

# Polynomial Curve Fitting

Karan Nathwani

# Topics

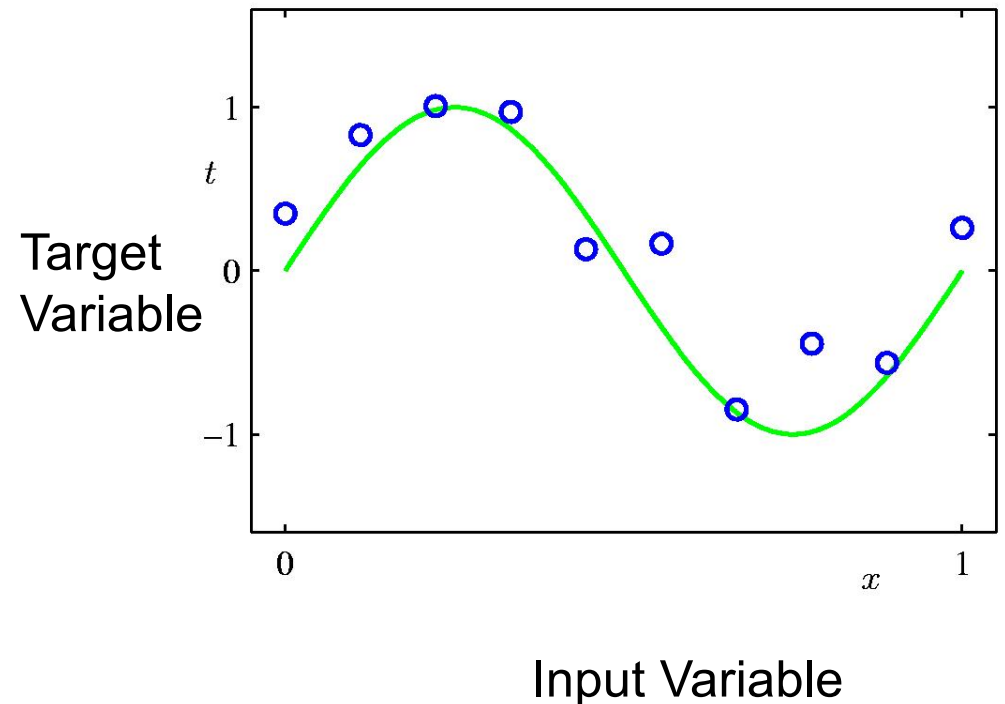
1. Simple Regression Problem
2. Polynomial Curve Fitting
3. Probability Theory of multiple variables
4. Maximum Likelihood
5. Bayesian Approach
6. Model Selection
7. Curse of Dimensionality

# Simple Regression Problem

- Begin discussion on ML by introducing a simple regression problem
  - It motivates a no. of key concepts
- Problem:
  - Observe Real-valued input variable  $x$
  - Use  $x$  to predict value of target variable  $t$
- We consider an artificial example using synthetically generated data
  - Because we know the process that generated the data, it can be used for comparison against a learned model

# Synthetic Data for Regression

- Data generated from the function  $\sin(2\pi x)$ 
  - Where  $x$  is the input
- Random noise in target values



Input values  $\{x_n\}$  generated uniformly in range  $(0,1)$ . Corresponding target values  $\{t_n\}$  Obtained by first computing corresponding values of  $\sin\{2\pi x\}$  then adding random noise with a Gaussian distribution with std dev 0.3

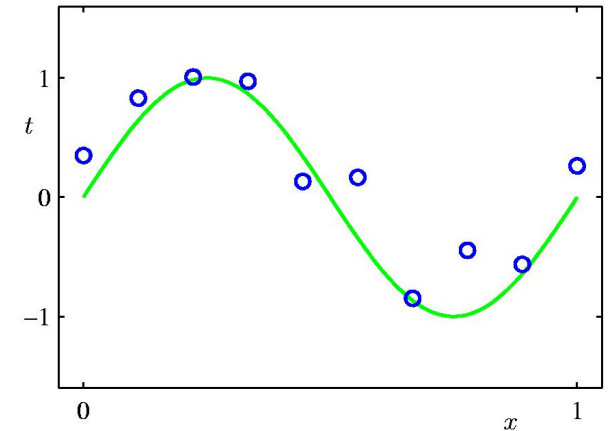
# Training Set

- $N$  observations of  $x$

$$\mathbf{x} = (x_1, \dots, x_N)^T$$

$$\mathbf{t} = (t_1, \dots, t_N)^T$$

- Goal is to exploit training set to predict value  $\hat{t}$  for some new value  $\hat{x}$
- Inherently a difficult problem
- Probability theory provides framework for expressing uncertainty in a precise, quantitative manner
- Decision theory allows us to make a prediction that is optimal according to appropriate criteria



Data Generation:  
 $N = 10$   
Spaced uniformly in range  $[0,1]$   
Generated from  $\sin(2\pi x)$  by adding small Gaussian noise  
Noise typical due to unobserved variables

# A Simple Approach to Curve Fitting

- Fit the data using a *polynomial function*

$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

– where  $M$  is the order of the polynomial

- Is higher value of  $M$  better? We'll see shortly!
- Coefficients  $w_0, \dots, w_M$  are collectively denoted by vector  $\mathbf{w}$
- It is a nonlinear function of  $x$ , but a linear function of the unknown parameters  $\mathbf{w}$
- Have important properties and are called Linear Models

# Error Function

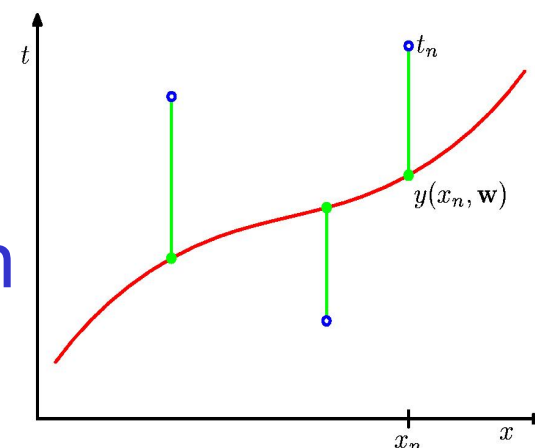
- We can obtain a fit by minimizing an error function
  - Sum of squares of the errors between the predictions  $y(x_n, \mathbf{w})$  for each data point  $x_n$  and target value  $t_n$

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

- Factor  $1/2$  included for later convenience

- Solve by choosing value of  $\mathbf{w}$  for which small as possible

Red line is best polynomial fit



# Minimization of Error Function

- Error function is a quadratic in coefficients  $\mathbf{w}$
- Thus derivative with respect to coefficients will be linear in elements of  $\mathbf{w}$
- Thus error function has a unique solution which can be found in closed form
  - Unique minimum denoted  $\mathbf{w}^*$
- Resulting polynomial is  $y(x, \mathbf{w}^*)$

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

Since  $y(x, \mathbf{w}) = \sum_{j=0}^M w_j x^j$

---


$$\begin{aligned} \frac{\partial E(\mathbf{w})}{\partial w_i} &= \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\} x_n^i \\ &= \sum_{n=1}^N \left\{ \sum_{j=0}^M w_j x_n^j - t_n \right\} x_n^i \end{aligned}$$

Setting equal to zero

$$\sum_{n=1}^N \sum_{j=0}^M w_j x_n^{i+j} = \sum_{n=1}^N t_n x_n^i$$

Set of  $M+1$  equations ( $i=0, \dots, M$ )

over  $M+1$  variables are

solved to get elements of  $\mathbf{w}^*$



# Solving Simultaneous equations

- $\mathbf{Aw} = \mathbf{b}$

where  $\mathbf{A}$  is  $N \times (M+1)$  : Design Matrix

$\mathbf{w}$  is  $(M+1) \times 1$ : set of weights to be determined

$\mathbf{b}$  is  $N \times 1$  : Target values

- Can be solved using matrix inversion

$$\mathbf{w} = \mathbf{A}^{-1}\mathbf{b}$$

- Or by using Gaussian elimination

# Solving Linear Equations

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & = & b_m \end{array}$$

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

1. Matrix Formulation:  $\mathbf{Ax}=\mathbf{b}$   
Solution:  $\mathbf{x}=\mathbf{A}^{-1}\mathbf{b}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Here  $m=n=M+1$

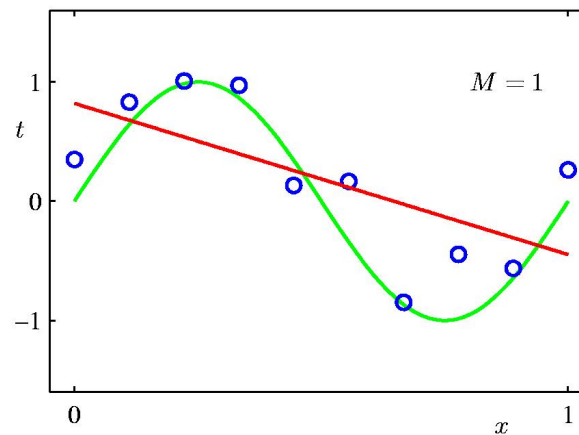
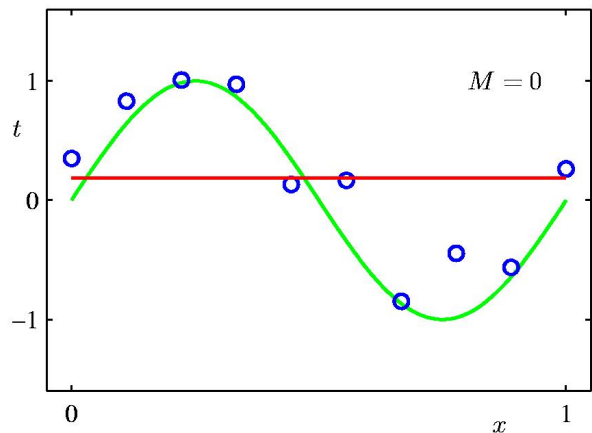
2. Gaussian Elimination followed by back-substitution

$$\begin{array}{rcl} x + 3y - 2z & = & 5 \\ 3x + 5y + 6z & = & 7 \\ 2x + 4y + 3z & = & 8 \end{array}$$

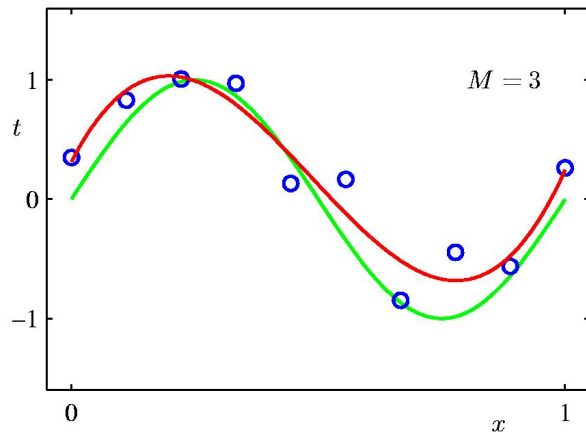
$$\begin{array}{c} \begin{array}{ccc} \text{L}_2 - 3\text{L}_1 \rightarrow \text{L}_2 & \text{L}_3 - 2\text{L}_1 \rightarrow \text{L}_3 & -\text{L}_2/4 \rightarrow \text{L}_2 \end{array} \\ \left[ \begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 3 & 5 & 6 & 7 \\ 2 & 4 & 3 & 8 \end{array} \right] \sim \left[ \begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & -4 & 12 & -8 \\ 2 & 4 & 3 & 8 \end{array} \right] \sim \left[ \begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & -4 & 12 & -8 \\ 0 & -2 & 7 & -2 \end{array} \right] \sim \left[ \begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & -3 & 2 \\ 0 & -2 & 7 & -2 \end{array} \right] \\ \sim \left[ \begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 1 & 2 \end{array} \right] \sim \left[ \begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array} \right] \sim \left[ \begin{array}{ccc|c} 1 & 3 & 0 & 9 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array} \right] \sim \left[ \begin{array}{ccc|c} 1 & 0 & 0 & -15 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array} \right] \end{array}$$

# Choosing the order of $M$

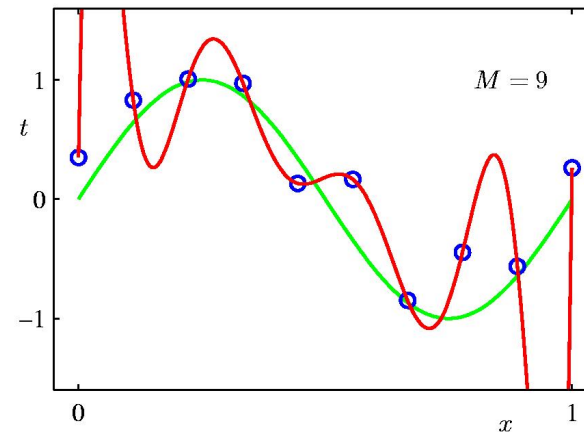
- Model Comparison or Model Selection
- Red lines are best fits with
  - $M = 0, 1, 3, 9$  and  $N=10$



← Poor representations of  $\sin(2\pi x)$



← Best Fit to  $\sin(2\pi x)$



Over Fit  
Poor representation of  $\sin(2\pi x)$

# Generalization Performance

- Consider separate *test* set of 100 points
- For each value of  $M$  evaluate

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

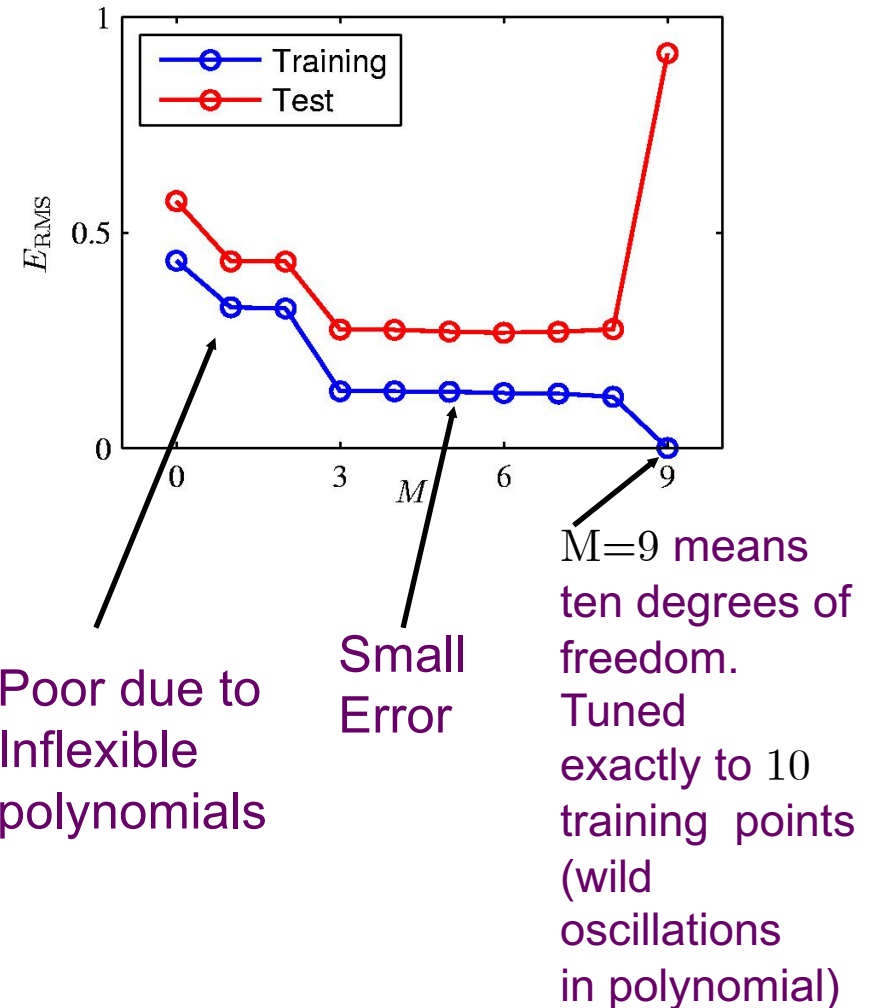
$$y(x, w^*) = \sum_{j=0}^M w_j^* x^j$$

for training data and test data

- Use RMS error

$$E_{RMS} = \sqrt{2E(w^*) / N}$$

- Division by  $N$  allows different sizes of  $N$  to be compared on equal footing
- Square root ensures  $E_{RMS}$  is measured in same units as  $t$



# Values of Coefficients $w^*$ for different polynomials of order $M$

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

As  $M$  increases magnitude of coefficients increases

At  $M=9$  finely tuned to random noise in target values

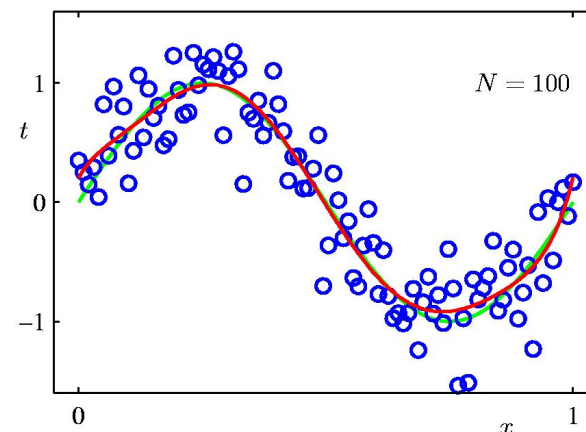
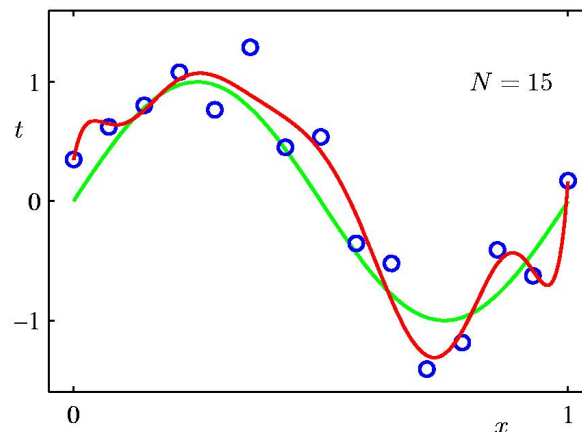
# Increasing Size of Data Set

$N=15, 100$

For a given model complexity  
overfitting problem is less  
severe as size of data set  
increases

Larger the data set, the more  
complex we can afford to fit  
the data

Data should be no less than 5  
to 10 times adaptive  
parameters in model



# Least Squares is case of Maximum Likelihood

- Unsatisfying to limit the number of parameters to size of training set
- More reasonable to choose model complexity according to problem complexity
- Least squares approach is a specific case of maximum likelihood
  - Over-fitting is a general property of maximum likelihood
- Bayesian approach avoids over-fitting problem
  - No. of parameters can greatly exceed no. of data points
  - Effective no. of parameters adapts automatically to size of data set

# Regularization of Least Squares

- Using relatively complex models with data sets of limited size
- Add a penalty term to error function to discourage coefficients from reaching large values

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

where

$$\|w\|^2 \equiv w^T w = w_0^2 + w_1^2 + \dots + w_M^2$$

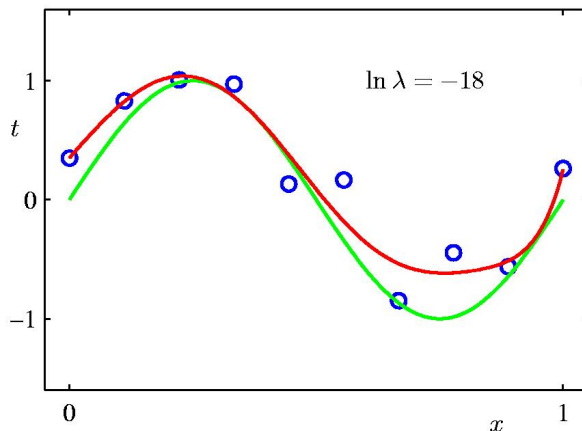
- $\lambda$  determines relative importance of regularization term to error term
- Can be minimized exactly in closed form
- Known as *shrinkage* in statistics  
*Weight decay* in neural networks



# Effect of Regularizer

$M=9$  polynomials using regularized error function

Optimal



No Regularizer

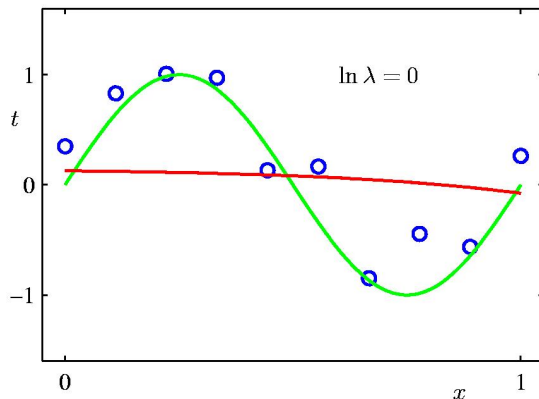
$\lambda = 0$

Large Regularizer

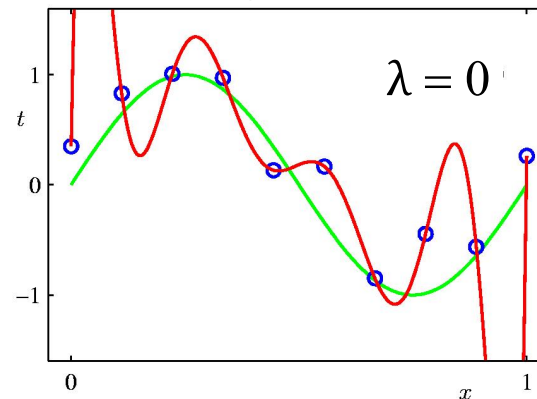
$\lambda = 1$

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

Large Regularizer

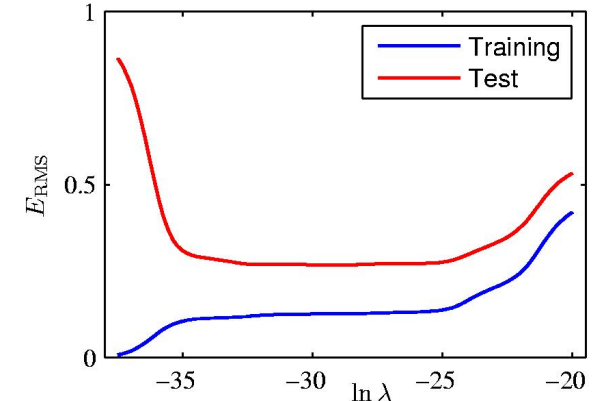


No Regularizer



# Impact of Regularization on Error

- $\lambda$  controls the complexity of the model and hence degree of over-fitting
  - Analogous to choice of  $M$
- Suggested Approach:
- Training set
  - to determine coefficients  $\mathbf{w}$
  - For different values of ( $M$  or  $\lambda$ )
- Validation set (holdout)
  - to optimize model complexity ( $M$  or  $\lambda$ )



$M=9$  polynomial

# Summary of Curve Fitting

- Partitioning data into *training set* (to determine coefficients  $w$ ) and a separate *validation set* (or *hold-out set*) to optimize model complexity  $M$  or  $\lambda$
- More sophisticated approaches are not as wasteful of training data
- More principled approach is based on probability theory
- Classification is a special case of regression where target value is discrete values