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

x i/p

y → o/p = f(x)

Simple Regression Problem

- Begin discussion on ML by introducing a simple regression problem
  - It motivates a no. of key concepts

training data

weights learned i/p → o/p

function known

- Problem:
  - Observe Real-valued input variable x
  - Use new 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

t - o/p
x - i/p

Synthetic Data for Regression

predict ??

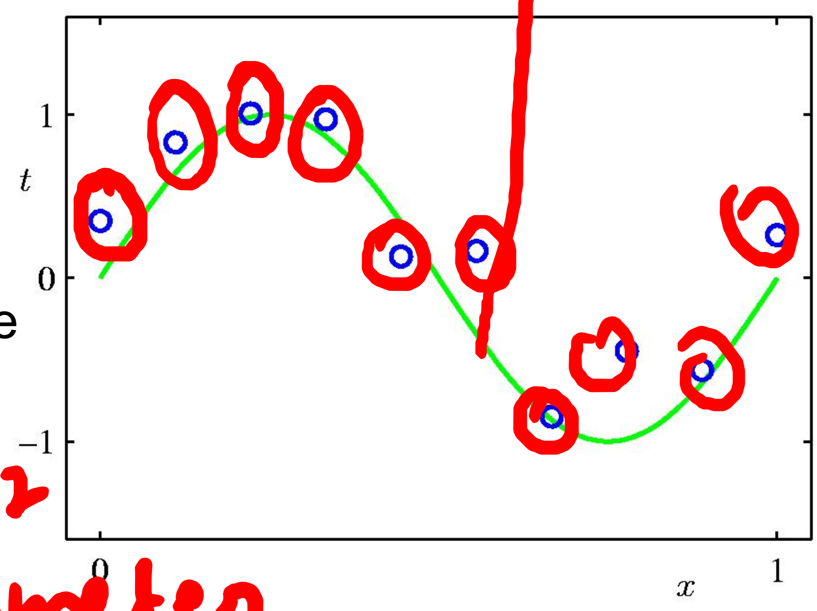
- Data generated from the function $\sin(2\pi\ x)$
  – Where x is the input
- Random noise in target values

A o h → C- B- R-

B o h → C B R

– Dict w_0
– Regularize w_1
– Regularize w_2
→ Parameter

Target Variable



Input Variable

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

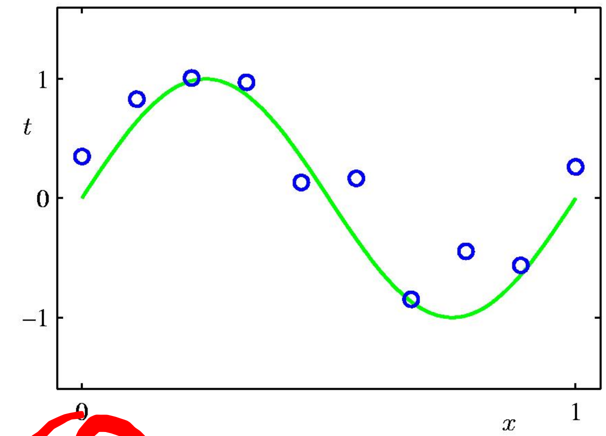
Training Set

N=10

- N observations of x

  $\mathrm{x} = (x_1,..,x_N)^T$ i/p

  $\mathrm{t} = (t_1,..,t_N)^T$ o/p



- 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:
$\mathrm{N} = 10$
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

finite   mapping

- Fit the data using a *polynomial function*   nth order

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

  - where M is the order of the polynomial
- Is higher value of M better? We'll see shortly!
- Coefficients $w_0 ,\ldots w_M$ 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

x - fixed

N - Sample

- 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

Actual

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

Predicted

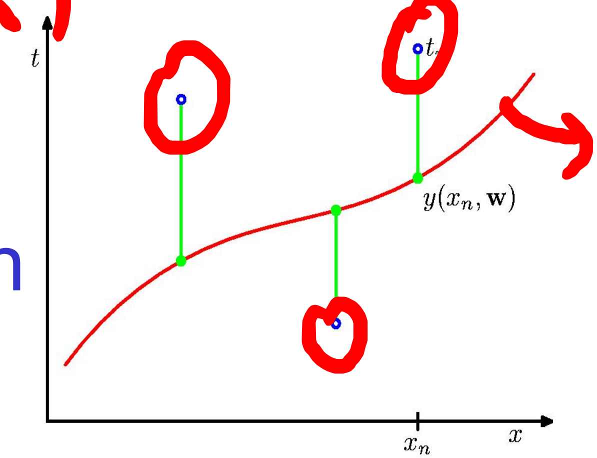
  - Factor ½ included for later convenience

$(y - t_n)$

$(t_n - y)$

- Solve by choosing value of w for which small as possible

Red line is best polynomial fit



$\frac{1}{2}\frac{d}{dx}(x^2) = \frac{2x}{2} = x$

Minimization of Error Function

1st — 11/12th Class

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

Set of Simultaneous linear equations

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

Details

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

$$\frac{\partial E(\mathrm{w})}{\partial w_i} = \sum_{n=1}^{N}\{y(x_n,\mathrm{w}) - t_n\}x_n^i$$

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

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,..,M) over M+1 variables are solved to get elements of w^*

Solving Simultaneous equations

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

  where 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

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

- Or by using Gaussian elimination

$w = A^{-1} b$

unstable

$A \Rightarrow |\det A| = 0$

$O(n^3)$

Solving Linear Equations

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

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

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

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

Here $m=n=M+1$

2. Gaussian Elimination followed by back-substitution

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

$L_2-3L_1 \to L_2$ $\quad$ $L_3-2L_1 \to L_3$ $\quad$ $-L_2/4 \to L_2$

$$\left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 3 & 5 & 6 & 7 \\ 2 & 4 & 3 & 8 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & -4 & 12 & -8 \\ 2 & 4 & 3 & 8 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & -4 & 12 & -8 \\ 0 & -2 & 7 & -2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & -3 & 2 \\ 0 & -2 & 7 & -2 \end{array}\right]$$

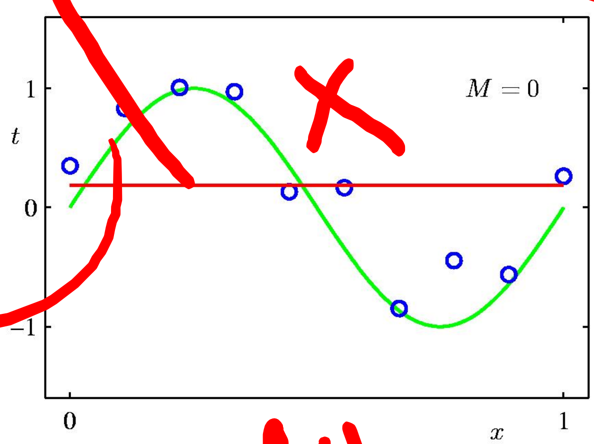
$$\sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 1 & 2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 3 & 0 & 9 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 0 & -15 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

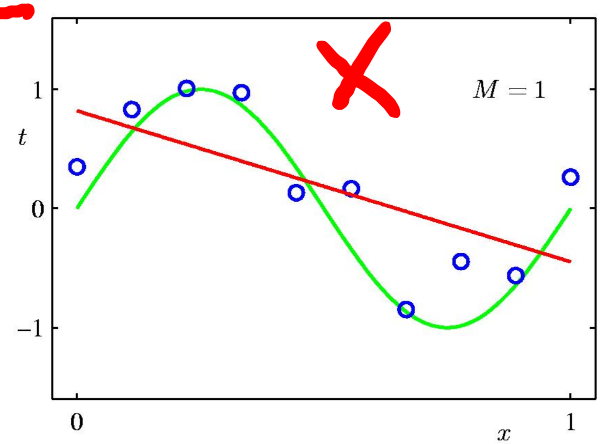
Choosing the order of M

Predicted

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

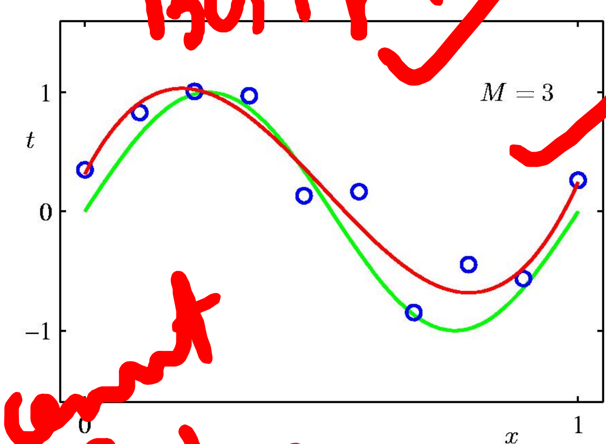
$y(x, w) = w_0 + w_1 x^1 + w_2 x^2 + \dots + w_M x^M$



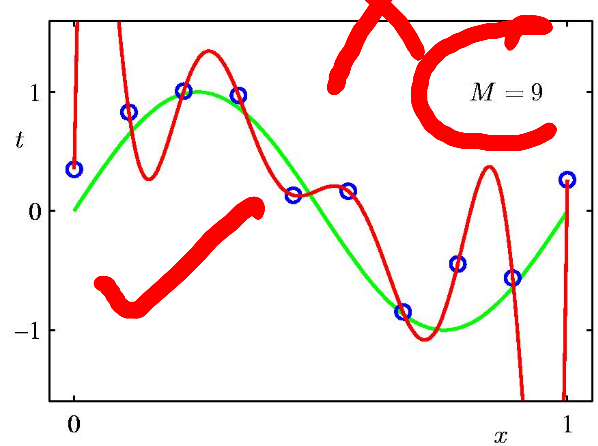


← Poor representations of $\sin(2\pi x)$

Best fit



← Best Fit to $\sin(2\pi x)$



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

Placement c 9pil A, B → clam same exam in test 100%

100 Students - 20, 20, 20, 20, 20 K-fold Validation

Generalization Performance

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

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

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

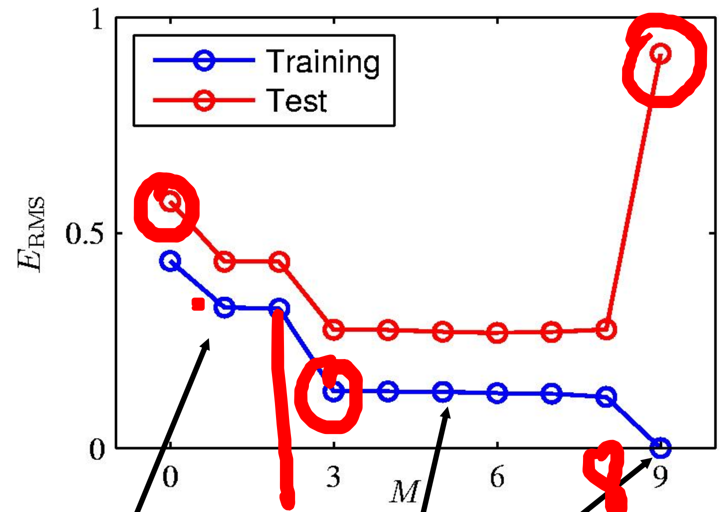
for training data and test data

- Use RMS error

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

  – 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

overfit



2

8

Poor due to Inflexible polynomials

Small Error

M=9 means ten degrees of freedom. Tuned exactly to 10 training points (wild oscillations in polynomial)

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

w0   w0, 1   w0, 1, 2, 3

	$M = 0$	$M = 1$	$M=3$	$M = 9$
$w_0^\star$	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

10^{5-6}

overfitting

As M increases magnitude of coefficients increases
At $M=9$ finely tuned to random noise in target values

$N = 10$ sambler $M = 9$

Increasing Size of Data Set

$M \downarrow$

$M=9$

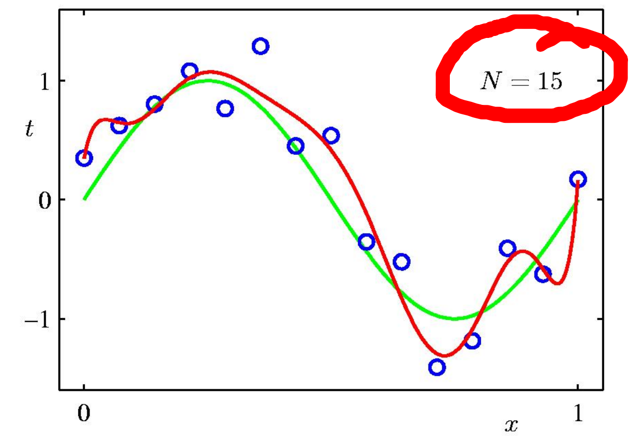
$N \uparrow$

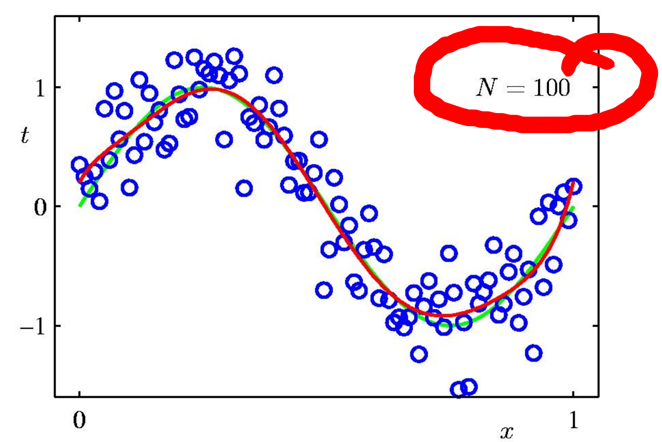
N=15, 100

- 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

thumb rule

- 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

M=9 , N=10

- Add a penalty term to error function to discourage coefficients from reaching large values

w_0 - 10

w_0 - 3

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

where

$$||\,\mathrm{w}^2\,|| \equiv w^T w = w_0^{\,2} + w_1^{\,2} + .. + w_M^{\,2}$$

w_4 - 0.0 - -

- λ determines relative importance of regularization term to error term
- Can be minimized exactly in closed form
- Known as *shrinkage* in statistics

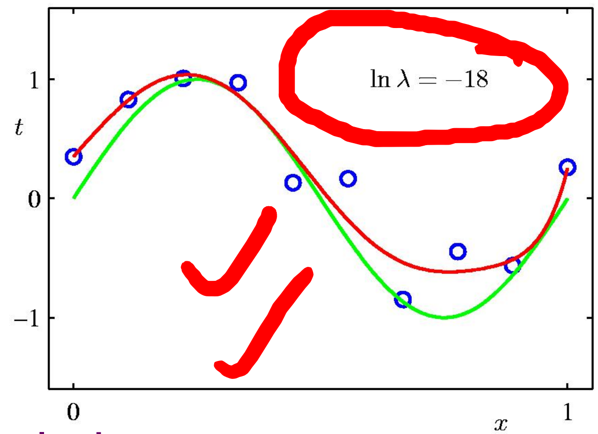
  Weight decay in neural networks

$0 < \lambda < 1$

Effect of Regularizer $\ln \lambda = 0$

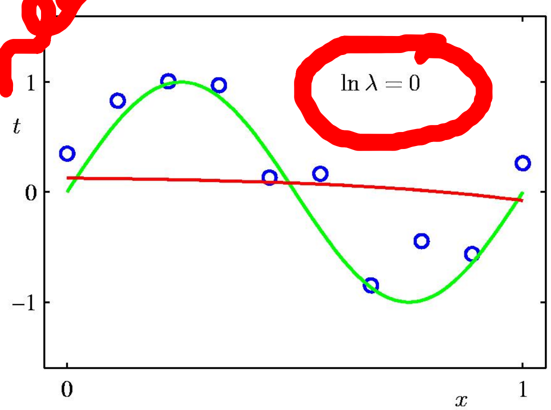
M=9 polynomials using regularized error function

Optimal



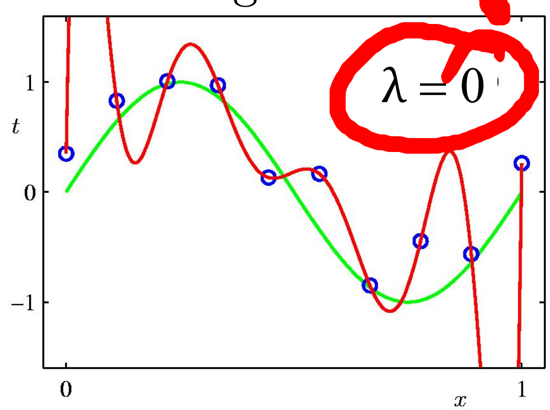
Not learned

Large Regularizer



overfitted

No Regularizer



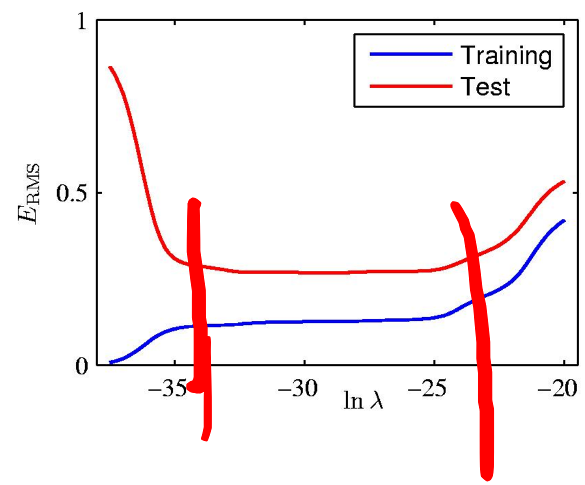
No Regularizer $\lambda = 0$

Large Regularizer $\lambda = 1$

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	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

Impact of Regularization on Error

- λ 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 λ)
- Validation set (holdout)
  - to optimize model complexity (M or λ)



$M=9$ polynomial

Summary of Curve Fitting

- Partitioning data into *training set* (to determine coefficients $\mathbf{w}$) and a separate *validation set* (or *hold-out* set) to optimize model complexity M or λ
- 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