Machine Learning

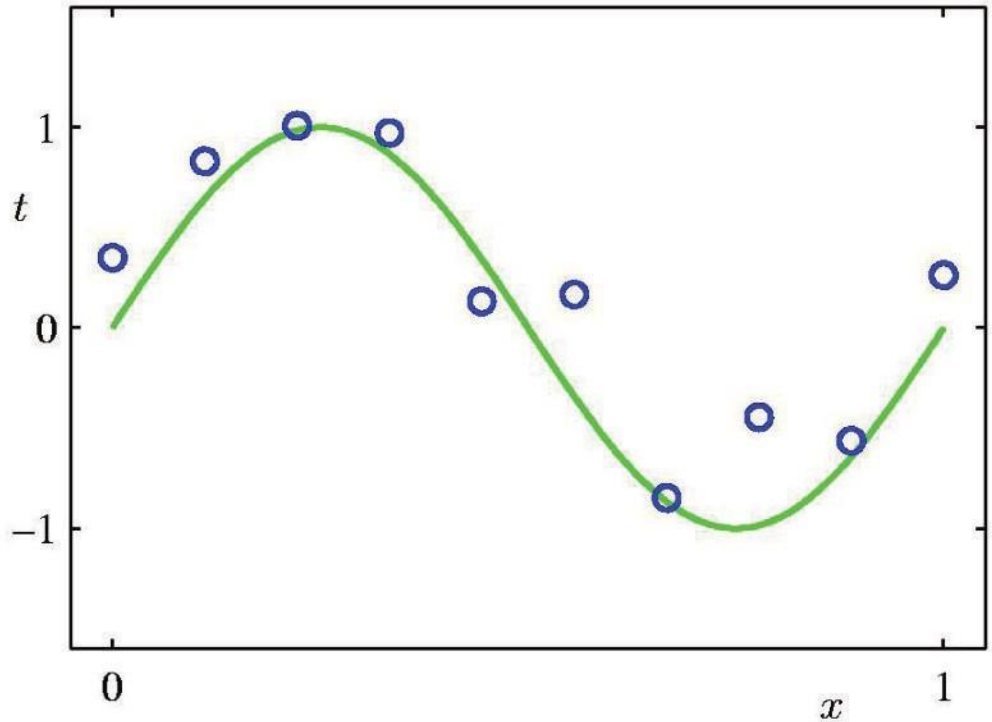
Regularization

Indian Institute of Information Technology
Sri City, Chittoor

Today's Agenda

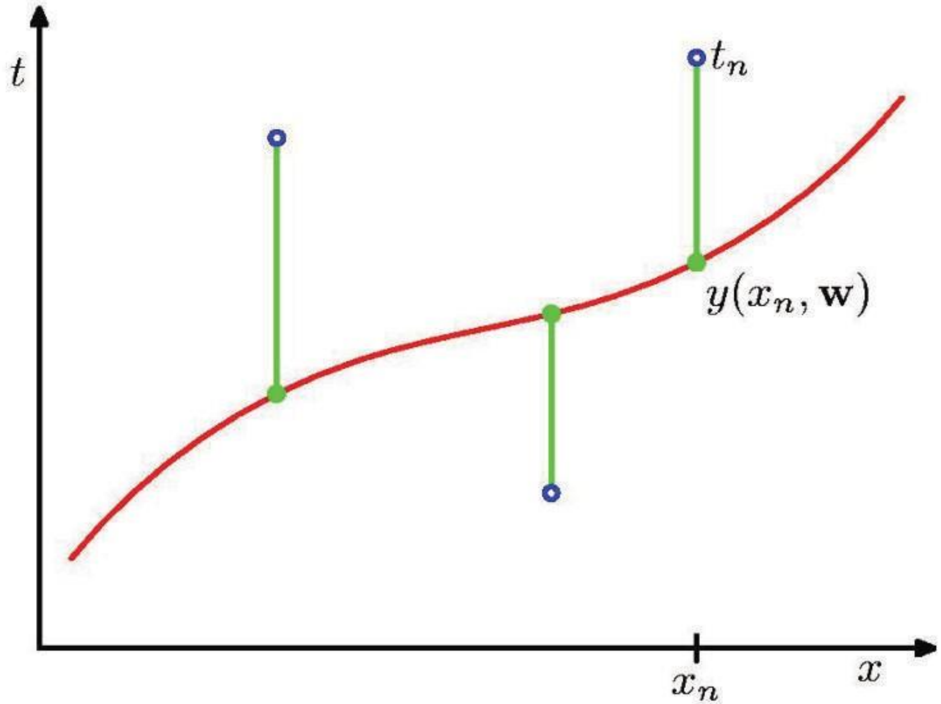
- Regularization

Polynomial Curve Fitting



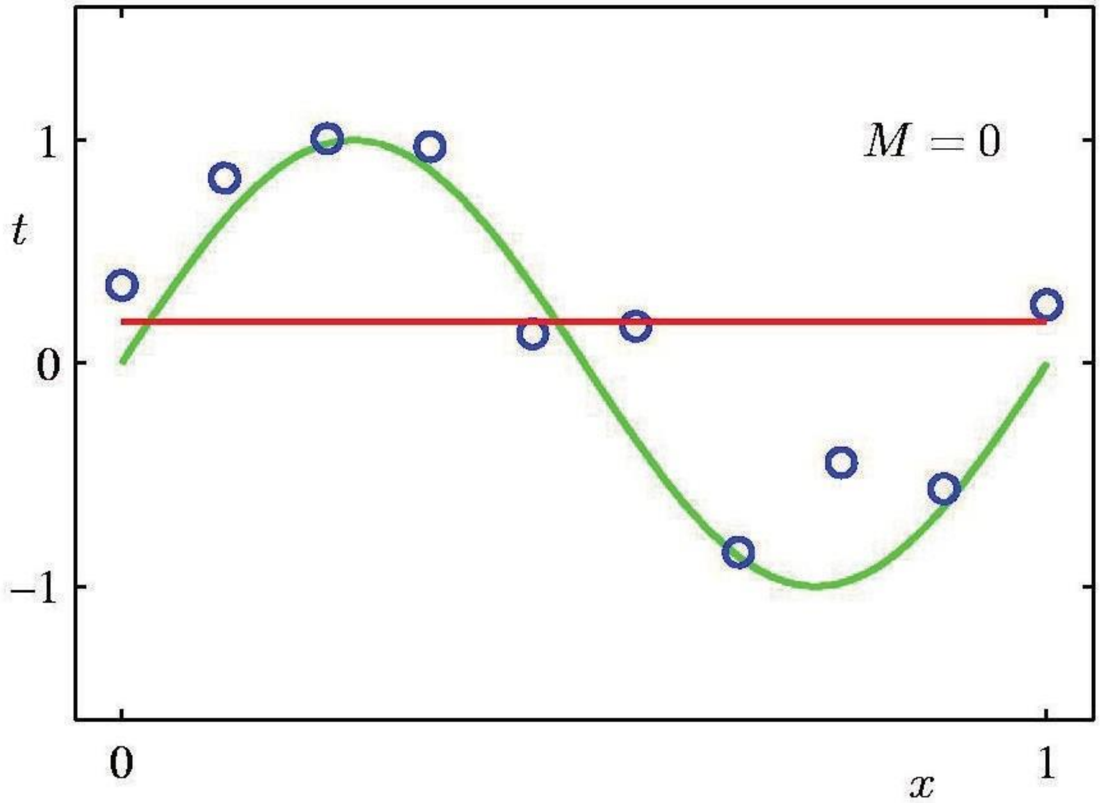
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function

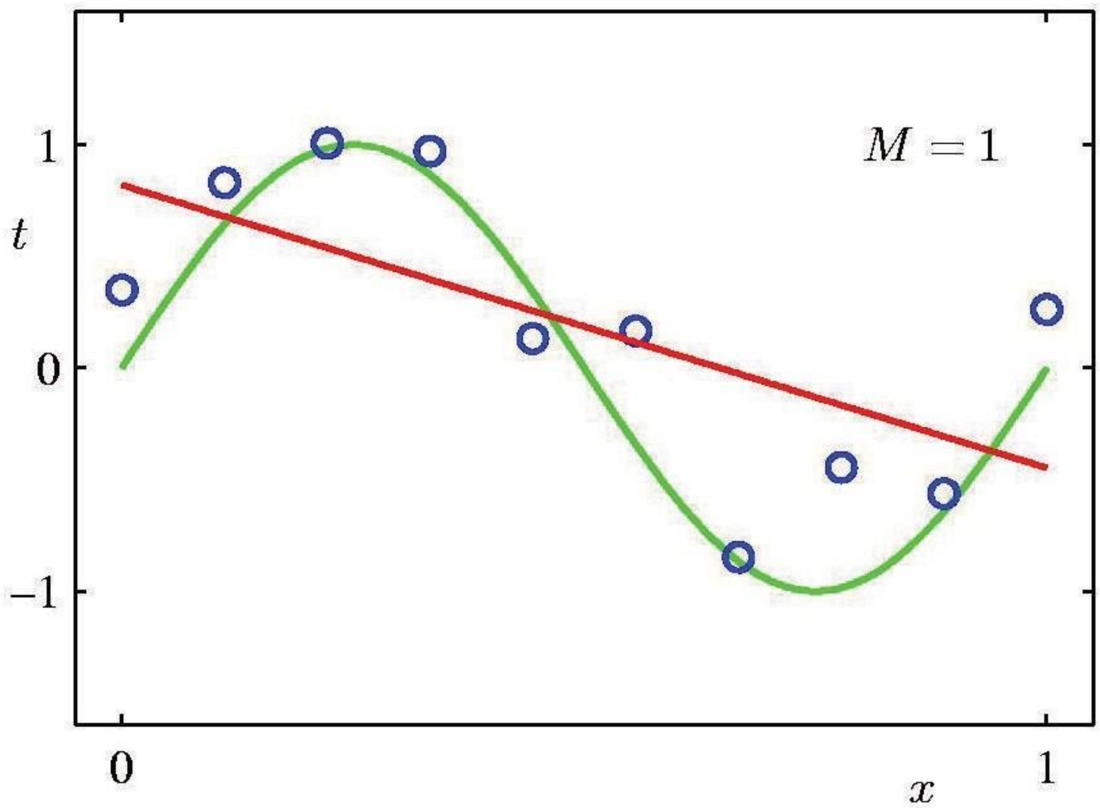


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

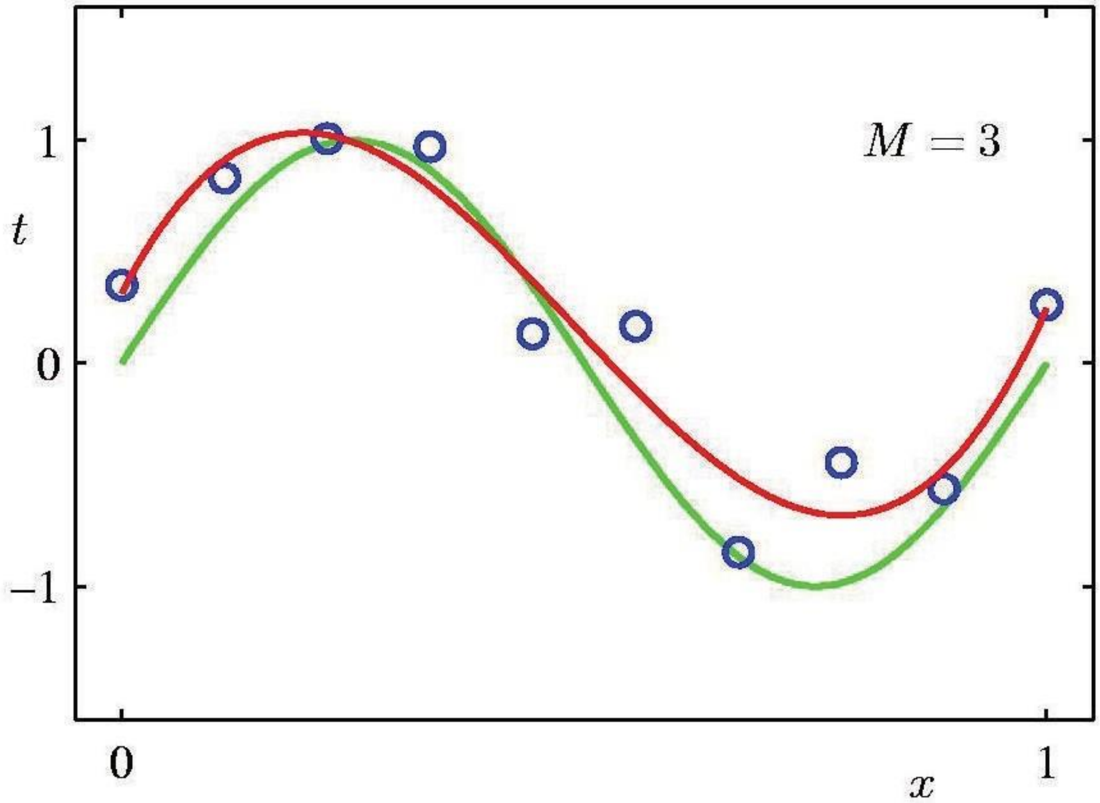
0^{th} Order Polynomial



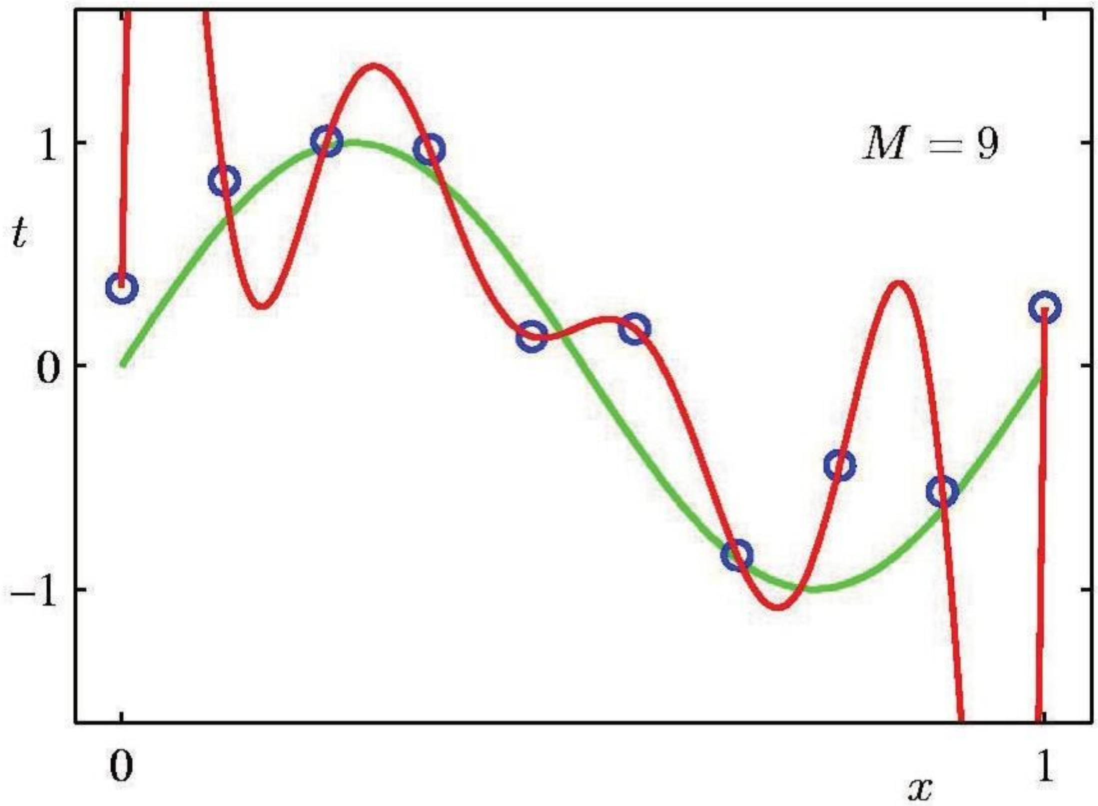
1^{st} Order Polynomial



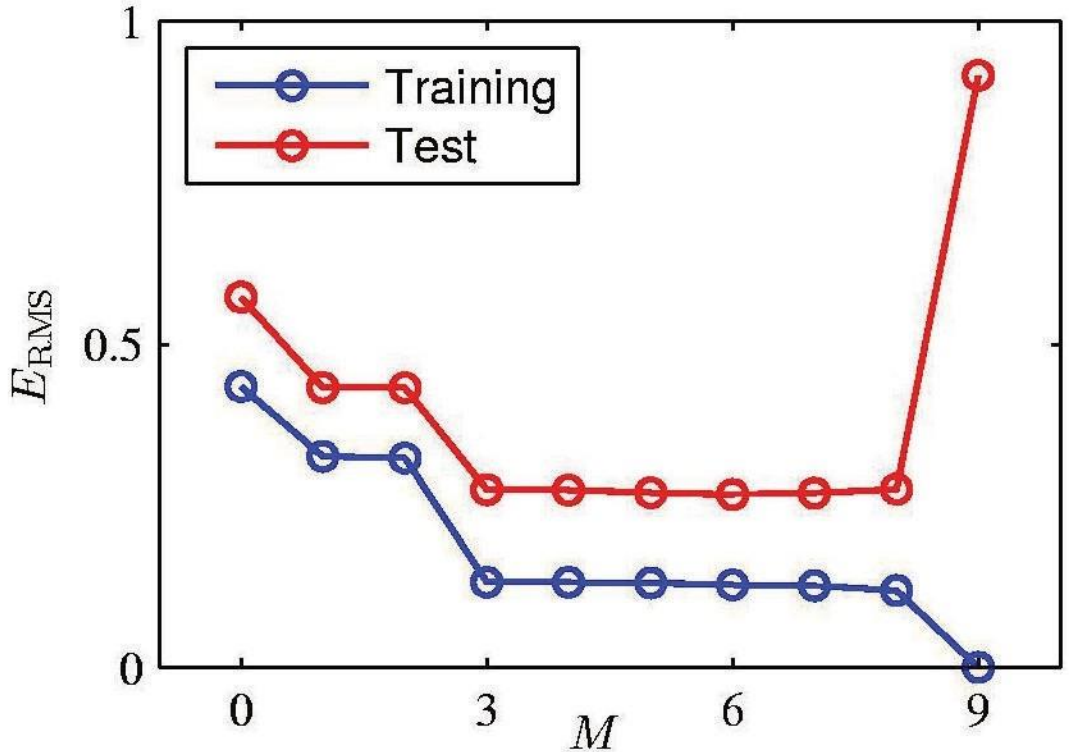
3rd Order Polynomial



9^{th} Order Polynomial



Over-fitting



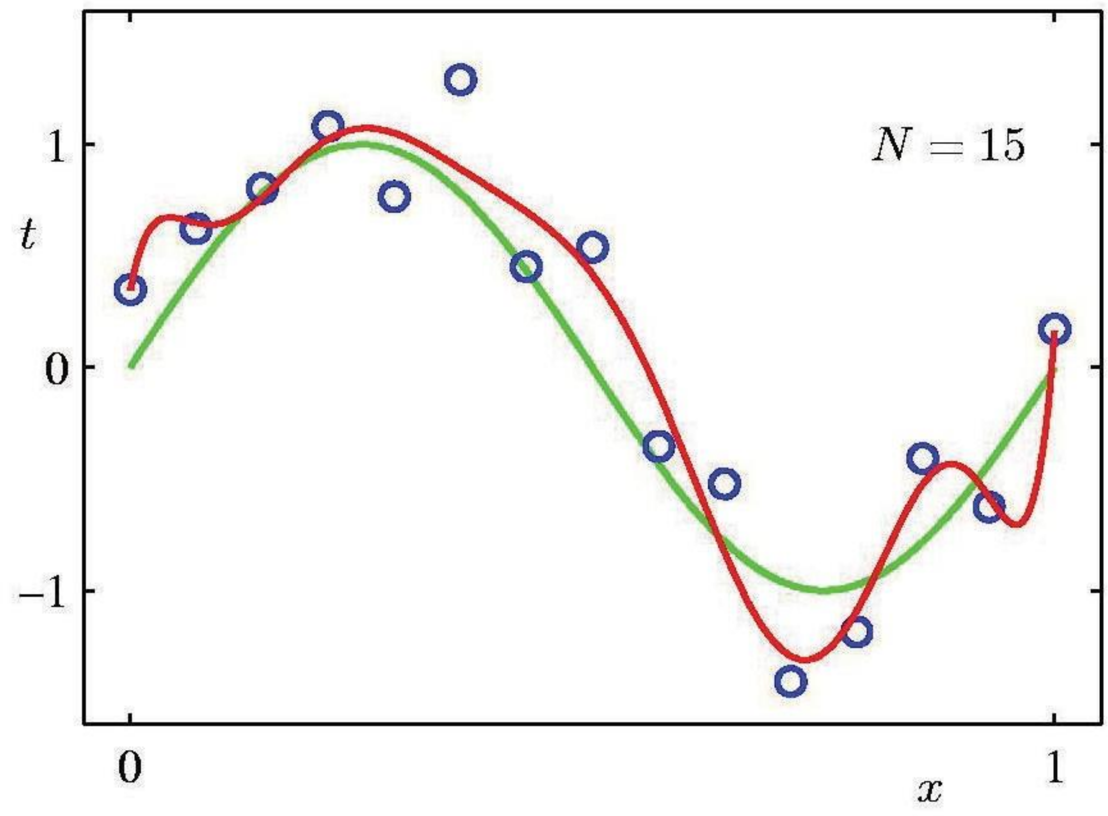
Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

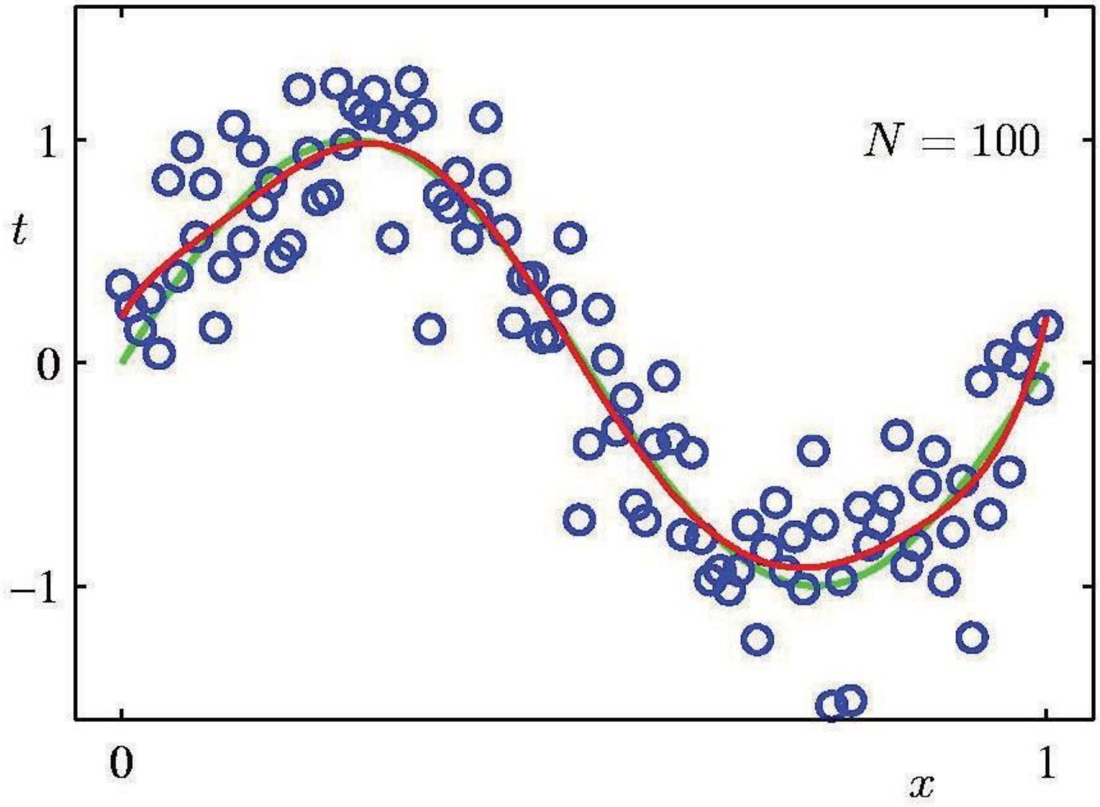
Data Set Size: $N = 15$

9^{th} Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial



Regularization

- Regularization is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.

- In context of machine learning, most regularization strategies are based on regularizing estimators. This is done through reducing variance at the expense of increasing the bias of the estimator.

- In other words, we could prevent overfitting by penalizing complex models, a principle called **regularization**.

Regularization

- In other words, instead of simply aiming to minimize loss (**empirical risk minimization**):

  minimize(Loss(Data|Model))

- we'll now minimize loss+complexity, which is called **structural risk minimization**:

  minimize(Loss(Data|Model)+complexity(model))

- Our training optimization algorithm is now a function of two terms: the **loss term**, which measures how well the model fits the data, and the **regularization term**, which measures model complexity.

https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/l2-regularization

Regularization

- In other words, instead of simply aiming to minimize loss (**empirical risk minimization**):

  minimize(Loss(Data|Model))

- we'll now minimize loss+complexity, which is called **structural risk minimization**:

  minimize(Loss(Data|Model)+complexity(model))

- Our training optimization algorithm is now a function of two terms: the **loss term**, which measures how well the model fits the data, and the **regularization term**, which measures model complexity.

https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/l2-regularization

Regularization

Regularization refers to the act of modifying a learning algorithm to favor “simpler” prediction rules to avoid overfitting.

Most commonly, regularization refers to modifying the loss function to **penalize** certain values of the weights you are learning.

- Specifically, penalize weights that are *large*.

Regularization

How do we define whether weights are *large*?

$$d(\mathbf{w}, \mathbf{0}) = \sqrt{\sum_{i=1}^{k} (w_i)^2} = ||\mathbf{w}||$$

This is called the **L2 norm** of **w**

- A norm is a measure of a vector's length
- Also called the Euclidean norm

Regularization

New goal for minimization:

$$\underbrace{L(\mathbf{w})}_{\text{This is whatever loss function we are using}} + \lambda \, ||\mathbf{w}||^2$$

Regularization

New goal for minimization:

$$L(\mathbf{w}) + \underbrace{\lambda\,||\mathbf{w}||^2}$$

By minimizing this, we prefer solutions where **w** is closer to **0**.

Regularization

New goal for minimization:

$$L(\mathbf{w}) + \underbrace{\lambda \|\mathbf{w}\|^2}_{\text{By minimizing this, we prefer solutions where } \mathbf{w} \text{ is closer to } \mathbf{0}.}$$

Why squared? It eliminates the square root; easier to work with mathematically.

By minimizing this, we prefer solutions where **w** is closer to **0**.

Regularization

New goal for minimization:

$$L(\mathbf{w}) + \underbrace{\lambda \, ||\mathbf{w}||^2}_{}$$

Why squared? It eliminates the square root; easier to work with mathematically.

By minimizing this, we prefer solutions where **w** is closer to **0**.

λ is a **hyperparameter** that adjusts the tradeoff between having low training loss and having low weights.

Regularization

More generally:

$$L(\mathbf{w}) + \lambda \underbrace{R(\mathbf{w})}$$

This is called the **regularization term** or **regularizer** or **penalty**

- The squared L2 norm is one kind of penalty, but there are others

λ is called the regularization **strength**

L2 Regularization

When the regularizer is the squared L2 norm $||\mathbf{w}||^2$, this is called L2 regularization.

- This is the most common type of regularization
- When used with linear regression, this is called *Ridge regression*
- Logistic regression implementations usually use L2 regularization by default
  - L2 regularization can be added to other algorithms like perceptron (or any gradient descent algorithm)

L2 Regularization

The function $R(\mathbf{w}) = ||\mathbf{w}||^2$ is convex, so if it is added to a convex loss function, the combined function will still be convex.

L1 Regularization

Another common regularizer is the L1 norm:

$$||\mathbf{w}||_1 = \sum_{j=1}^{k} |w_j|$$

- When used with linear regression, this is called *Lasso*
- Often results in many weights being exactly 0 (while L2 just makes them small but nonzero)

L2+L1 Regularization

L2 and L1 regularization can be combined:

$R(\mathbf{w}) = \lambda_2 ||\mathbf{w}||^2 + \lambda_1 ||\mathbf{w}||_1$

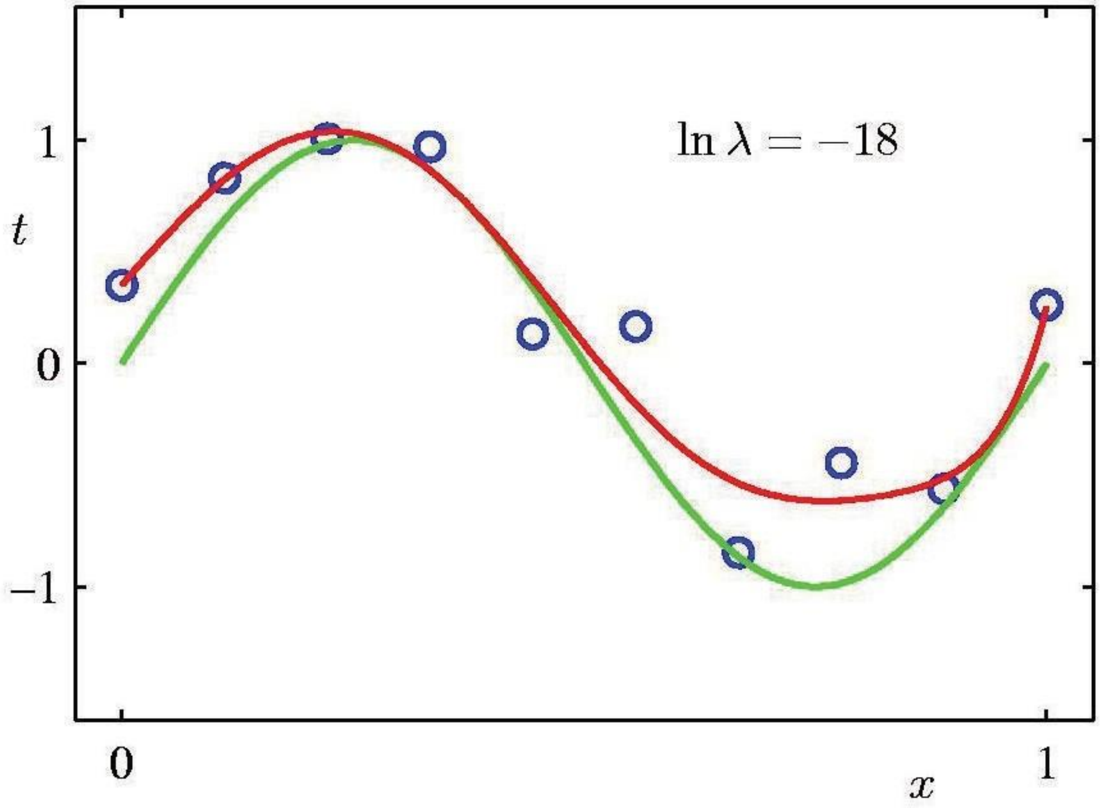
- Also called *ElasticNet*
- Can work better than either type alone
- Can adjust hyperparameters to control which of the two penalties is more important

Regularization

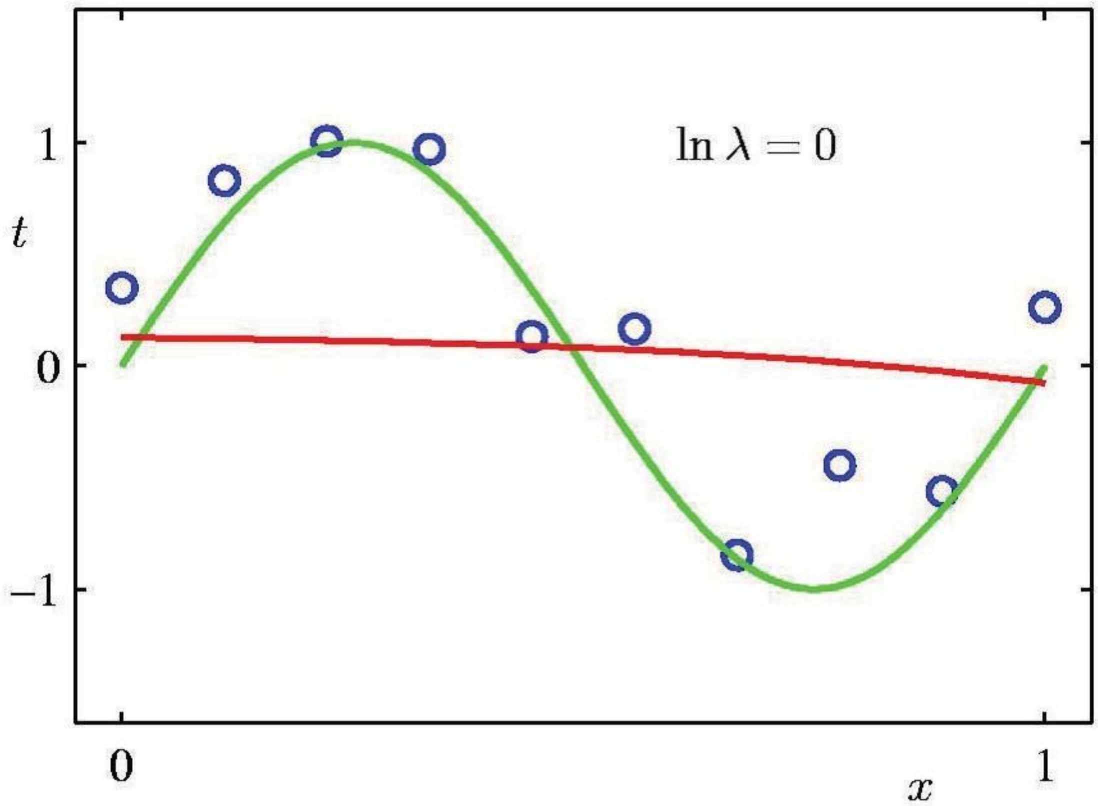
Penalize large coefficient values

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

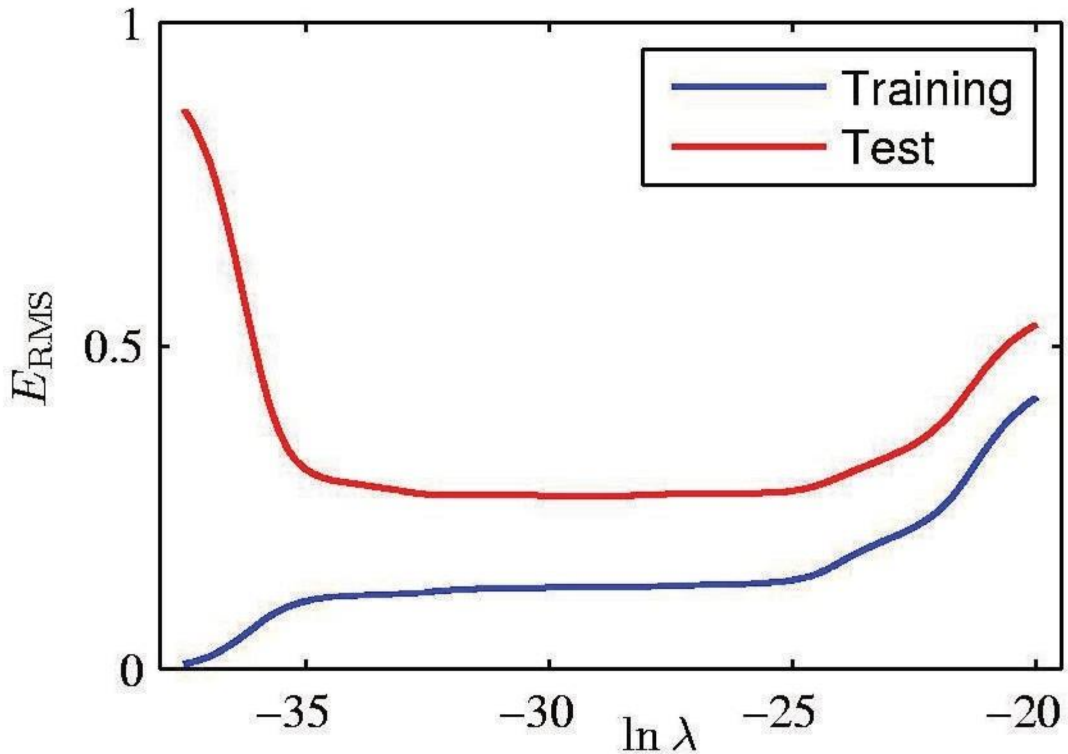
Regularization: $\ln \lambda = -18$



Regularization: $\ln \lambda = 0$



Regularization: E_{RMS} vs. $\ln \lambda$



Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01