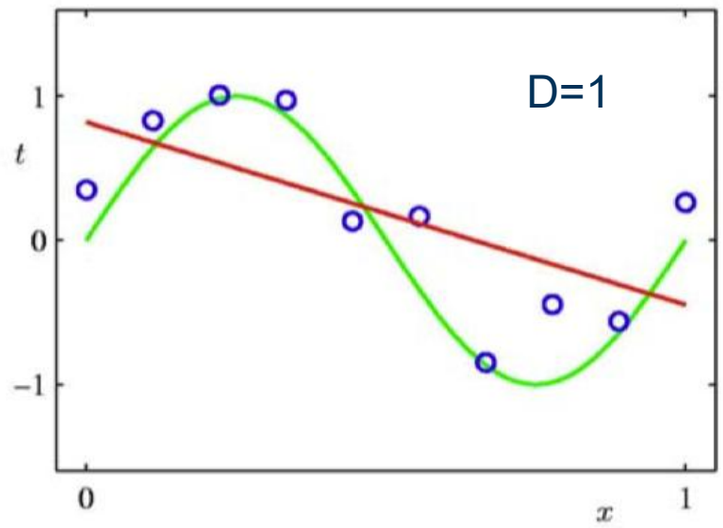
$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d + \epsilon$$

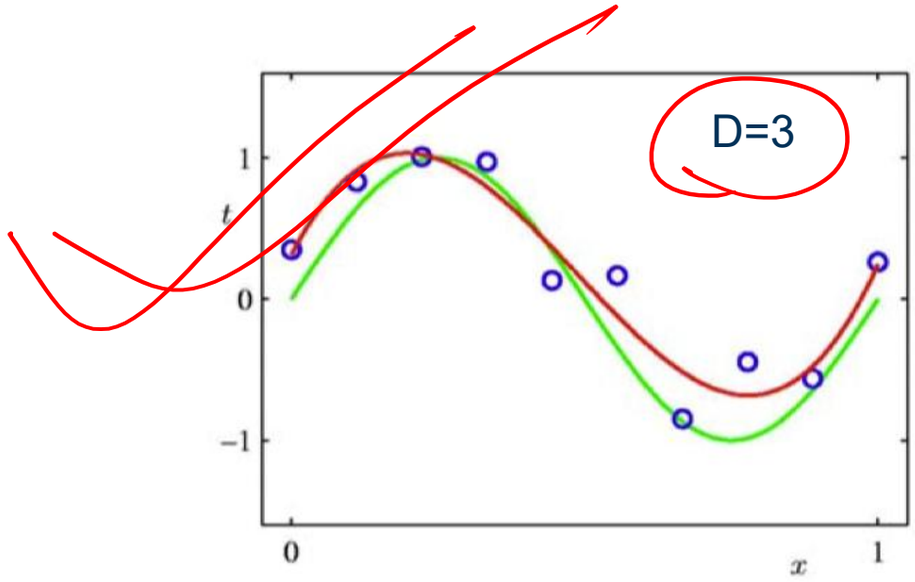
- $z = \{1, x, x^2, \ldots, x^d\} \in R^d$ and $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_d)^T$

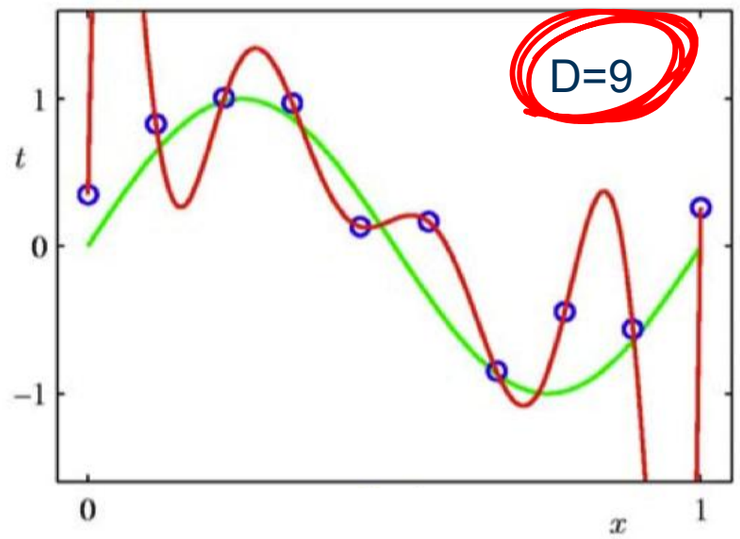
$$y = z\theta$$

Which One is Better?





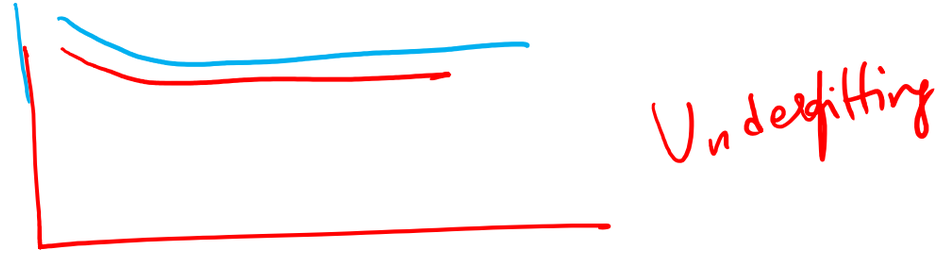


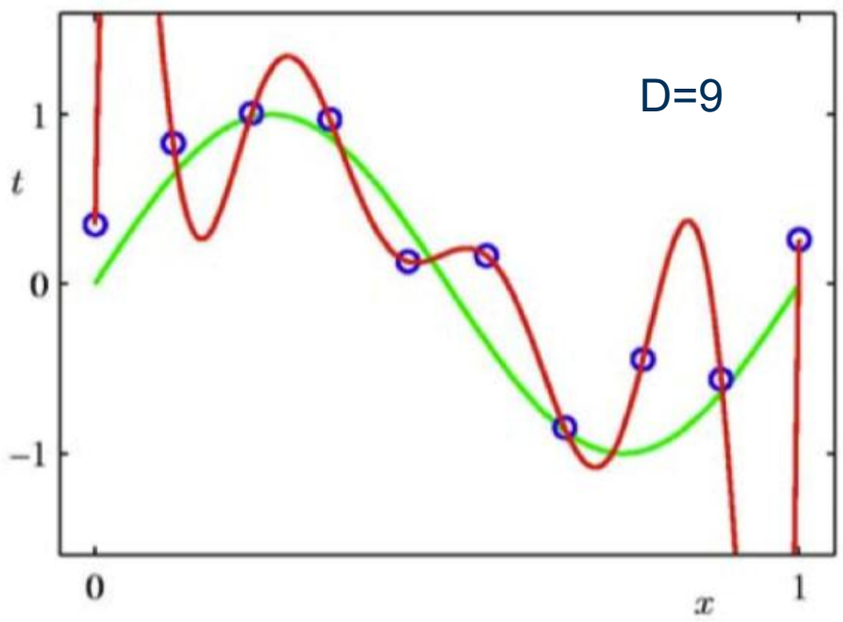


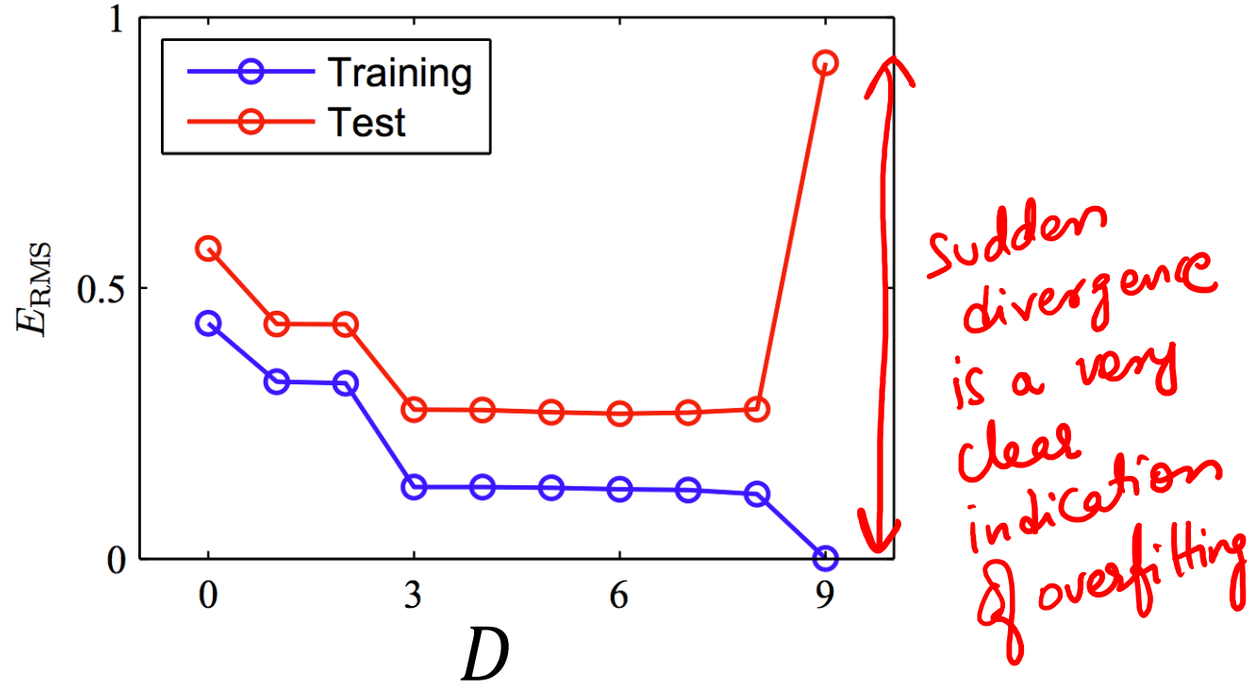
- Can we increase the maximal polynomial degree to very large, such that the curve passes through all training points?

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

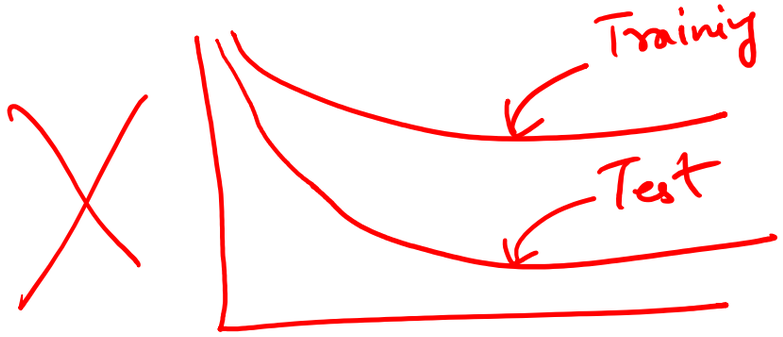
The Overfitting Problem







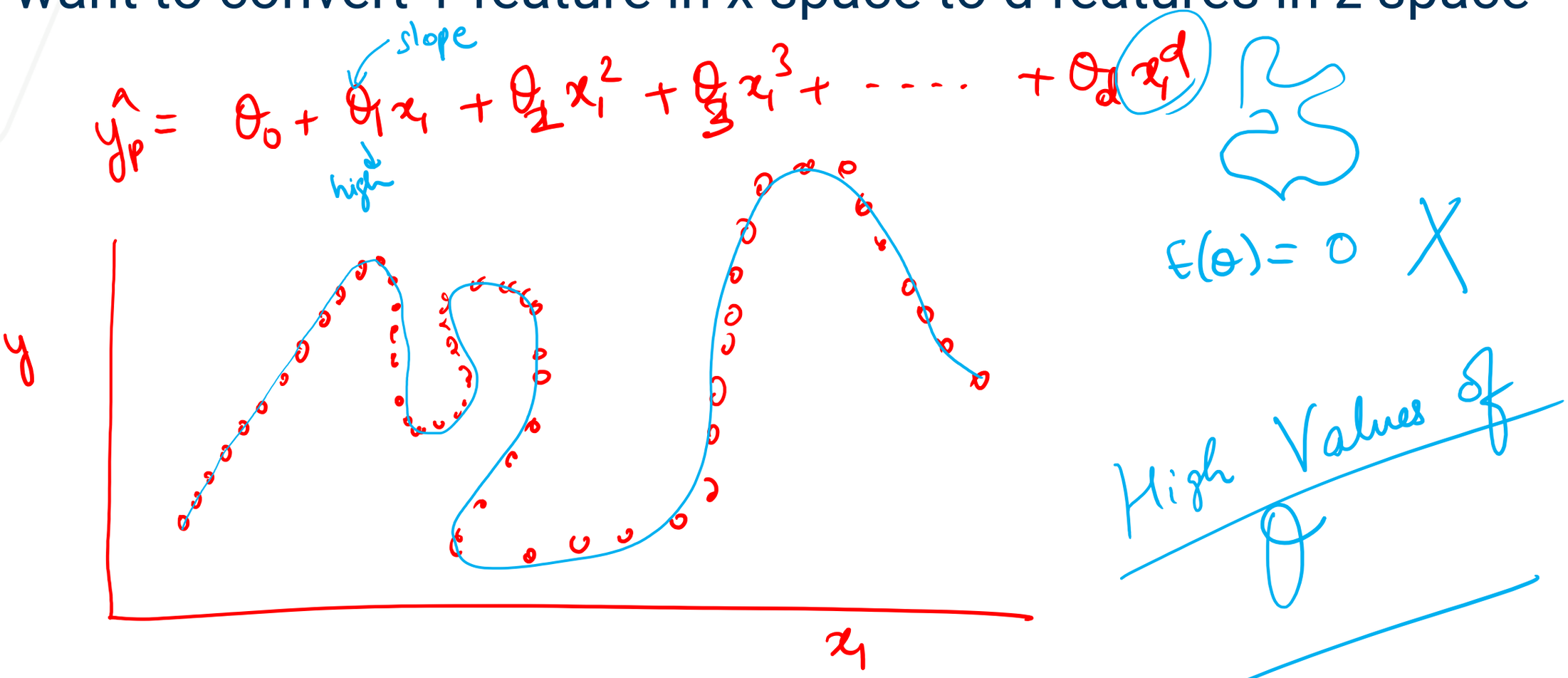
- The training error is very low, but the error on test set is large.
- The model captures not only patterns but also noisy nuisances in the training data.



Georgia Tech

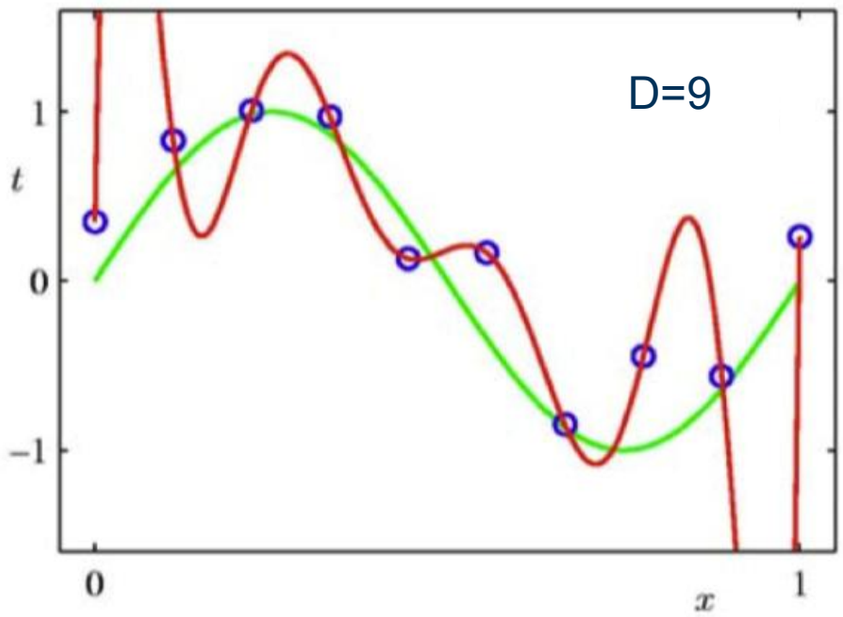
Root cause of overfitting?

- I want to convert 1 feature in x space to d features in z space



The Overfitting Problem

Regularization tries to prevent overfitting



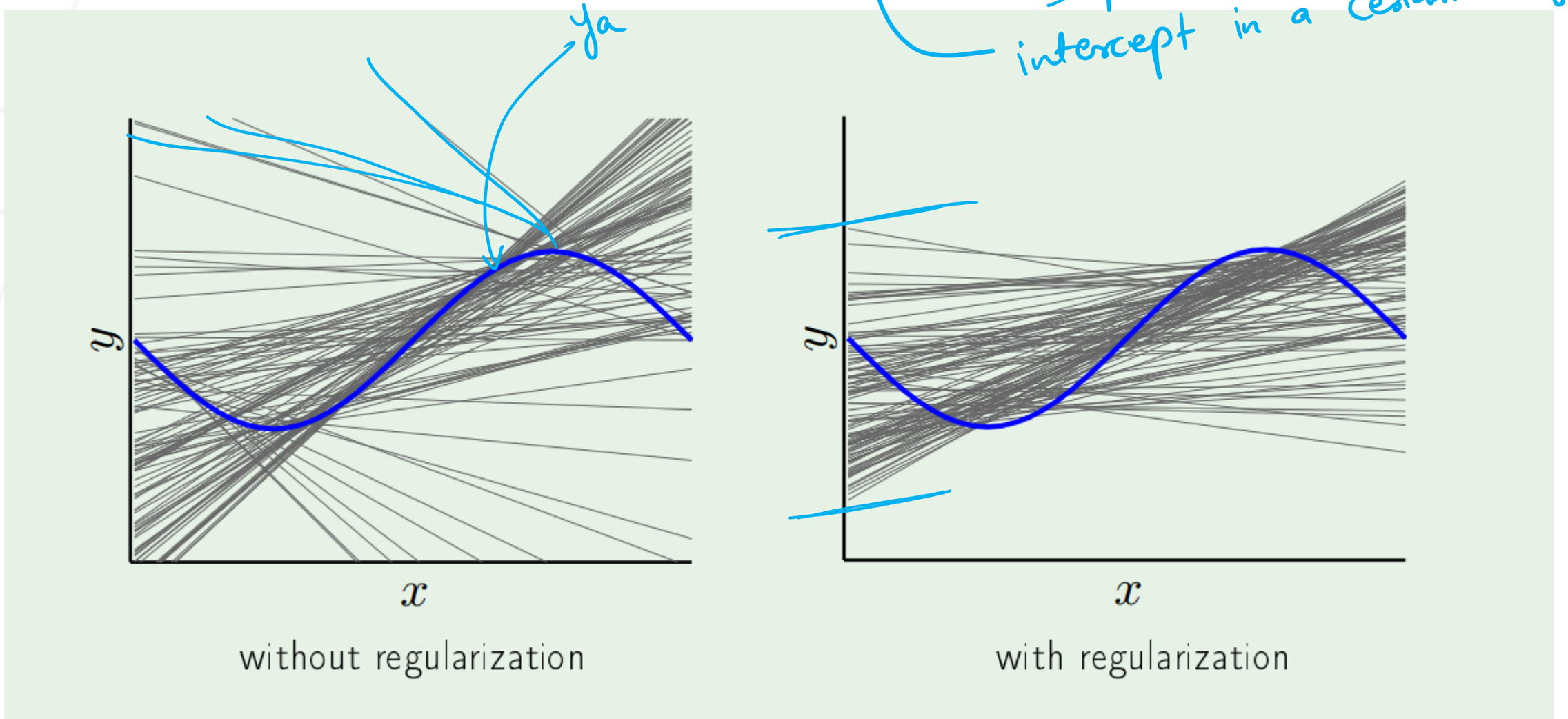
- In regression, overfitting is often associated with large Weights (severe oscillation)
- How can we address overfitting?

Regularization
(smart way to cure overfitting disease)

$\hat{y_p} = \theta_0 + \theta_1 x_1$

slope in a certain range

intercept in a certain range



Put a brake on fitting

Fit a linear line on sinusoidal with just two data points

Georgia Tech

Who is the winner?

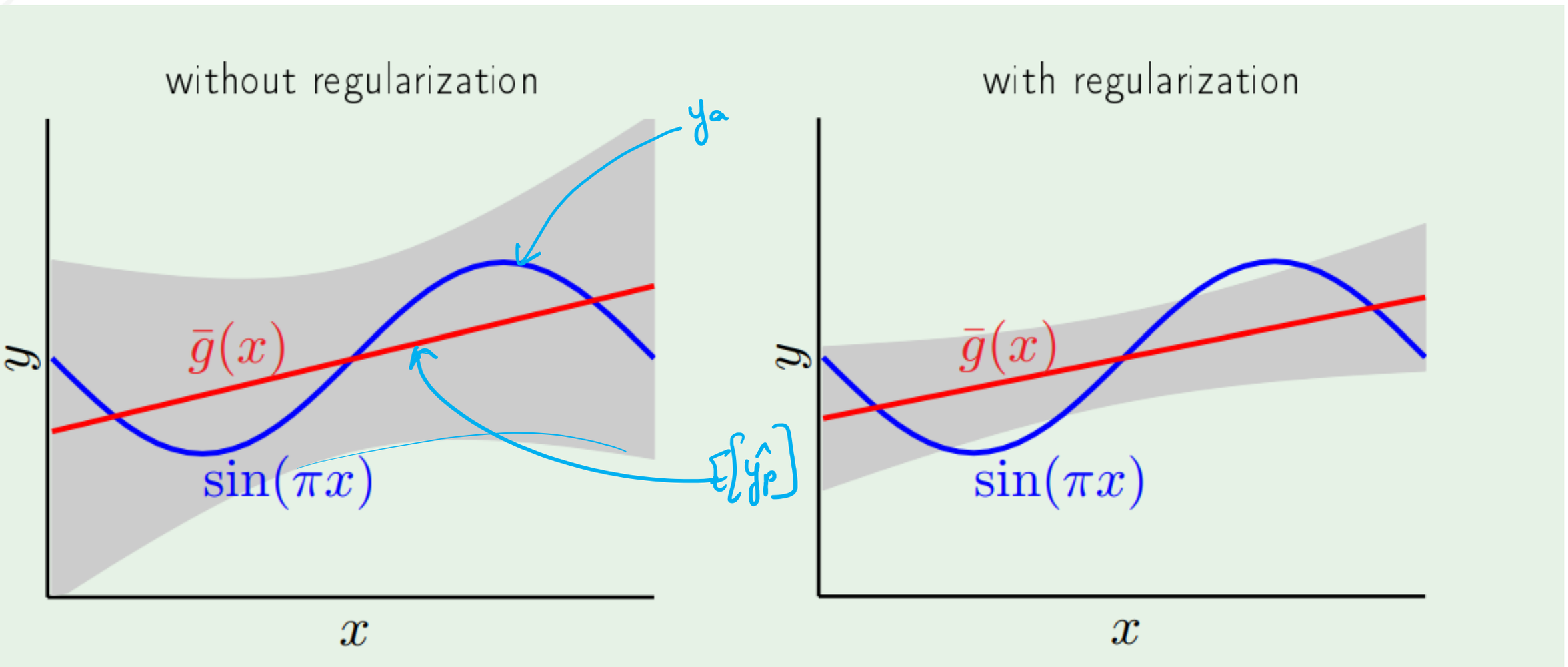
A

$E(\theta) = \text{bias}^2 + \text{variance}$

B

Variance is much lower

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



bias=0.21; var=1.69

bias=0.23; var=0.33

Georgia Tech

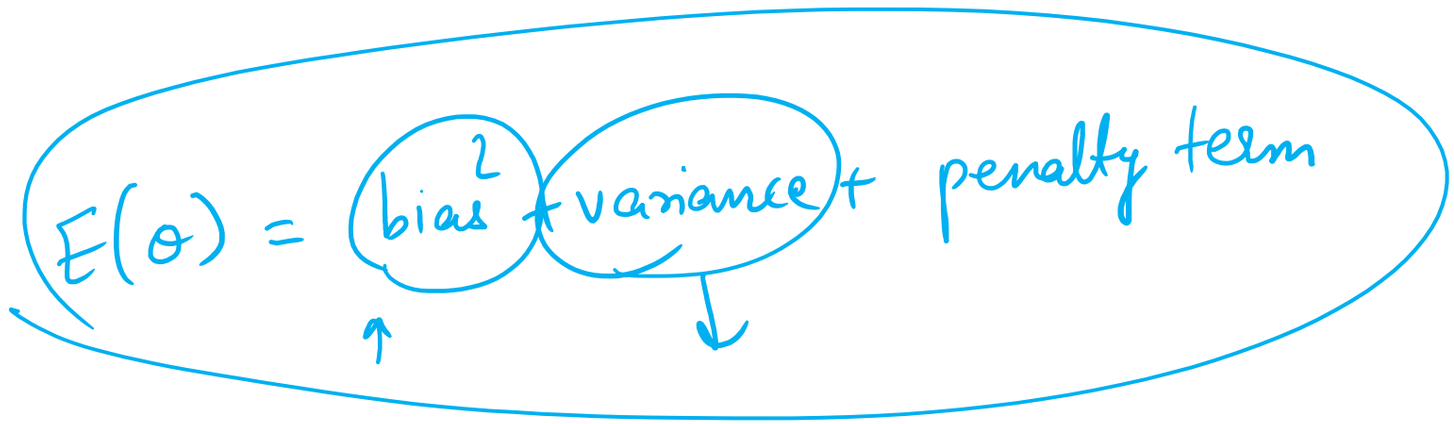
So, what is regularization in general?

$$E(\theta) = \frac{1}{N}\sum_i \left(y^{(i)} - X_{\cdot}^{(i)}\theta\right)^2 + \text{———}$$

Regularization adds a penalty term to the error or objective function:

- The penalty discourages large weights → the model becomes *simpler*.
- A simpler model can't bend perfectly to fit all training data → **bias increases slightly.**
- But since it's not so sensitive to the quirks of any one dataset → **variance decreases significantly.**
- That tradeoff leads to better **generalization**.

Georgia Tech



Regularization	Bias	Variance	Total Error	Stability
❌ Without	Low	**High**	High	Overfits
✅ With	Slightly higher	**Much lower**	Lower	Generalizes better

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:

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

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

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

$$y = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \cdots + \theta_d z_d$$

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

subject to

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

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

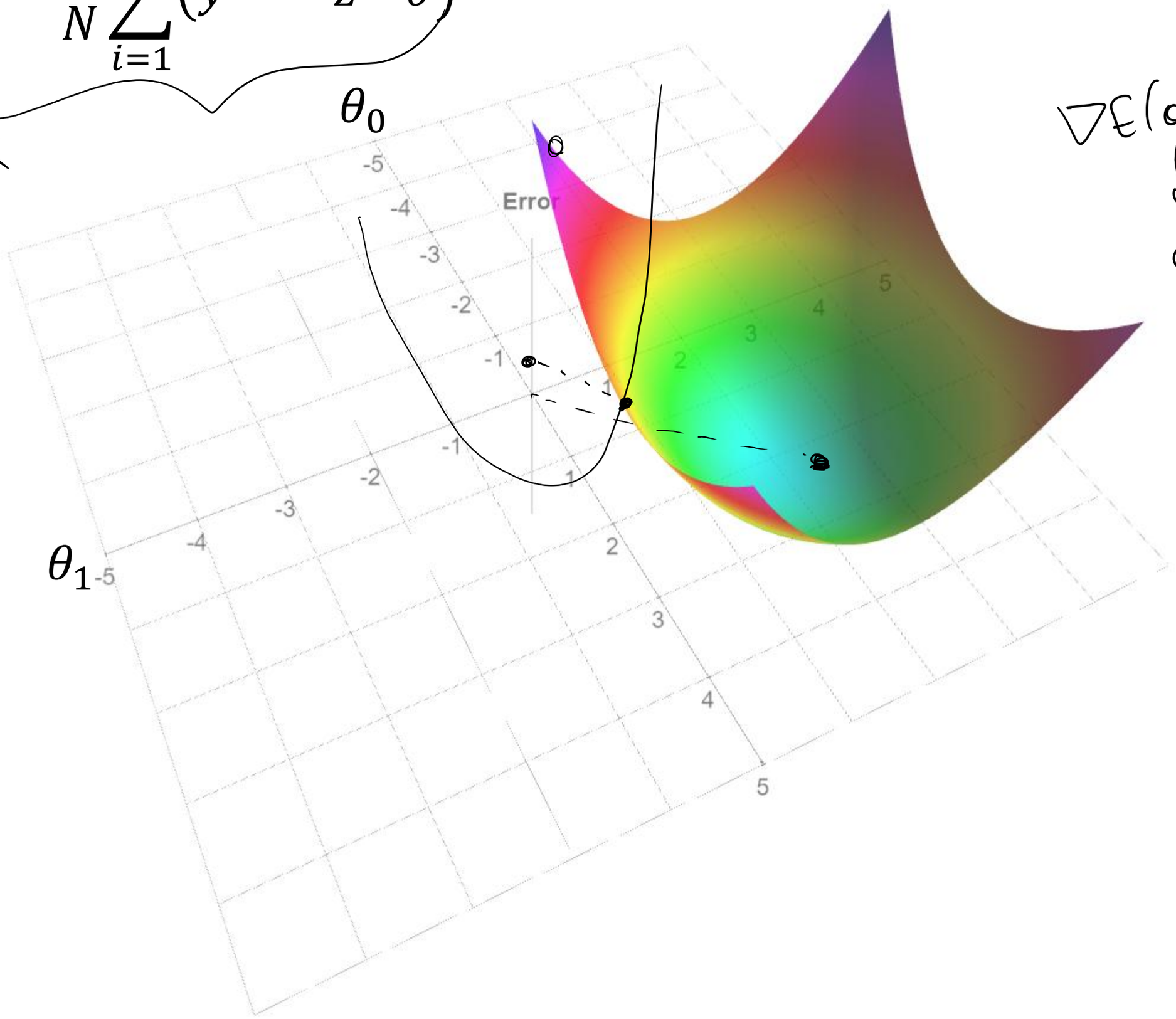
Georgia Tech

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

convex

θ_0

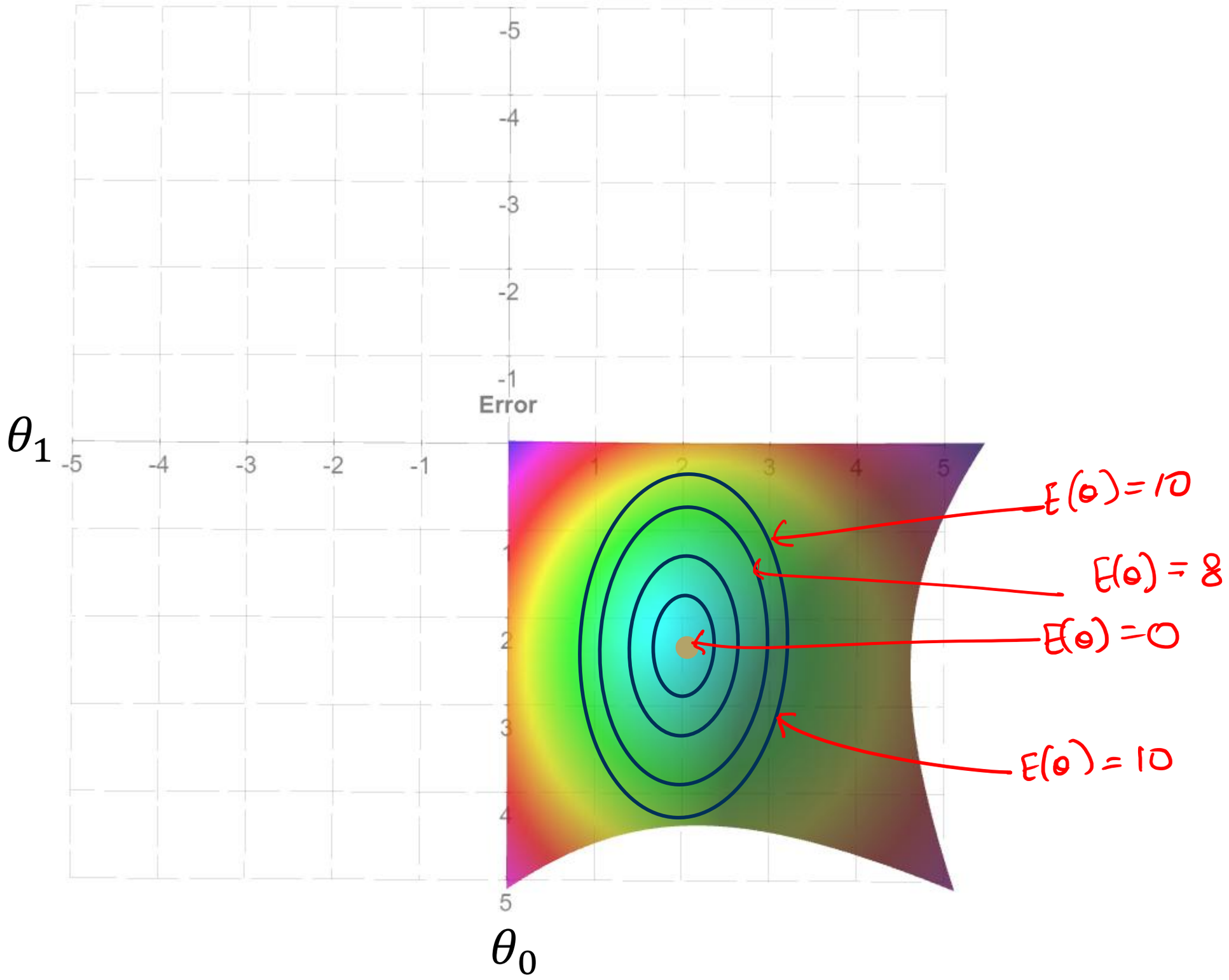
θ_1



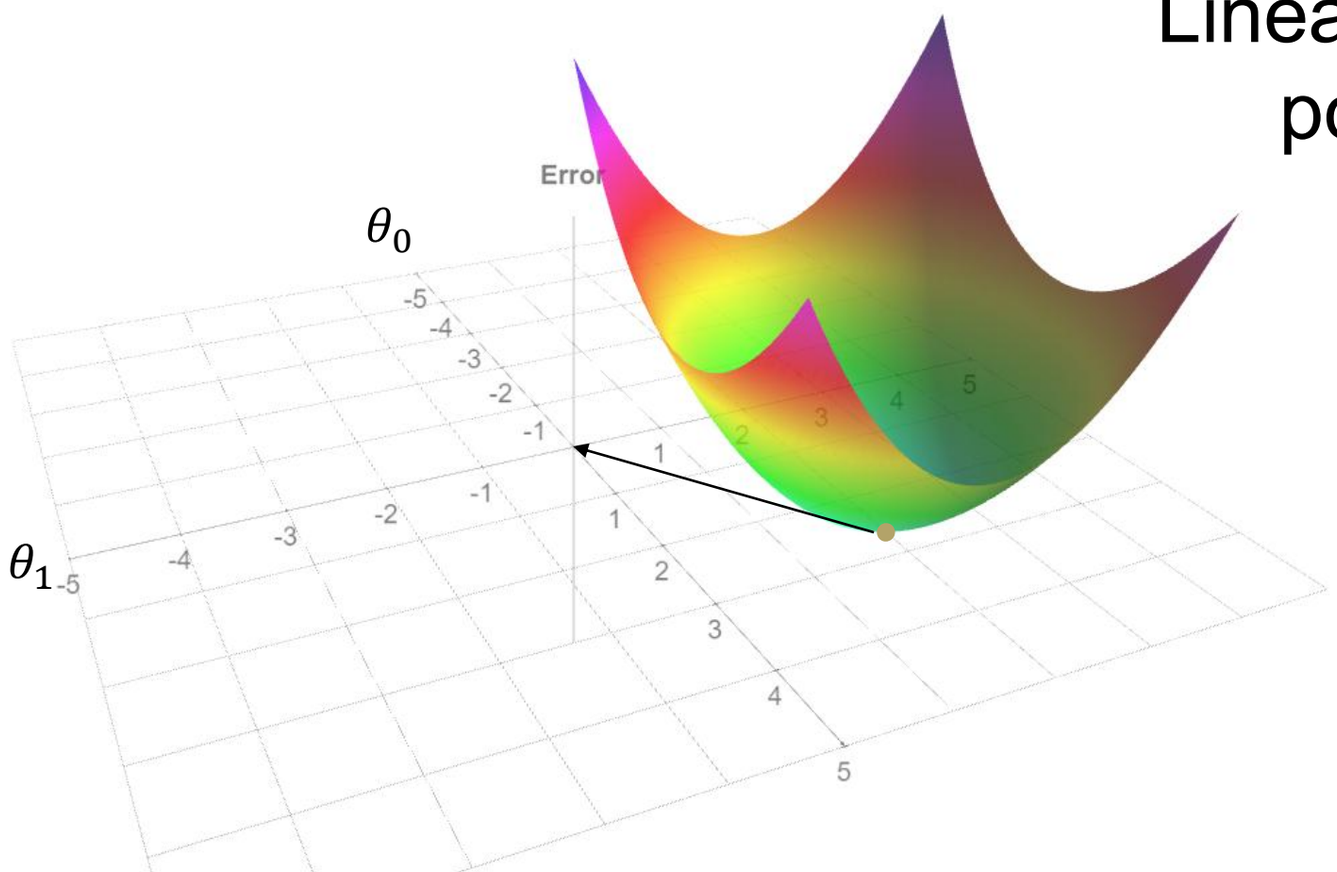
$\nabla E(\theta) = 0$

$\theta = 0$

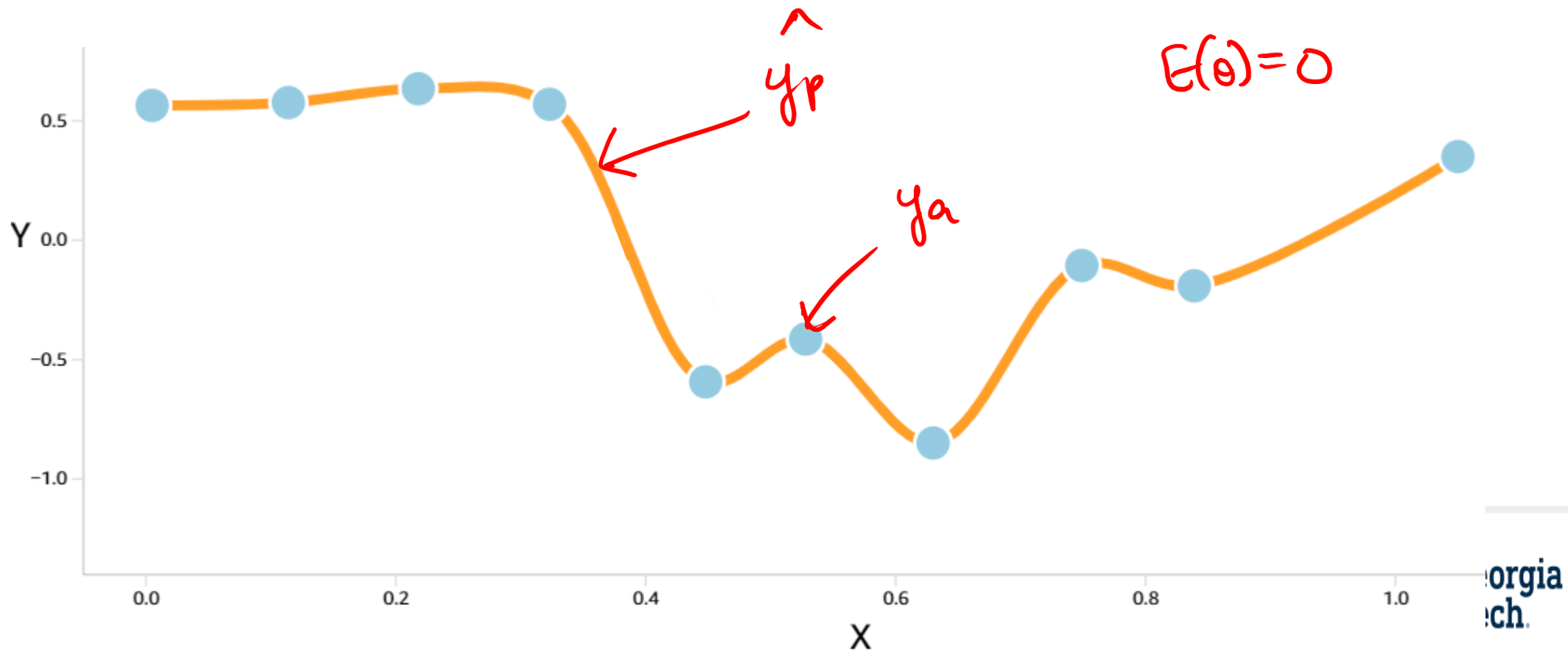
Project the same graph on x-y using contour plot

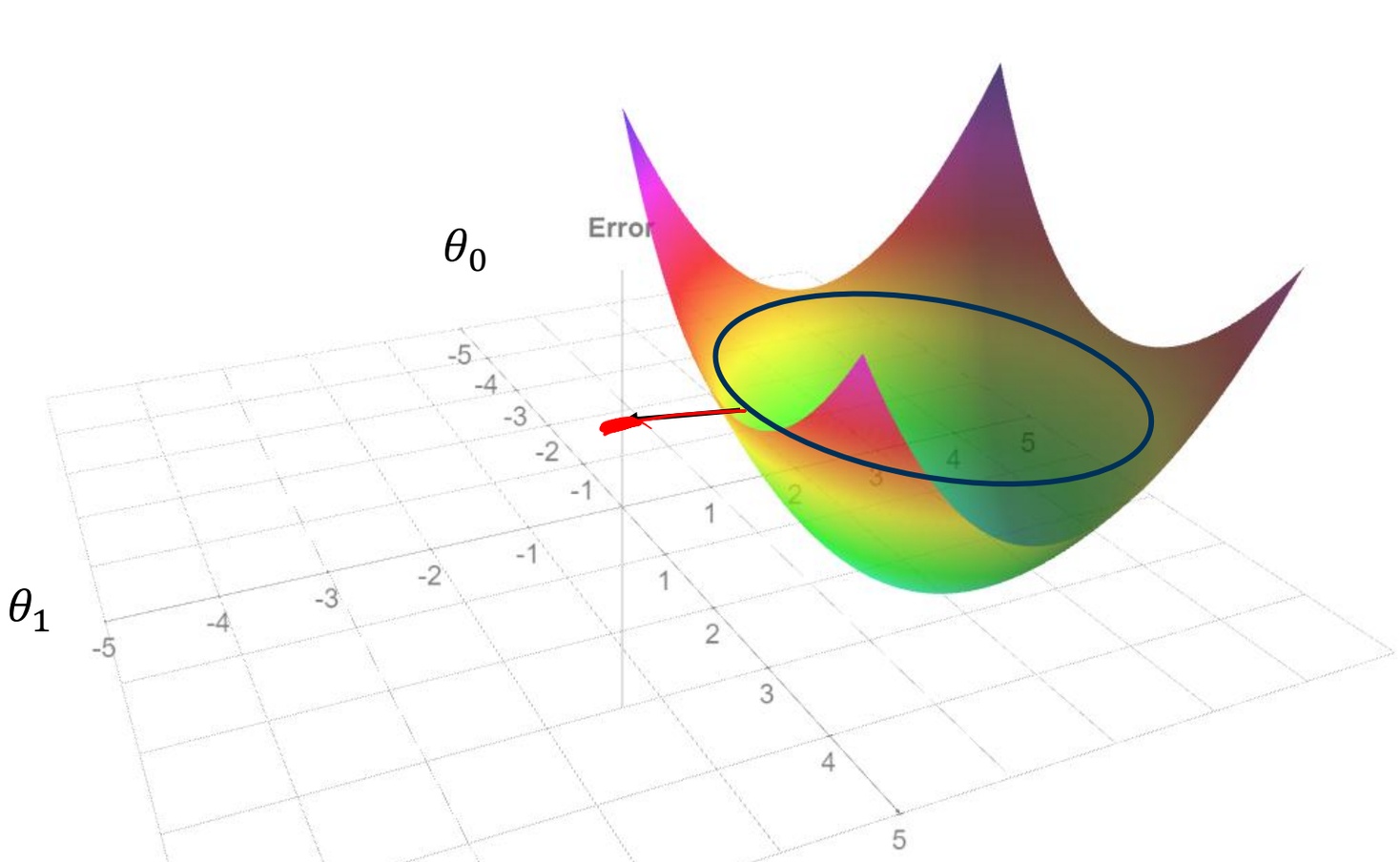


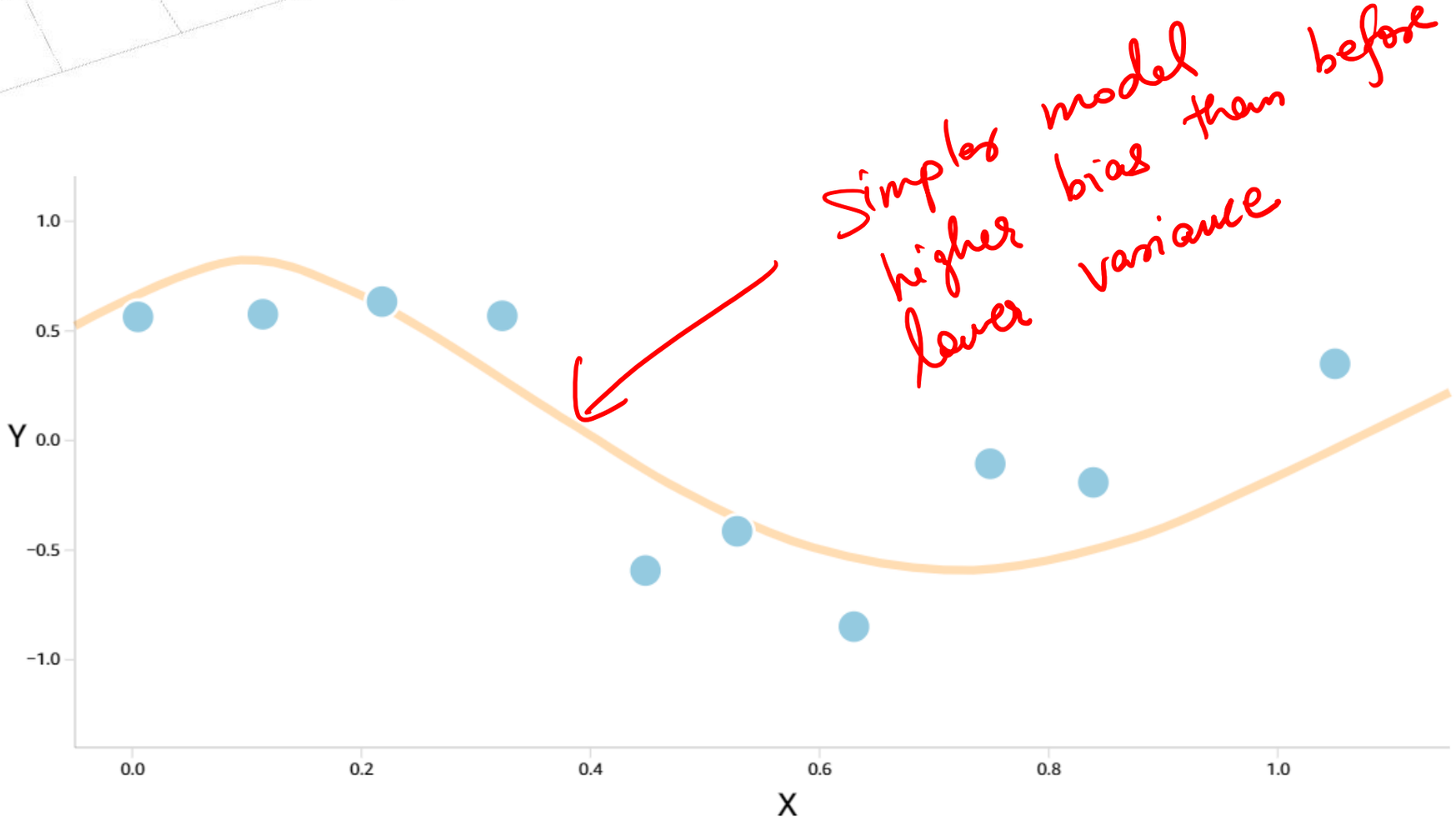
Linear regression with a very high polynomial degree solution



$E(\theta) = \text{bias}^2 + \text{variance}$




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


Simpler model
higher bias than before
lower variance
1.0
0.5
Y 0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
X

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

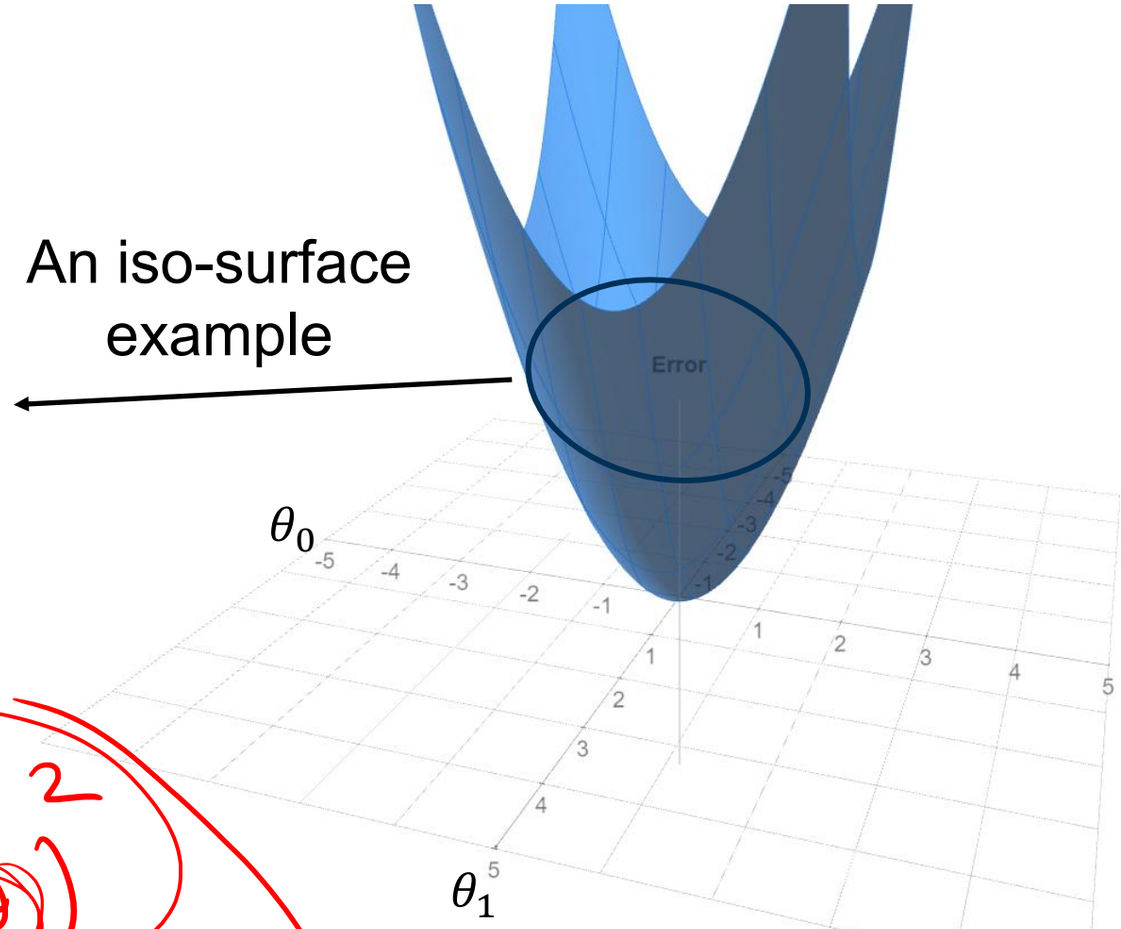
We need to introduce a constraint

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

$g(\theta) = \theta^T\theta - C = 0$

$$E(\theta) = \frac{1}{N}\sum_i \left(y_a^{\{i\}} - X^{\{i\}}\theta\right)^2 + \lambda \cdot g(\theta)$$
$$+ \lambda\theta^T\theta$$

An iso-surface example

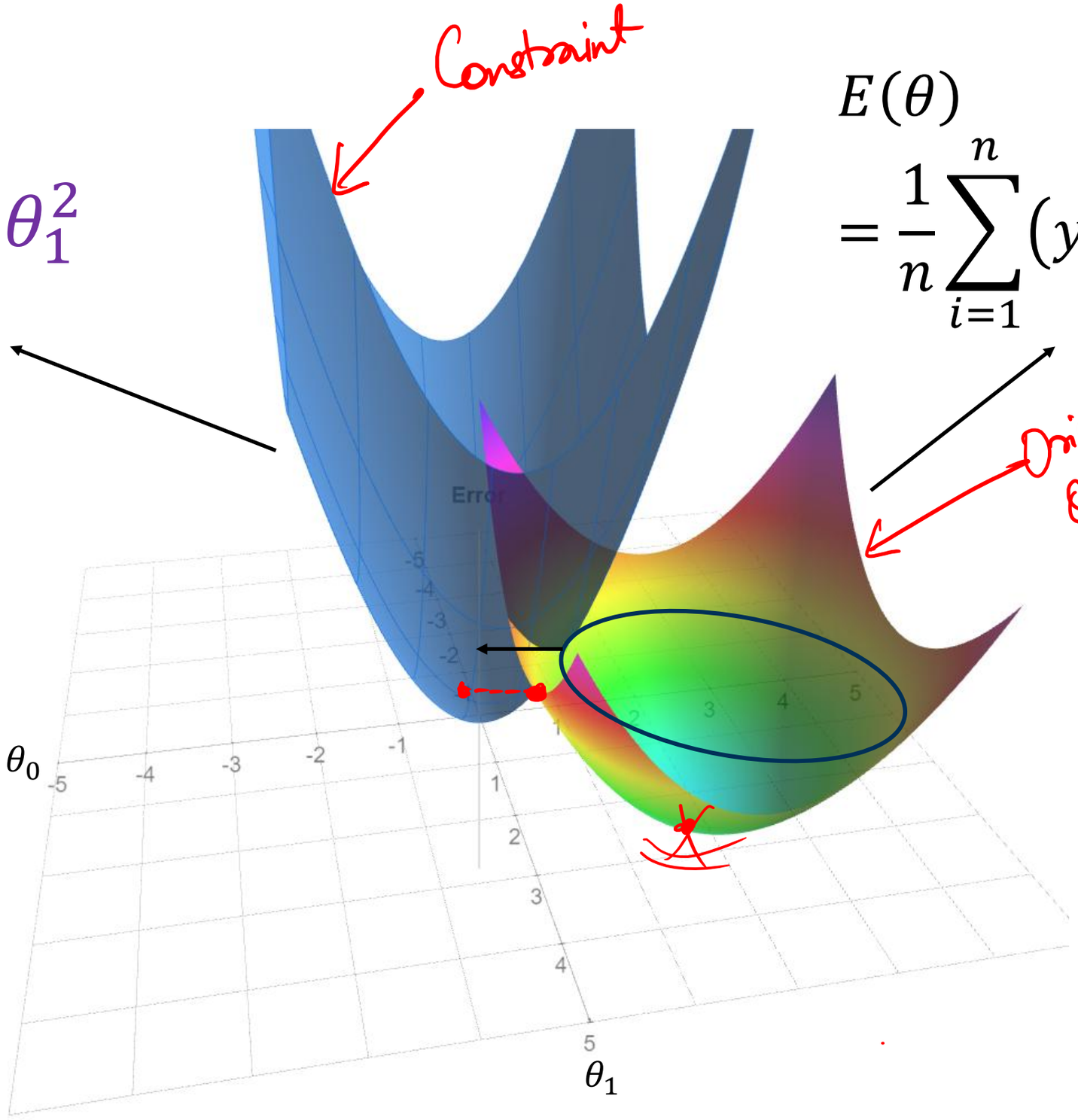


Georgia Tech

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

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

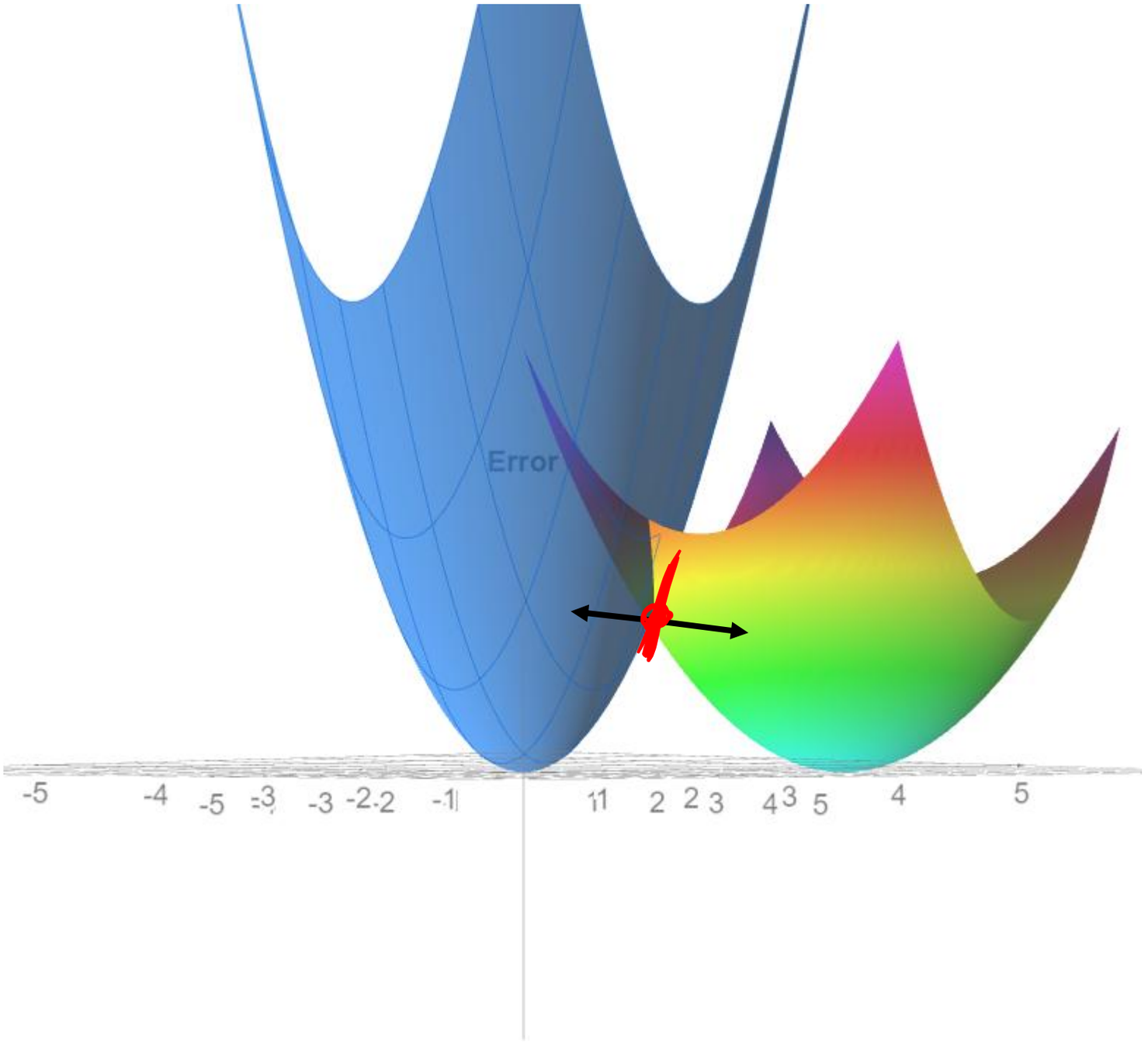
$$L(\theta, \lambda) = E(\theta) + \lambda \theta^T \theta - \lambda C$$

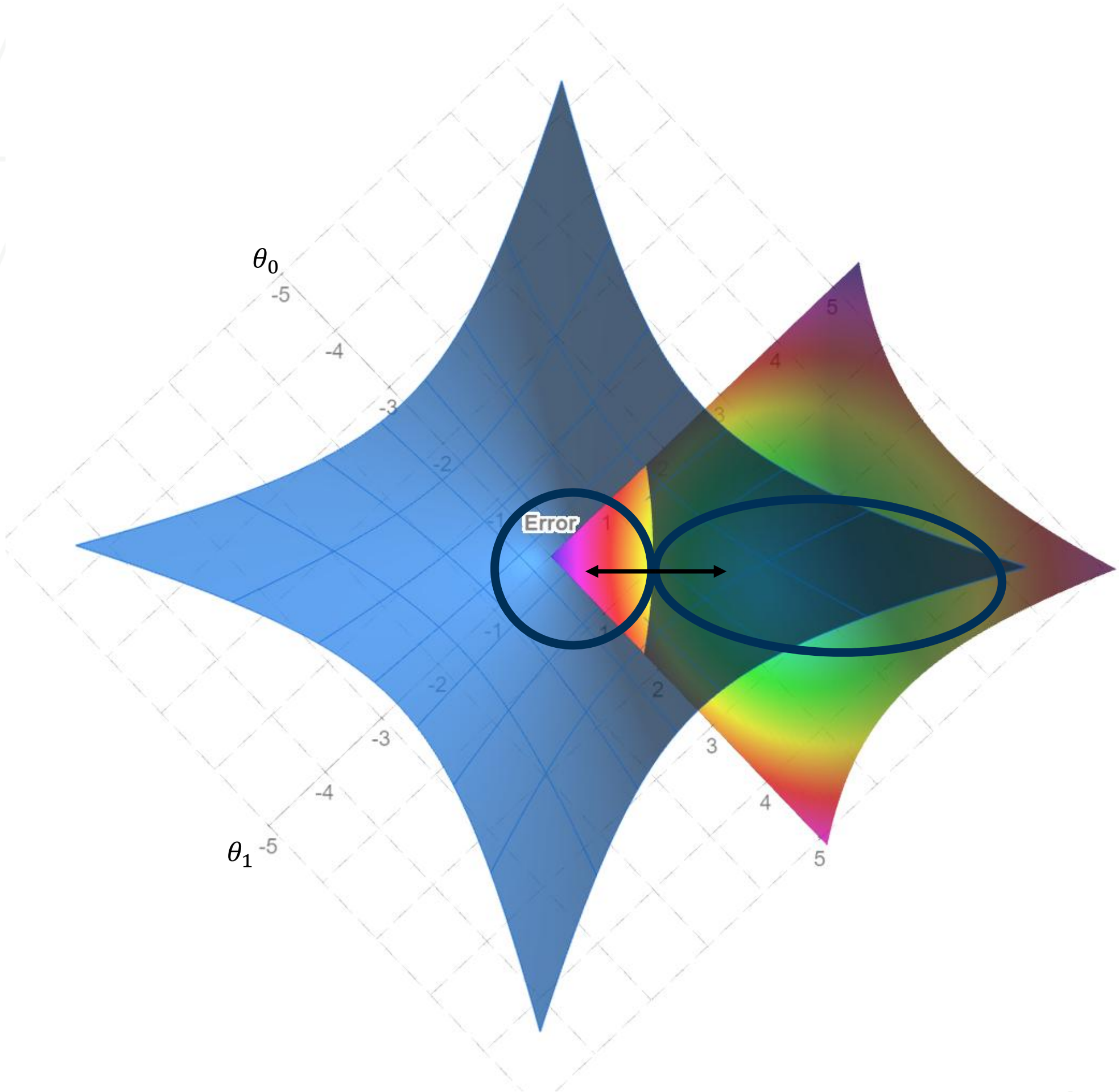
$$\nabla L(\theta, \lambda) = 0 \qquad \nabla [E(\theta) + \lambda \theta^T \theta - \lambda C] = 0$$

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

Georgia Tech

How to enforce the gradient of Lagrange function to be zero





Georgia Tech

$$L(\theta, \lambda) = E(\theta) + \lambda \cdot \theta^T\theta - \boxed{\lambda C}$$

Implicit Sol.

$\frac{\partial L}{\partial \theta} = 0$

θ will depend on λ

$\frac{\partial L}{\partial \lambda} = 0$

& λ will depend on θ

$\lambda \longleftarrow$ hyperparameter. λ is a constant

$\frac{\partial L}{\partial \theta} = 0$

$$\boxed{\nabla(E(\theta)) + \lambda \nabla(\theta^T\theta) = 0}$$

Georgia Tech

Let's calculate the gradients

$\tilde{E}(\theta) = E(\theta) + \lambda\theta^T\theta$

Regularized Error

Penalty OR Regularization term

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

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

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

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

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

Let's do integration

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

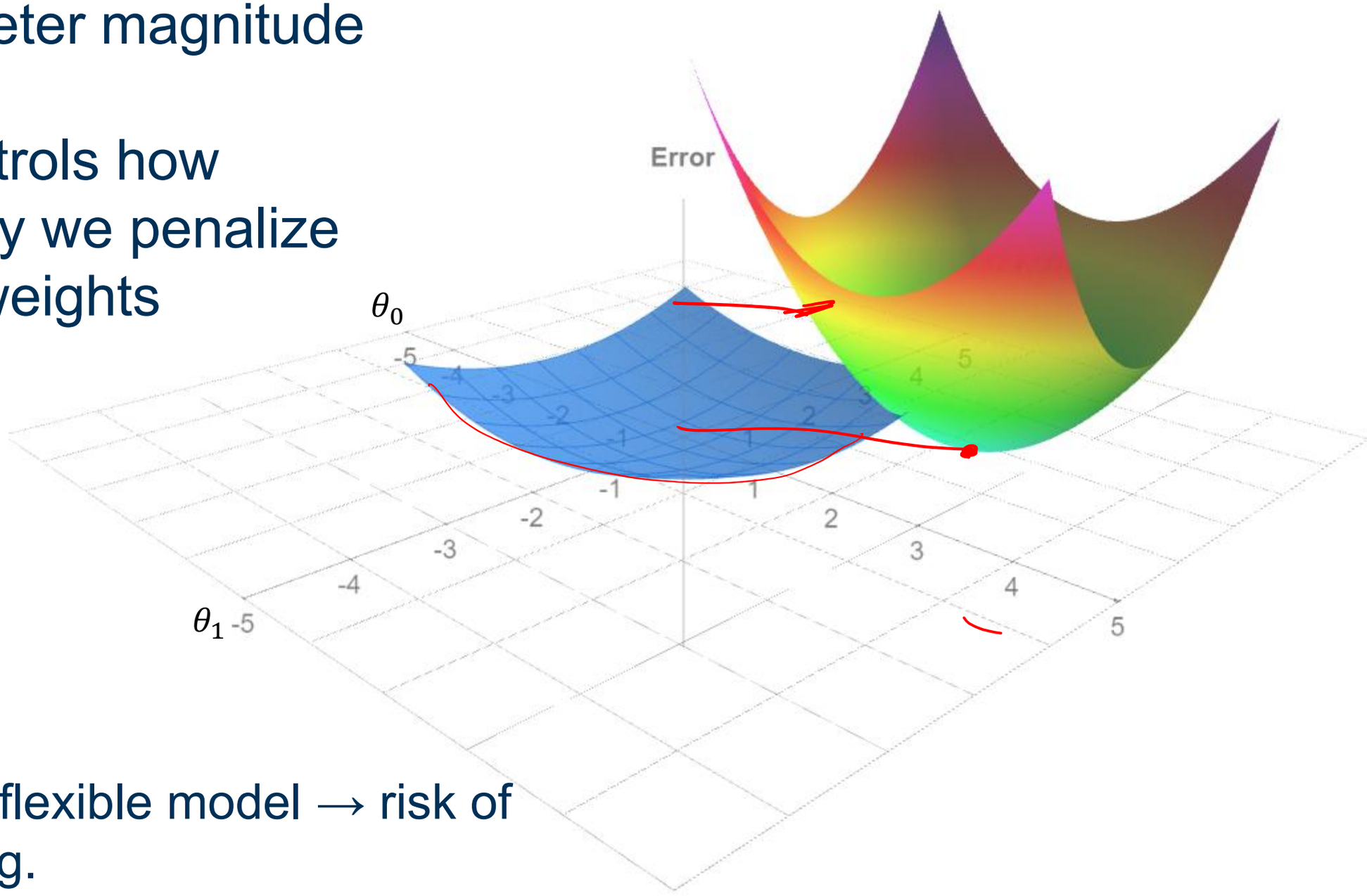
is an extremely imp. hyperparameter

The effect of low Lambda

$\frac{\lambda}{N}\theta^T\theta$: The **regularization penalty** on parameter magnitude

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

λ: Controls how strongly we penalize large weights



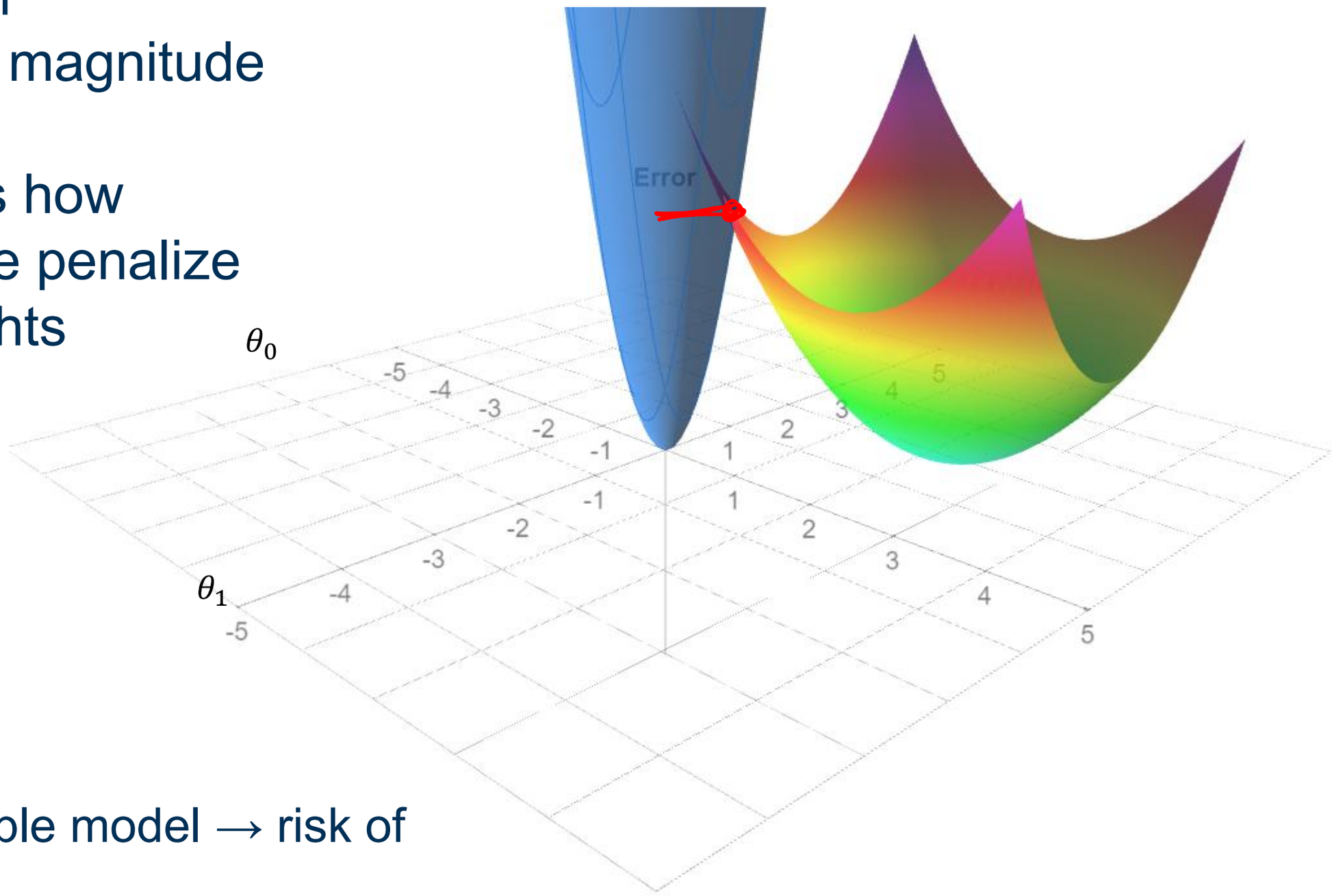
low λ → flexible model → risk of overfitting.

The effect of high Lambda

$\frac{\lambda}{N}\theta^T\theta$: The **regularization penalty** on parameter magnitude

$$\tilde{E}(\theta) = \frac{1}{N}E(\theta) + \frac{\lambda}{N}\theta^T\theta$$

λ: Controls how strongly we penalize large weights



high λ → stable model → risk of underfitting.

Regularized Learning

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

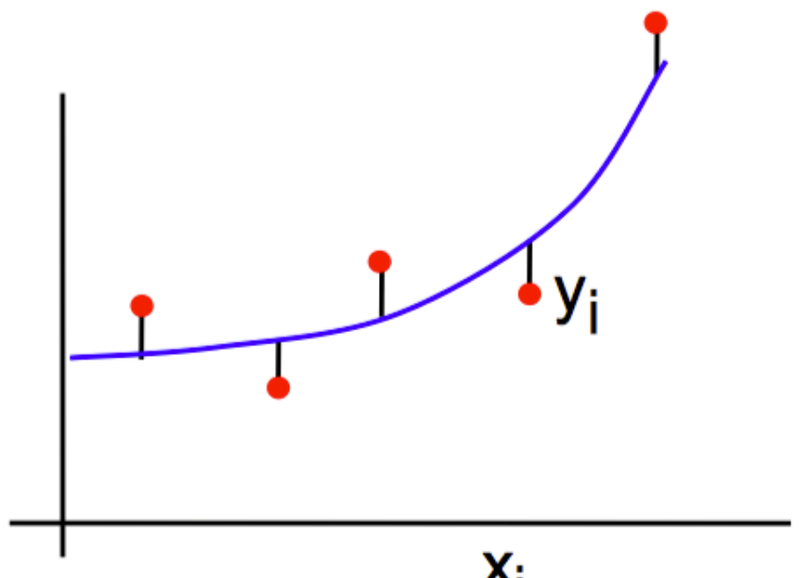
Outline

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

Ridge Regression

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

L_2 norm squared in our penalty term



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

Georgia Tech

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

Closed Form

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

$$\left(z^Tz + \lambda I\right)\theta = z^Ty$$

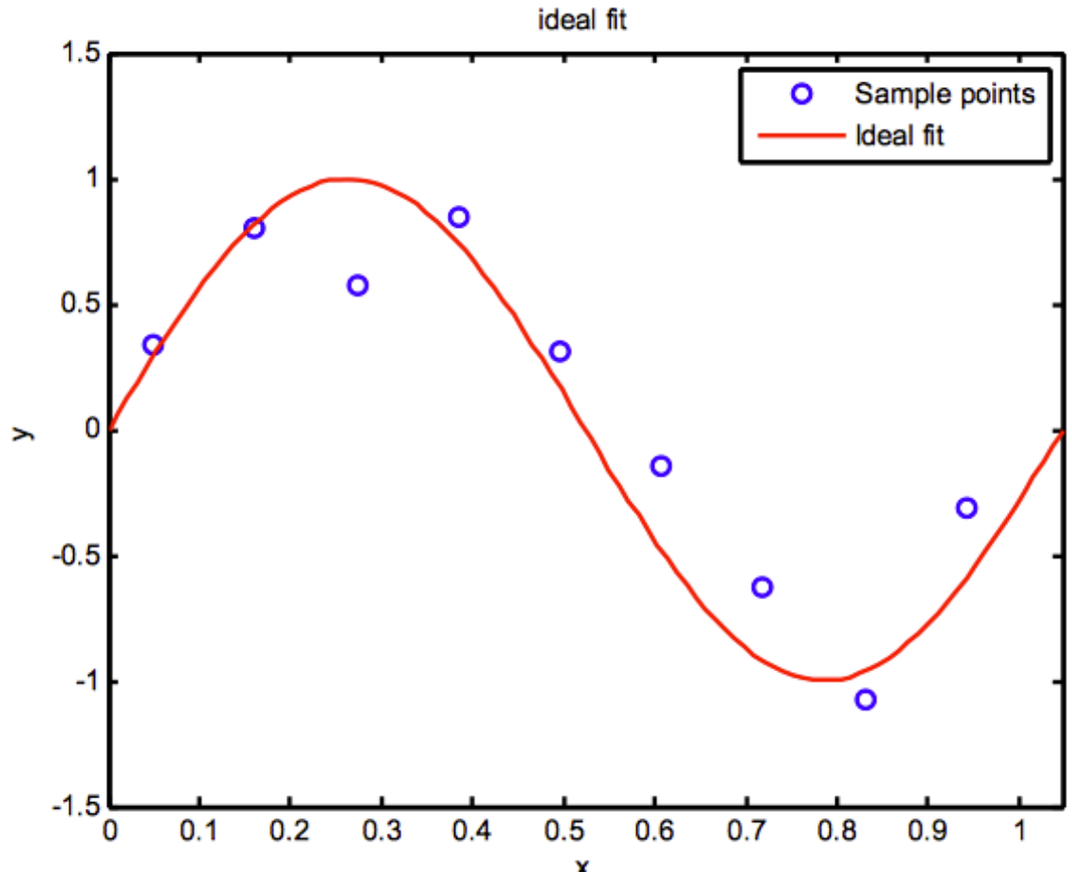
$$\theta = \left(z^Tz + \lambda I\right)^{-1}z^Ty$$

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



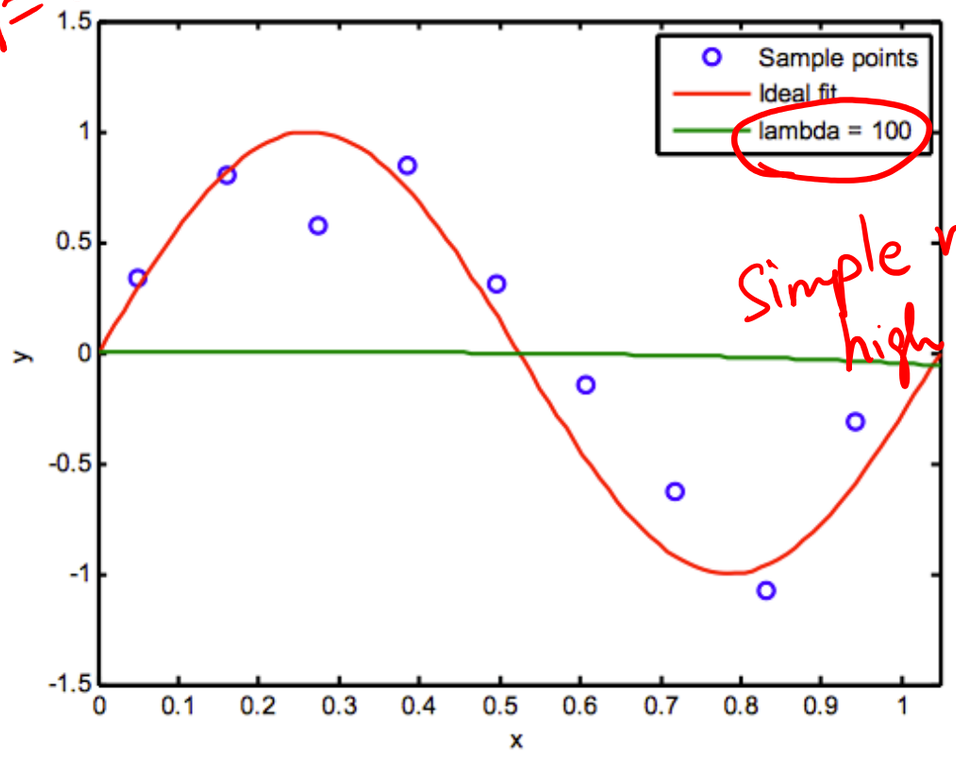
$$f(x,\theta) = z\theta \qquad z: x \to z$$

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

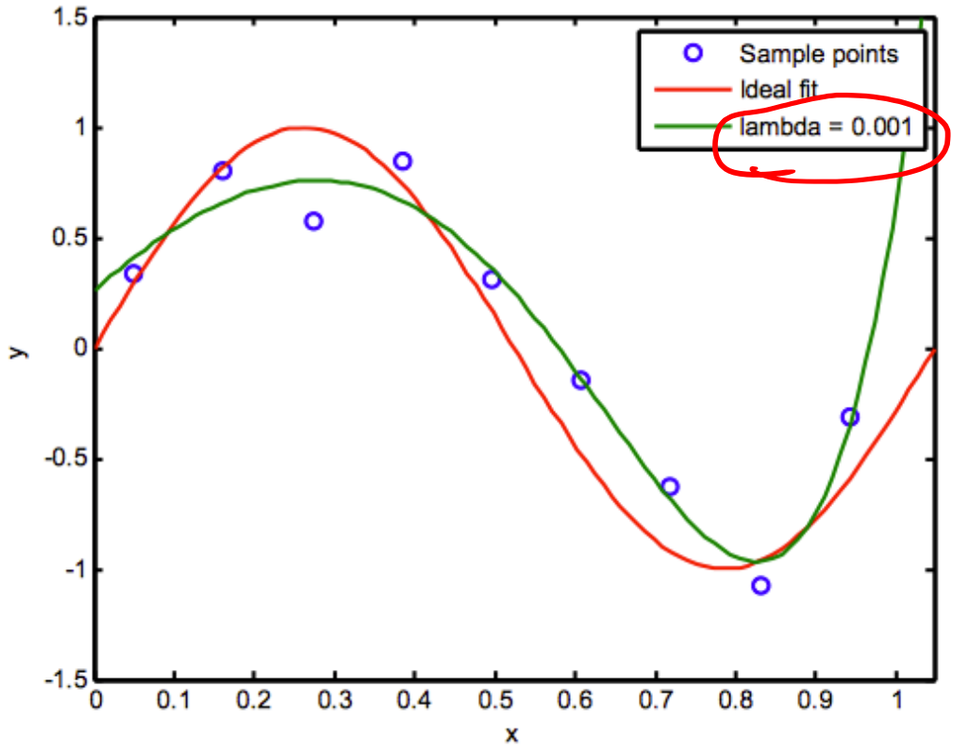
Georgia Tech

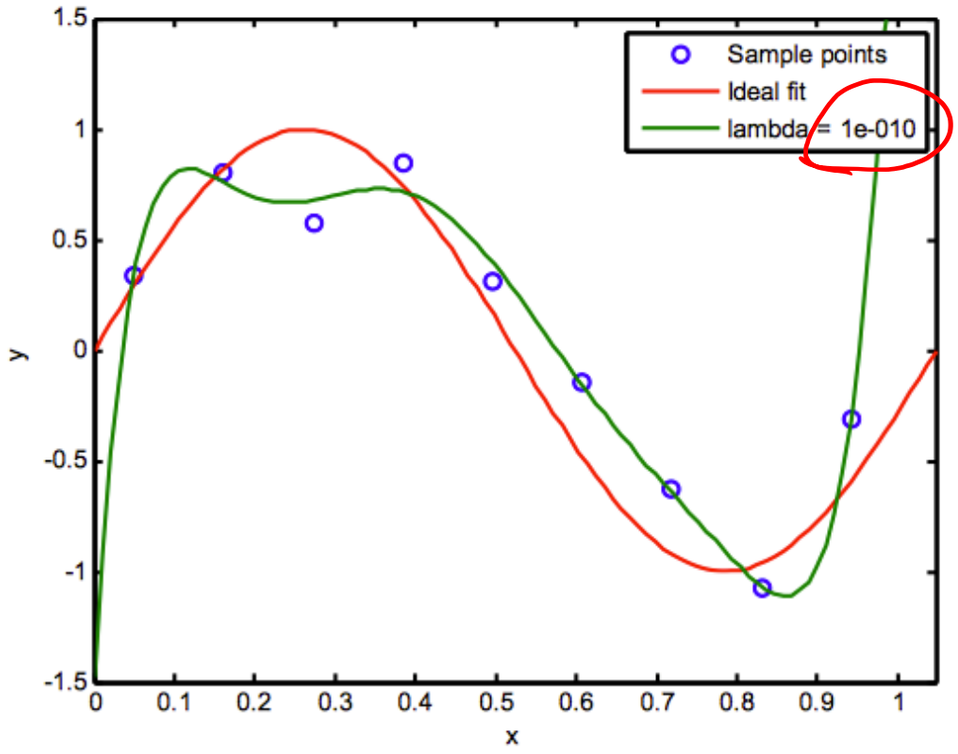
N = 9 samples, D = 7

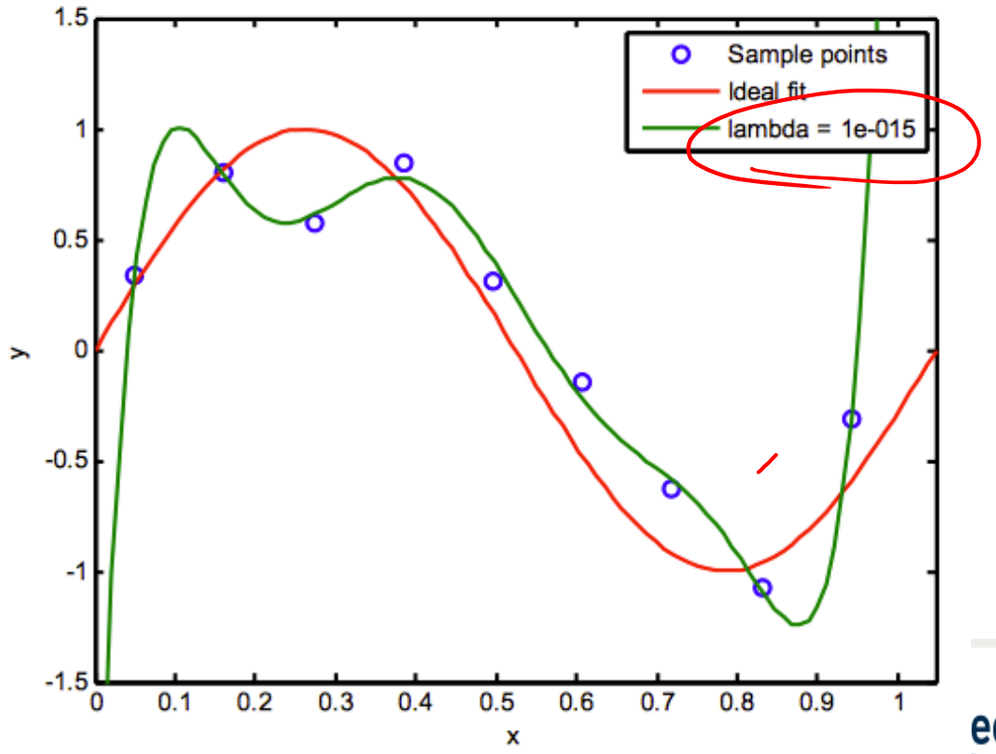
λ = 100



Simple model
high bias

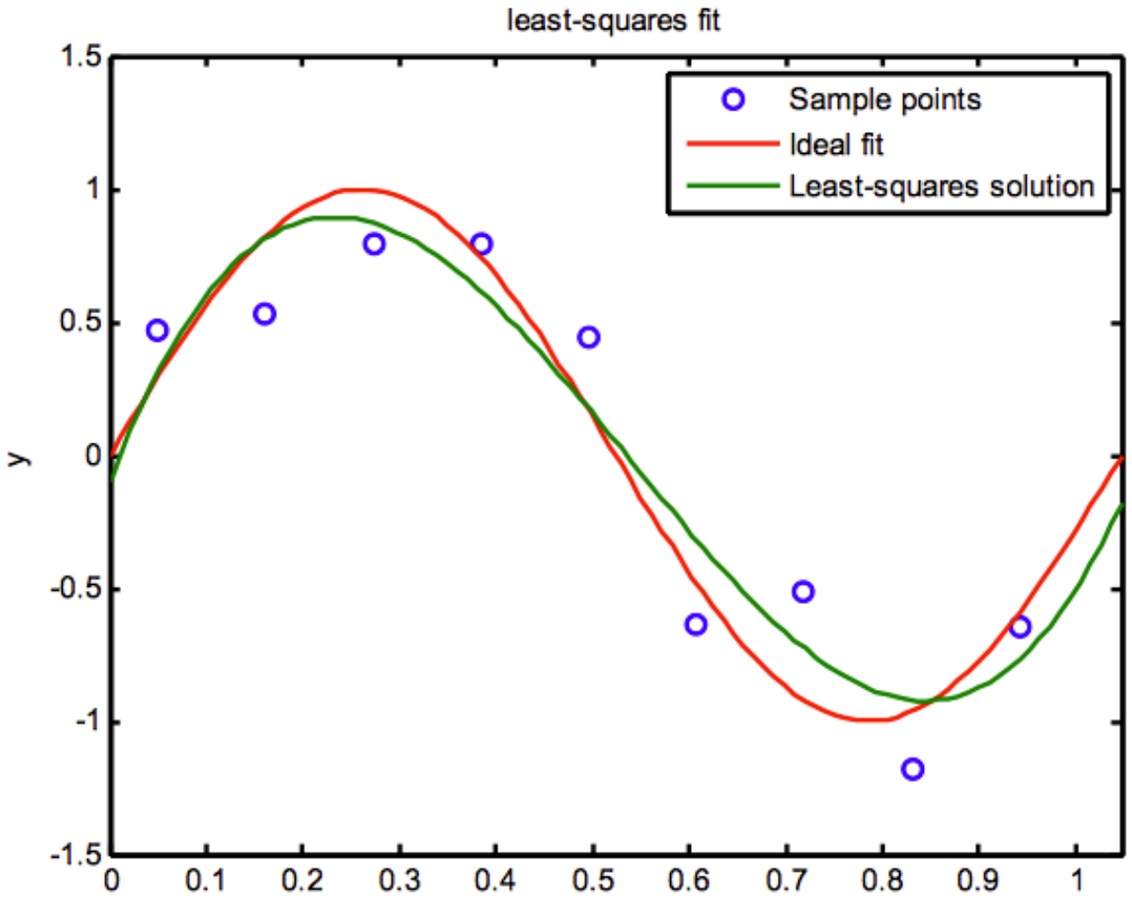




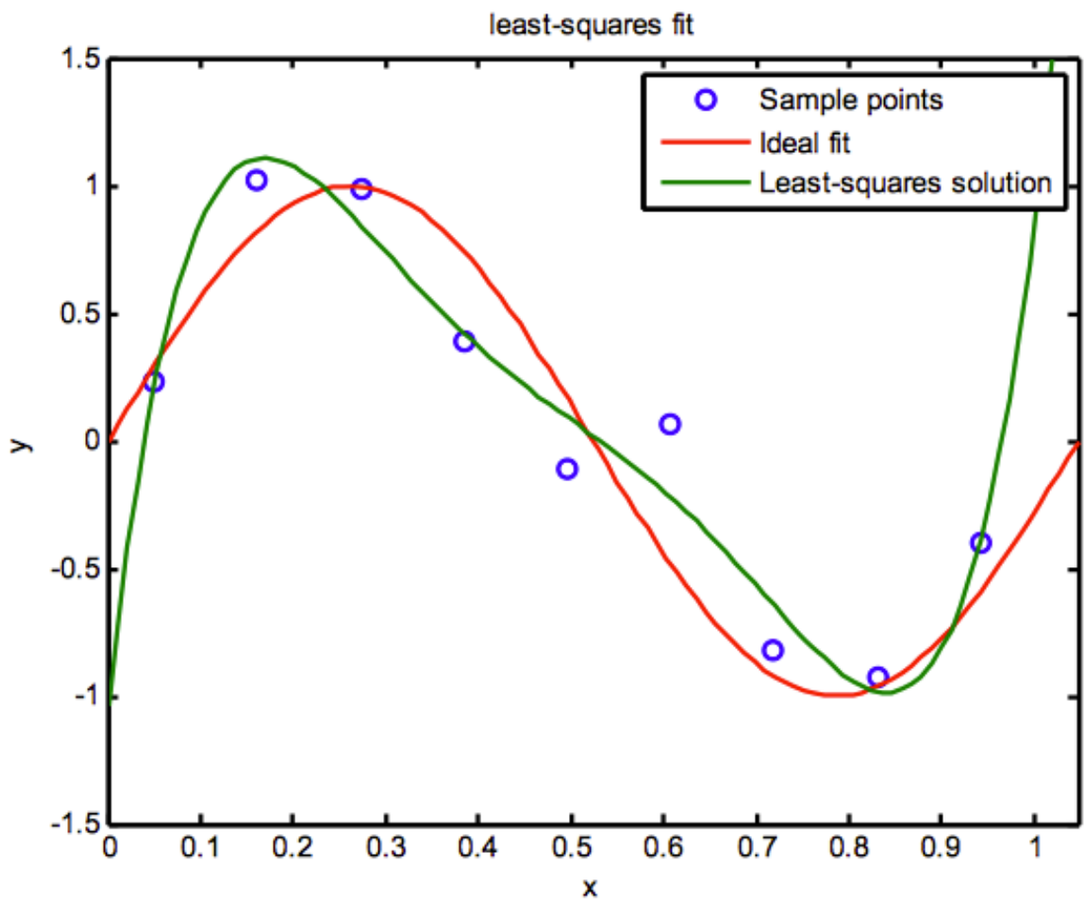


Whole point: Use high value of D so that we benefit from different shapes coming from different higher degrees WITHOUT OVERFITTING.

D = 3



D = 5



Ridge Regression In Action: Predicting App Engagement

Goal:
Predict user engagement using features like:

- time on app, clicks, scroll depth
- engineered features: time², time³ (like x, x^2, x^3)

Without Regularization (Overfitting)

- Model learns large weights:
- $\theta_1 = 500,\ \theta_2 = -480,\ \theta_3 = 300$
- Small input change → huge prediction change (user scrolls slightly more → prediction jumps wildly)
- Model is **unstable / wiggly**
- Fits noise in training data

With Ridge Regression

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

- Weights shrink:
  $\theta_1 = 80,\ \theta_2 = 60,\ \theta_3 = 40$
- Predictions become **smooth and stable**
- Less sensitive to noise
- Better generalization

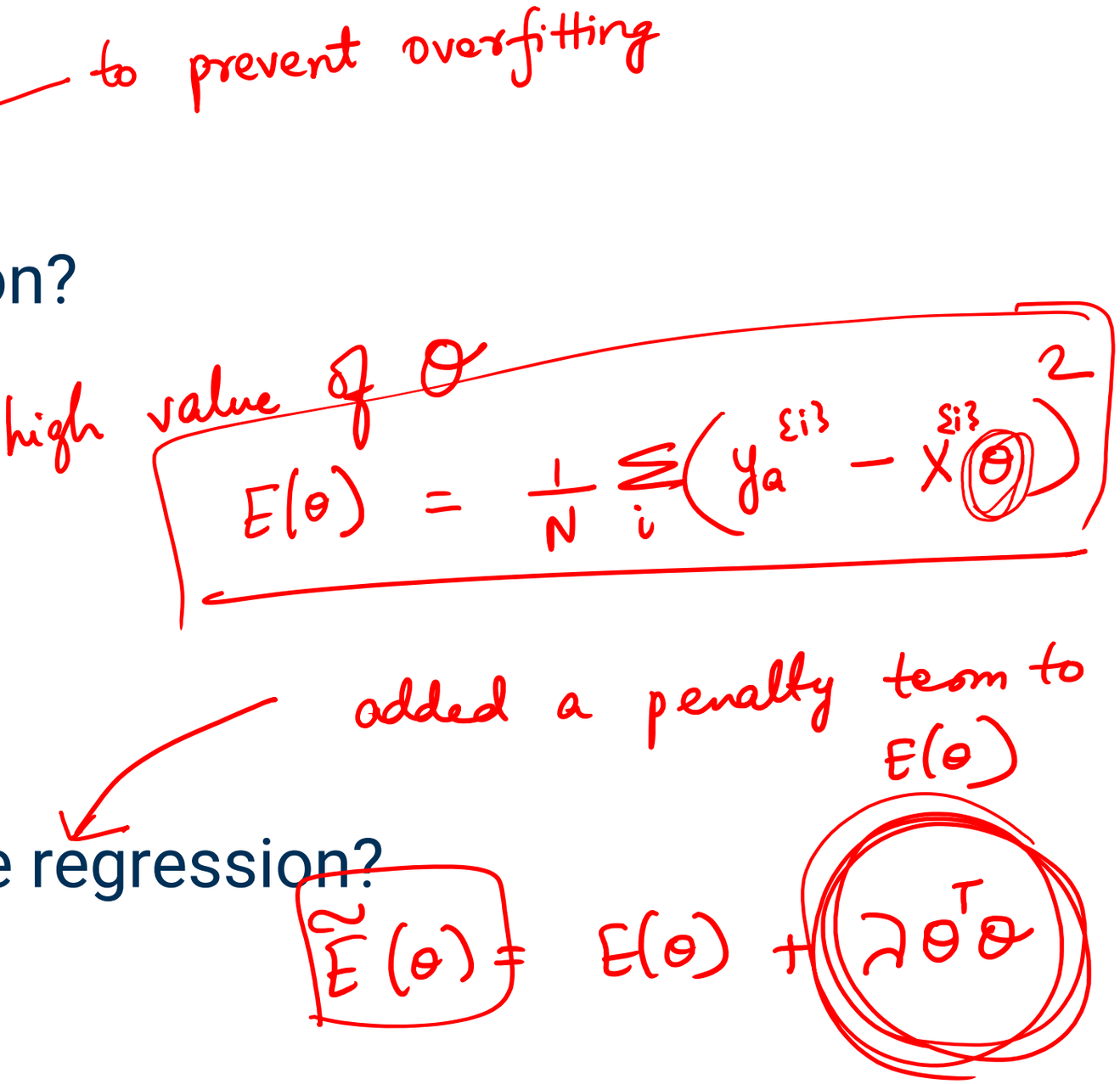
Key Takeaway
Ridge regression controls **large θ values**, preventing extreme reactions to inputs and reducing overfitting.

Georgia Tech

WHEN SPRING BREAK VACATION IS SO CLOSE YOU CAN SMELL IT

Recap

- Why do we need regularization?
- What is the root cause?
- How do we regularize in ridge regression?
- Why is the penalty term chosen as $\theta^T \theta$?



Recap

Ridge Regression → $\tilde{E}(\theta) = E(\theta) + \lambda \|\theta\|_2^2$

$\frac{\partial \tilde{E}(\theta)}{\partial \theta} = 0$ & $\frac{\partial \tilde{E}(\theta)}{\partial \lambda} = 0$

Implicit Sol.
θ is dep. on λ &
λ is dep. on λ

- New regularized objective function?
- Why was lambda chosen as a hyperparamter to optimize the objective function?
- Small vs high values of lambda

λ ↓↓↓ → θ could be anything → Overfitting is more common → high θ

λ ↑↑↑ → underfitting

λ ↑ to be high.

- Practical usage of ridge regression

Collinear features $(X^TX)^{-1}$ may not exist.

$\theta_{ridge} = (X^TX + \lambda I)^{-1} X^T y$.

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

L2 Regularizer $\quad \lambda\|\theta\|_2^2$ L_2 norm for ridge

Now let's look at another regularization choice.

The Lasso Regularization (L1 norm) and sparsity

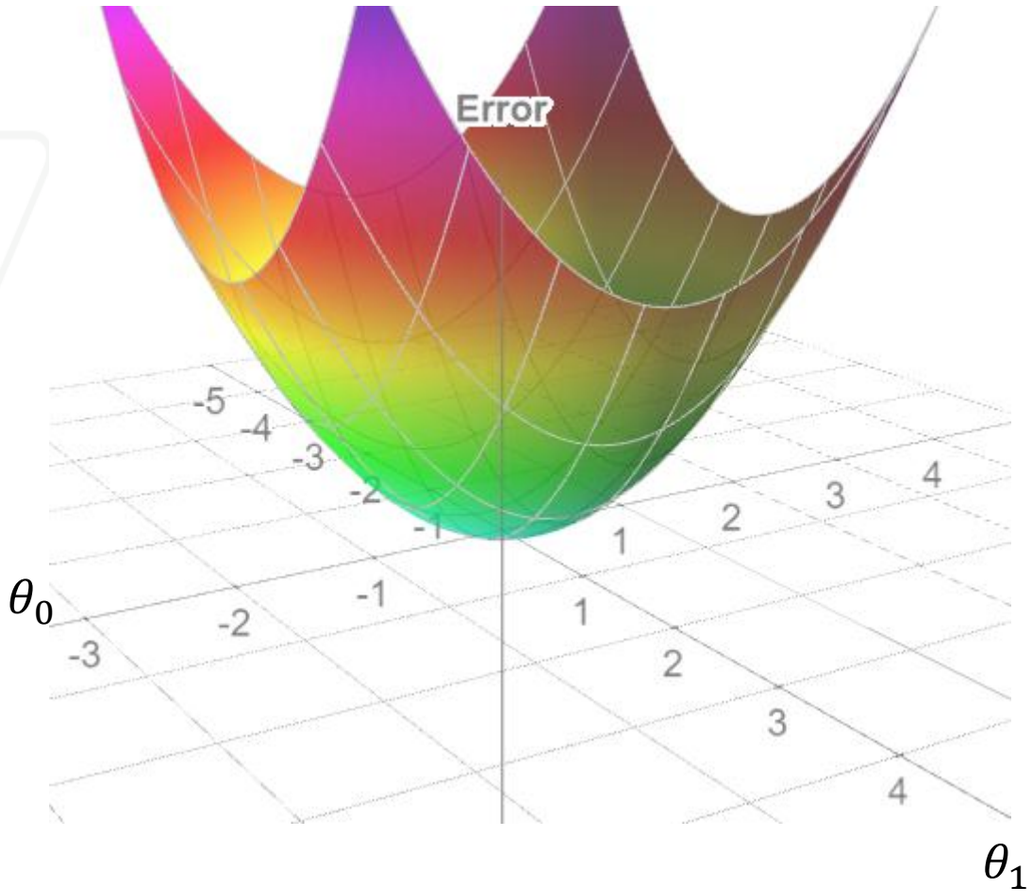
Lasso = **L**east Absolute **S**hrinkage and **S**election **O**perator

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

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

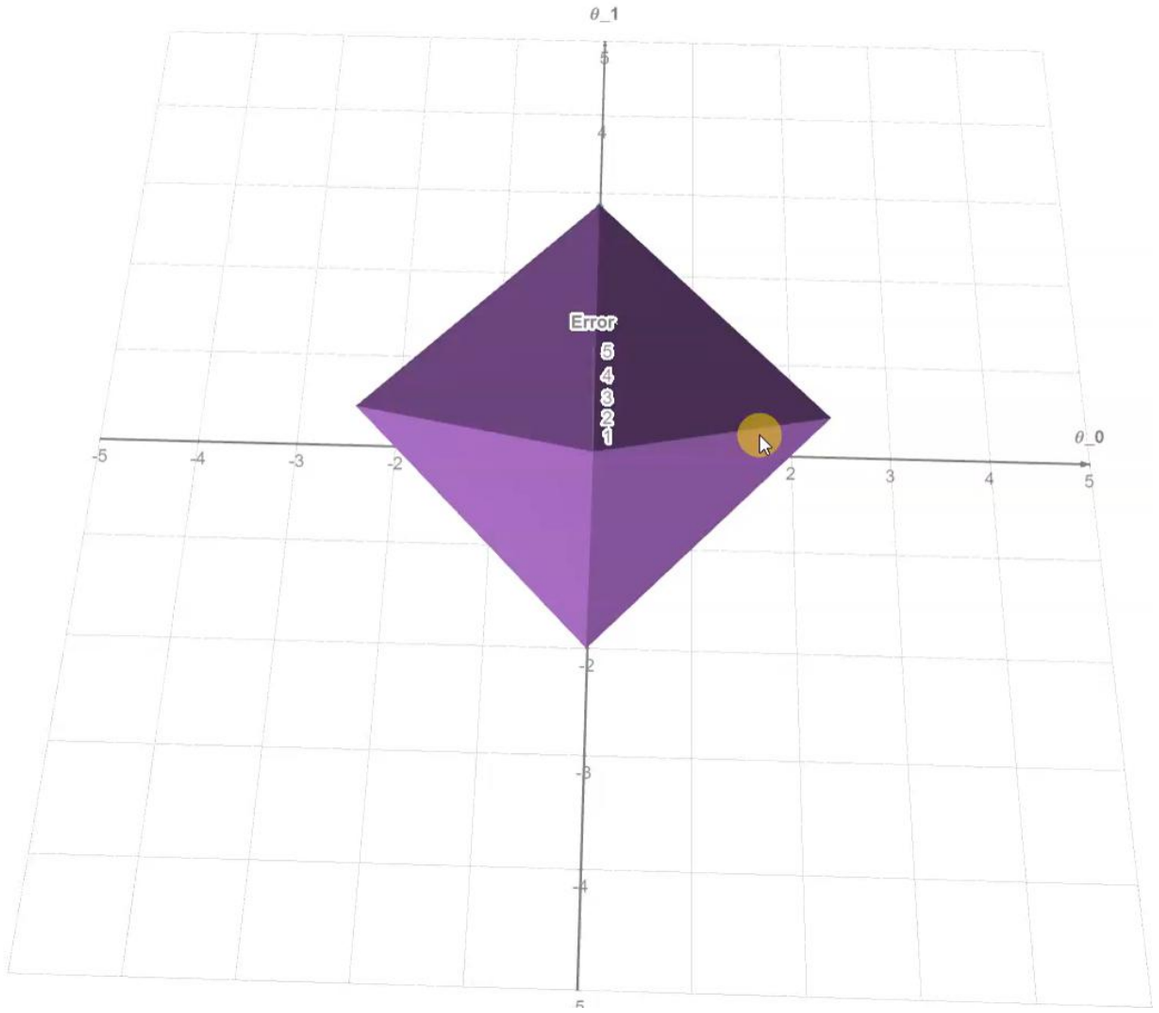
Ridge Regularizer

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



Lasso Regularizer

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



[Animation]

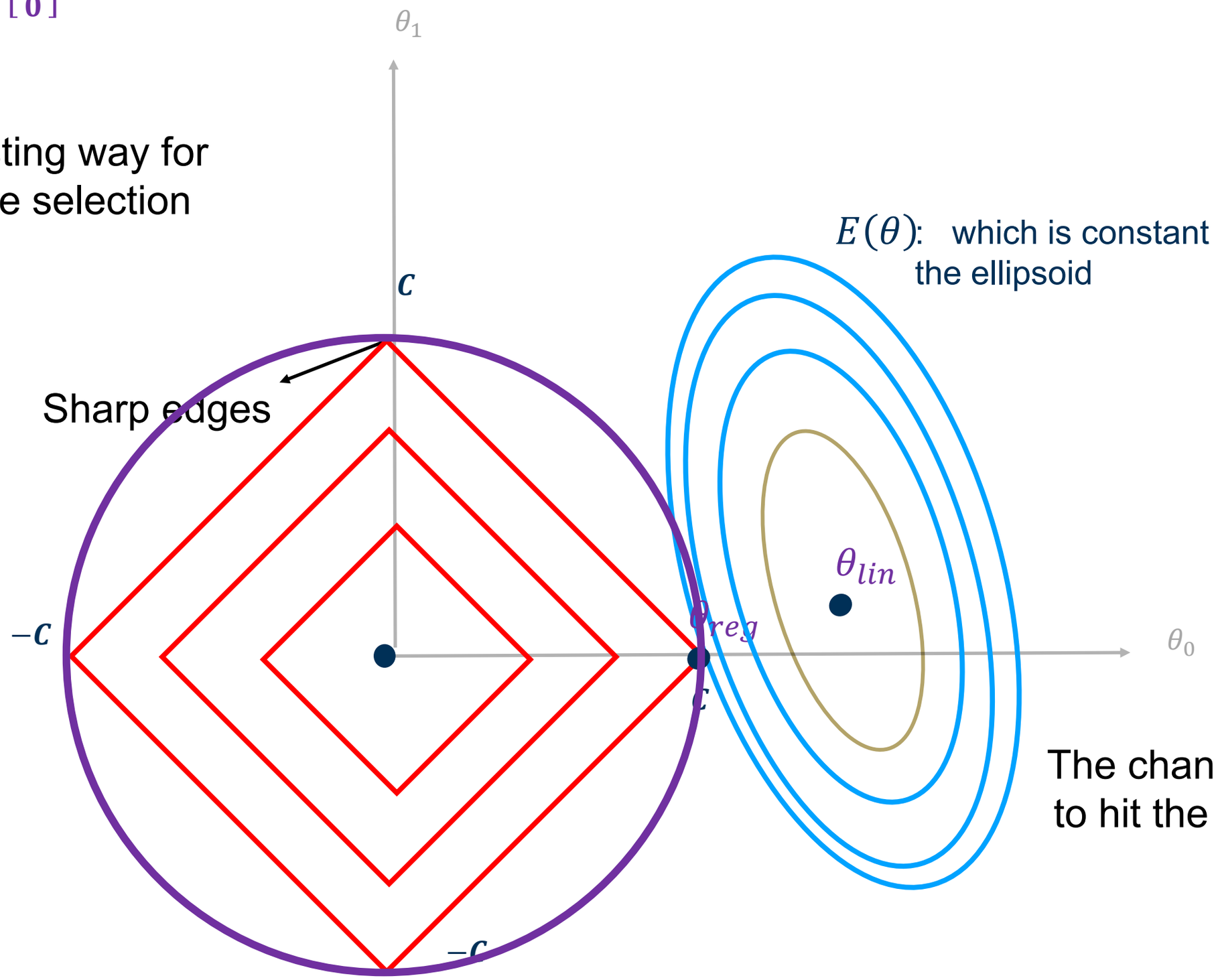
Georgia Tech

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

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

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

Interesting way for feature selection



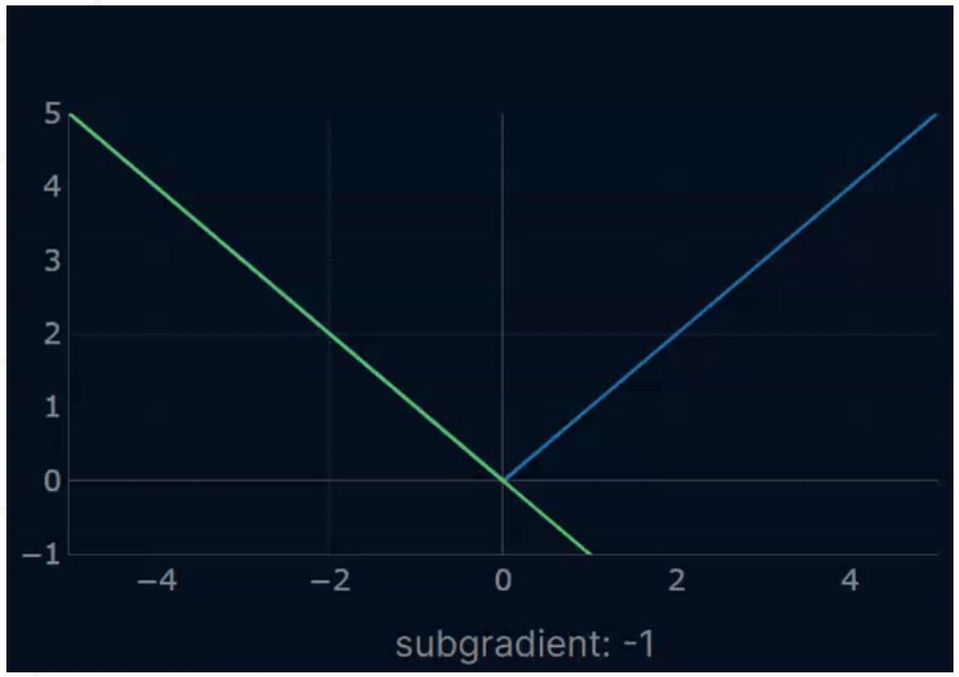
$E(\theta)$: which is constant on the surface of the ellipsoid

The chance is very high to hit the sharp corners first

Graph

Sub-gradient Descend in Lasso

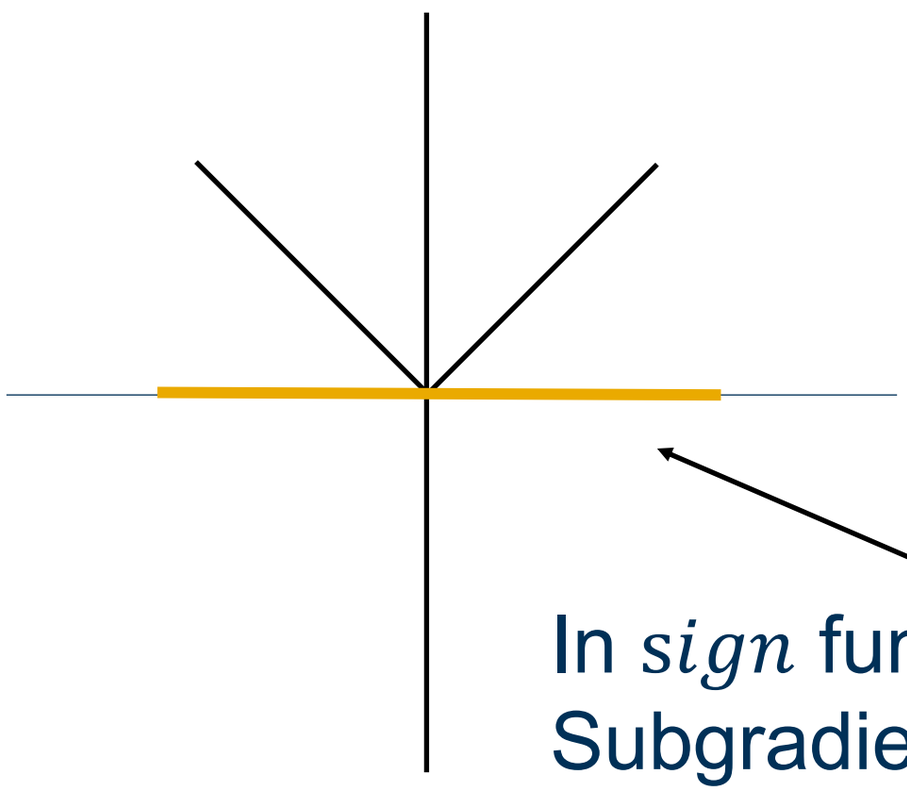
Gradient Descent.



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

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

$sign(\theta) = -1 \quad \theta < 0$
$= 1 \quad \theta > 0$
$= \quad \theta = 0$
$= [-1, 1]$



Using Sub-gradient

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

In $sign$ function, we use this sub-gradient line
Subgradient at 0 = any value in [-1, 1]

Georgia Tech

Ridge versus Lasso

Ridge

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

Both mean squared error and L2 regularizer are differentiable.

We can get a closed form solution

shrinks coefficients (but rarely to 0)

Lasso

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

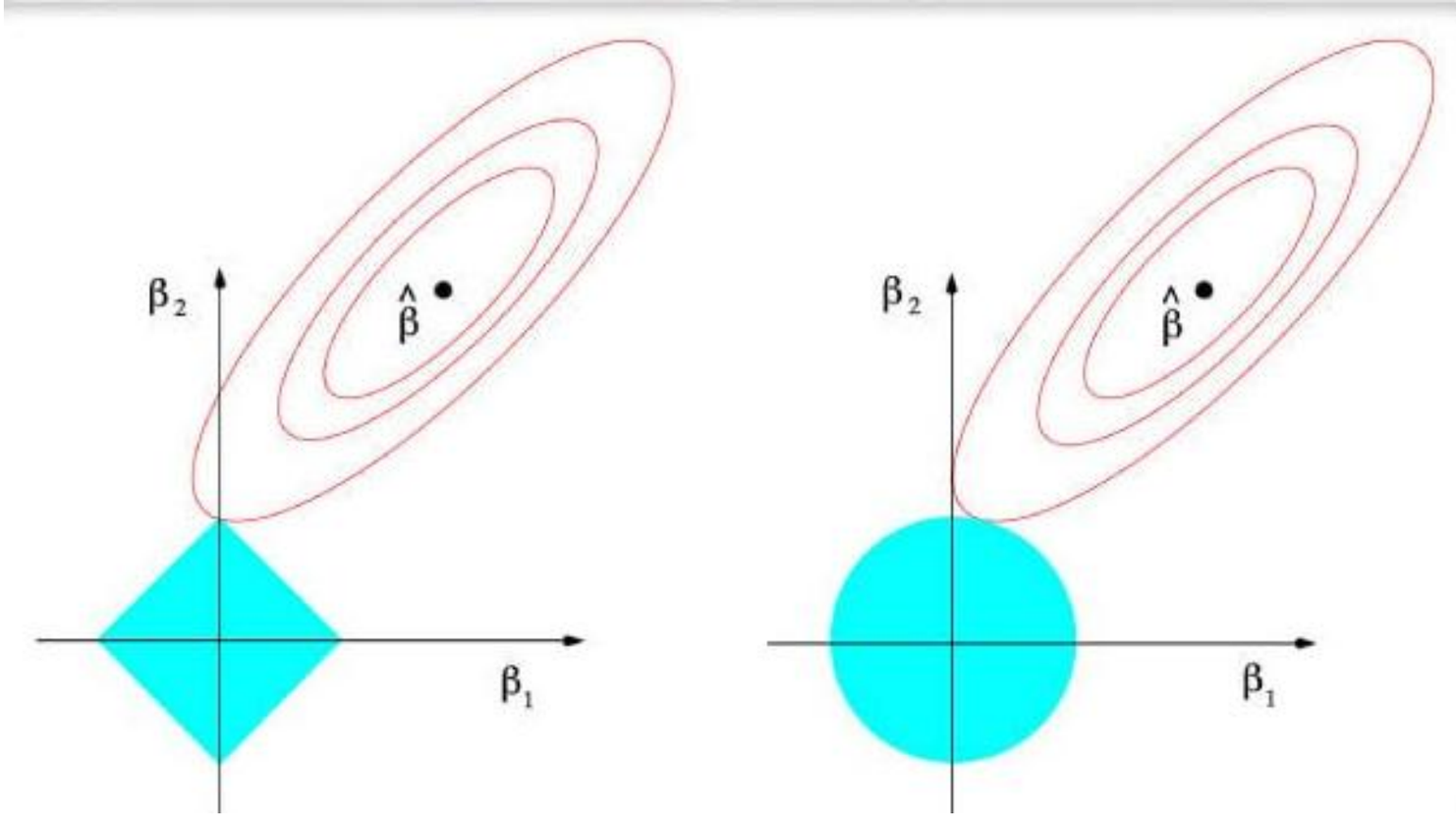
L1 regularizer is NOT differentiable.

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

can **force coefficients to exactly 0** → does **feature selection**

Georgia Tech

Ridge versus Lasso

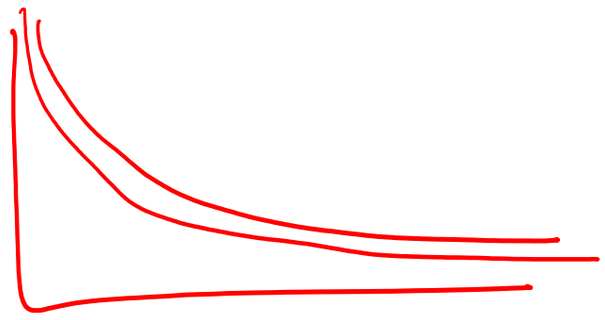


Scenario	Use Lasso?
High-dimensional data (p >> n)	✅ Yes
Need feature selection	✅ Yes
Interpretability important	✅ Yes
Many correlated features	❌ Prefer Ridge
Small, dense signal across features	❌ Prefer Ridge

Lasso assumes **“only a few features truly matter”**, while Ridge assumes **“many features matter a little.**

Georgia Tech

Regularization	Bias	Variance	Total Error	Stability
❌ Without	Low	**High**	High	Overfits
✅ With	Slightly higher	**Much lower**	Lower	Generalizes better




THE BEST WAY TO
EXPLAIN OVERFITTING

Recap

- Other forms of regularization covered in previous classes?

Ridge Regression

Lasso Regression → feature selection

PCA → Regularization

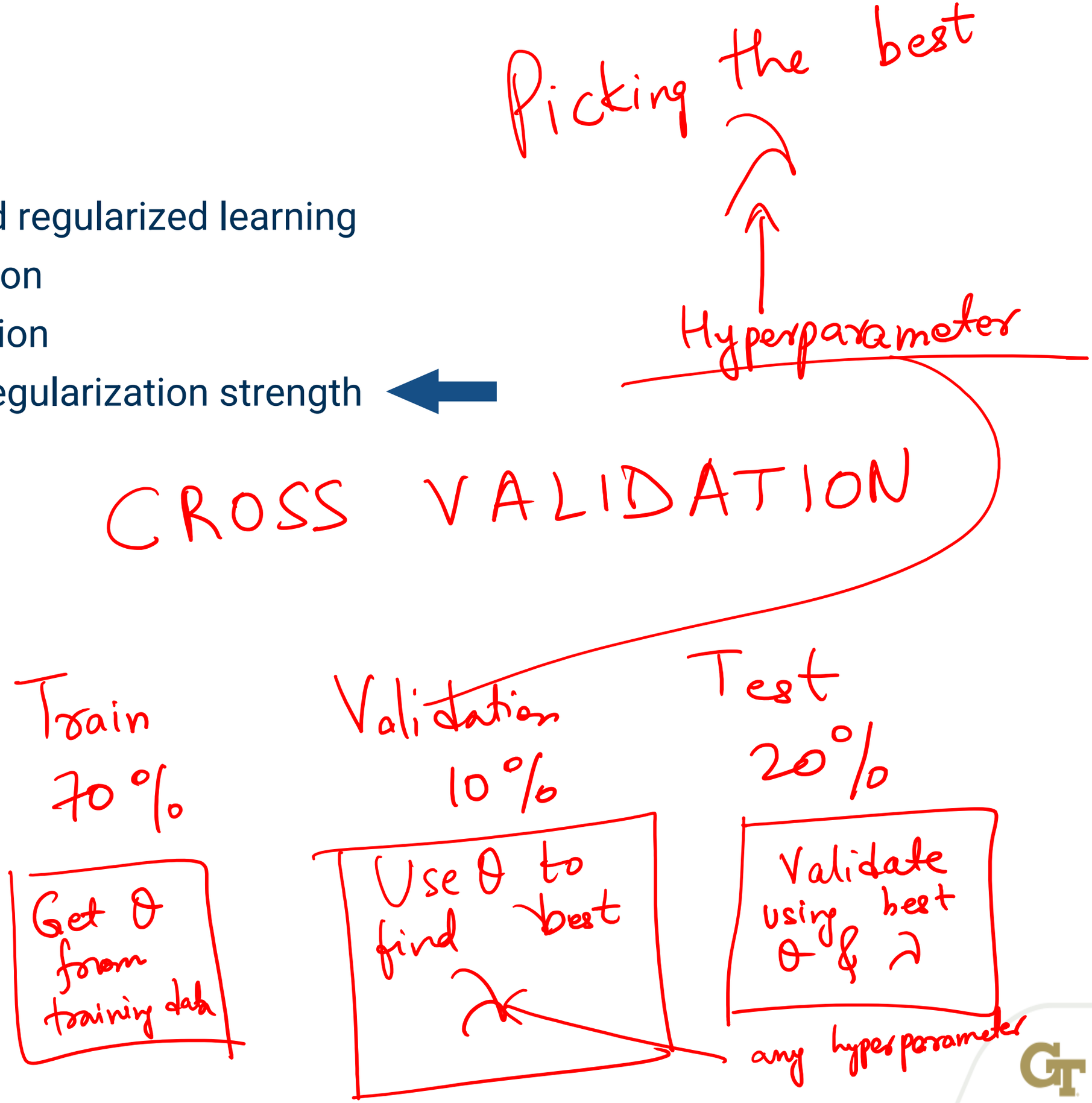
Backward Feature Selection

Forward Feature Selection

LDA → RF Feature Selector

Outline

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



is bad

Leave-One-Out Cross Validation

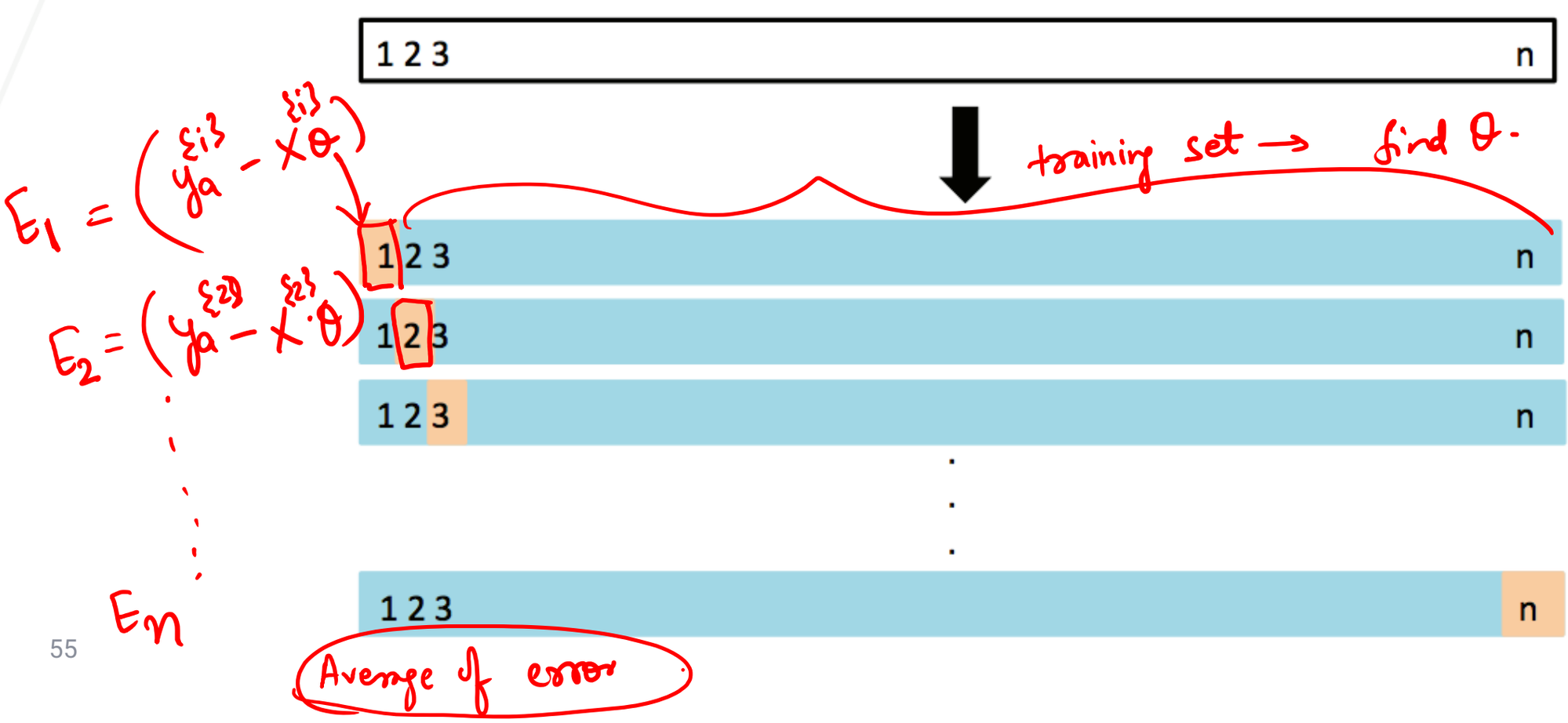
$\lambda = 0.0001$

$\lambda = 0.001$

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

- train the model on every point except i,
- compute the test error on the held out point.

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



Georgia Tech

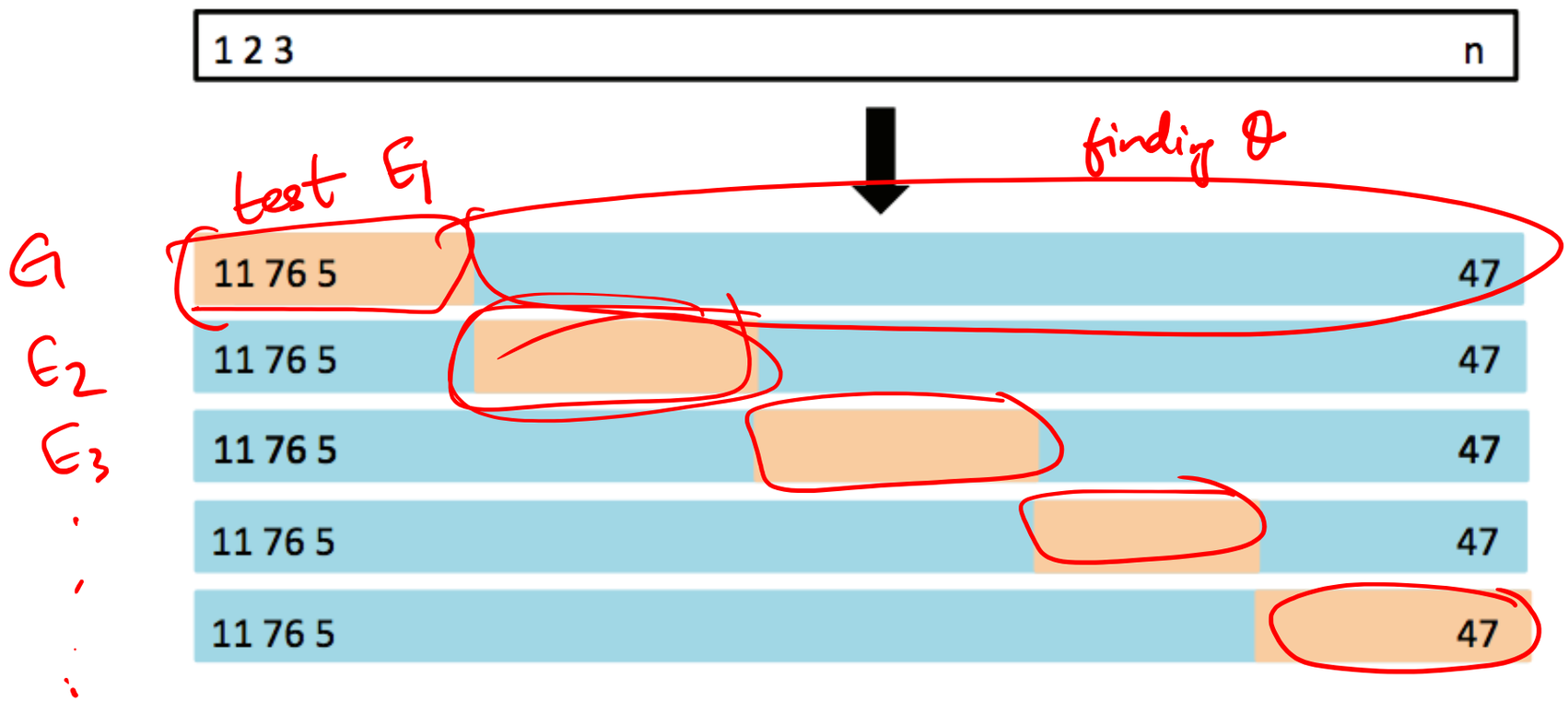
K-Fold Cross Validation

Split the data into k subsets or *folds*.

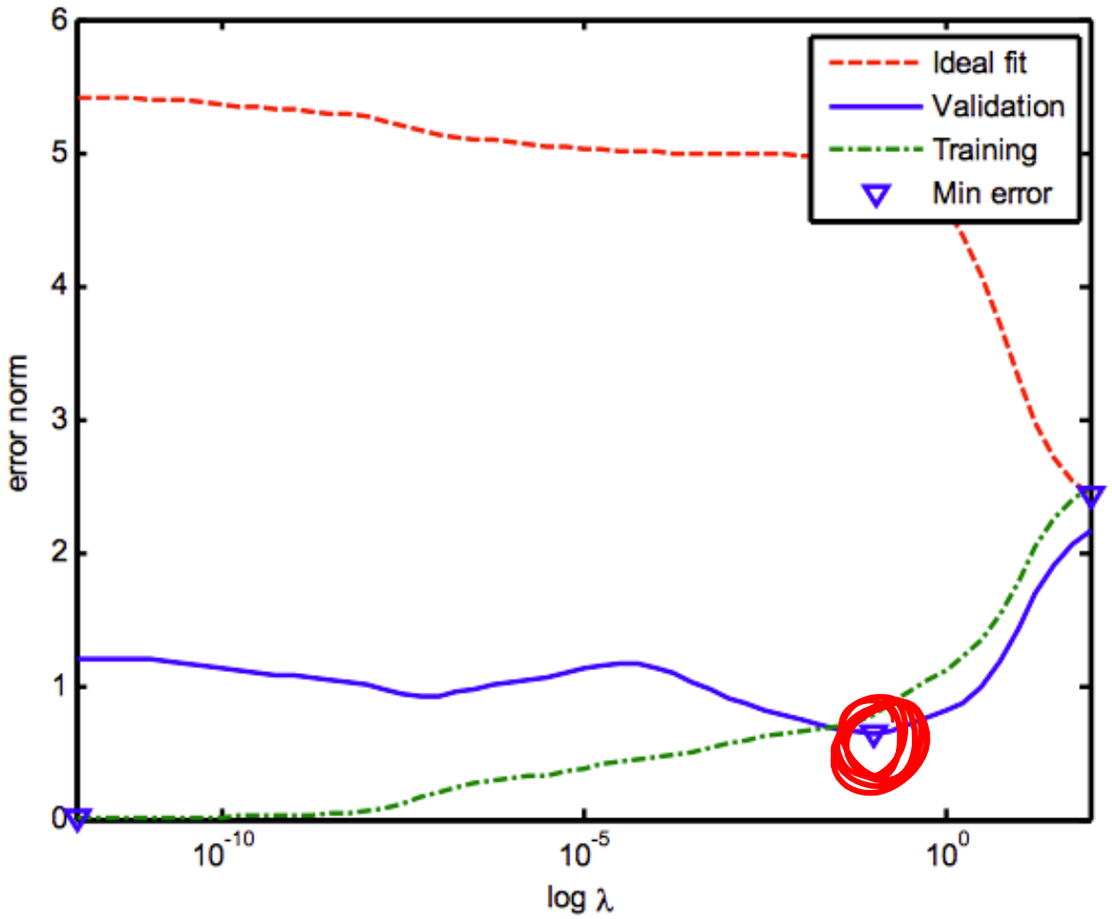
For every $i = 1, \ldots,$ k: $= 5$

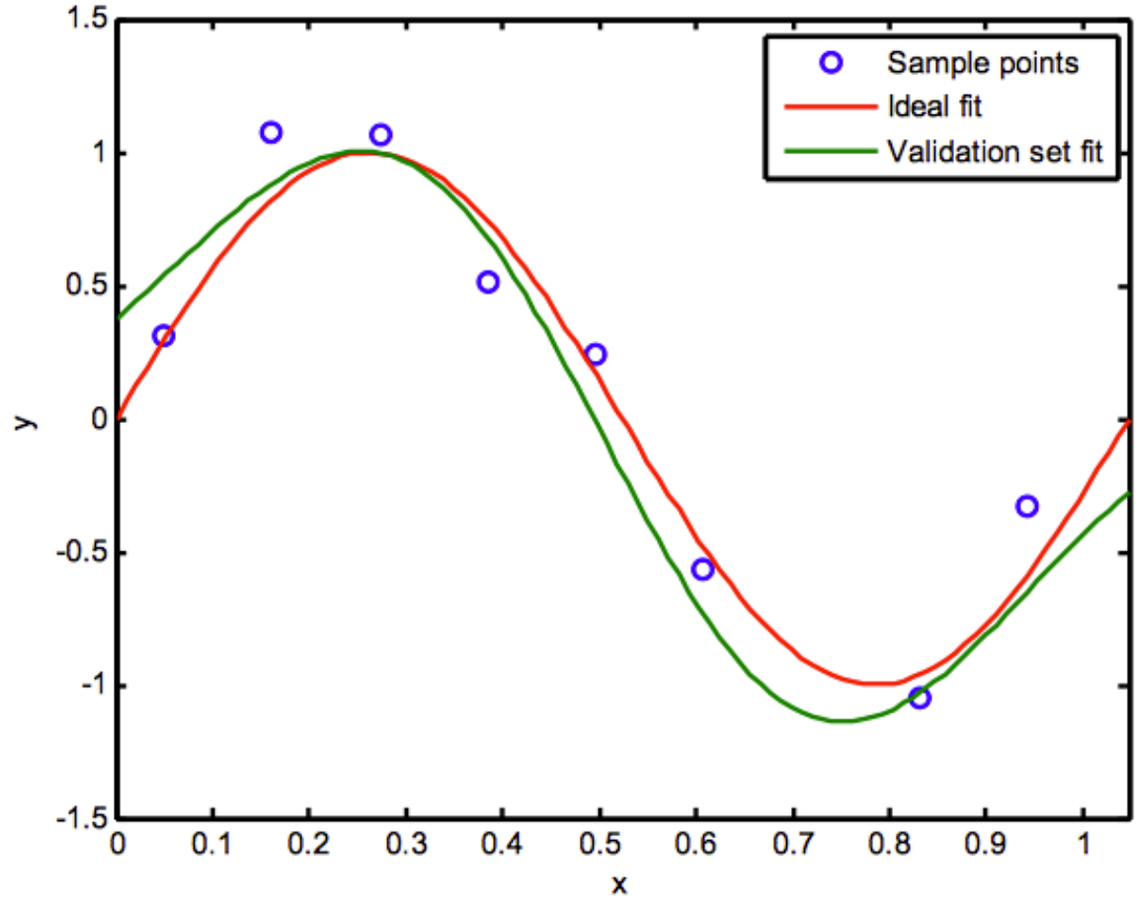
- train the model on every fold except the ith fold,
- compute the test error on the ith fold.

Average the test errors.



Choosing λ Using Validation Dataset





Pick up the lambda with the lowest mean value of mse 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

Quick Knowledge Check

1. Which of the following is the *main reason* for overfitting in regression? A. Too few features. B. Too small a training dataset. C. Large model weights. D. High bias
2. When λ (lambda) is **very small**, what happens to the model? A. High bias, low variance. B. Low bias, high variance. C. Both bias and variance decrease. D. Both bias and variance increase
3. Which regularization technique results in **sparser** models? A. Lasso (L1). B. Ridge (L2). C. Both equally. D. Neither
4. In **k-fold cross-validation**, what does each fold provide? A. A new model parameter. B. A separate test set. C. A different λ value. D. A validation error estimate
5. Fill in the blank for ridge regression:

$$J(\theta;\lambda) = \frac{1}{2n}\|y - X\theta\|^2 + \frac{\lambda}{2}\_\_\_$$

A. $||\theta||_2$ B. $X^T\theta$ C. $||\theta||_2^2$ D. $|\theta|$

Georgia Tech