Polynomial Curve Fitting

Sargur N. Srihari

Topics

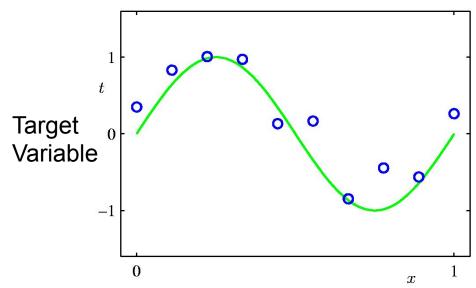
- 1. Simple Regression Problem
- 2. Polynomial Curve Fitting
- 3. Probability Theory of multiple variables
- 4. Maximum Likelihood
- 5. Bayesian Approach
- 6. Model Selection
- 7. Curse of Dimensionality

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 Variable

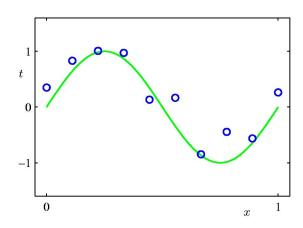
Input values $\{x_n\}$ generated uniformly in range (0,1). Corresponding target values $\{t_n\}$ Obtained by first computing corresponding values of $\sin\{2\pi x\}$ then adding random noise with a Gaussian distribution with std dev 0.3

Training Set

• N observations of x

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

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



- 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=10Spaced 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

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{i=0}^{M} w_i x^i$$

- 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 ${\bf w}$
- It is a nonlinear function of x, but a linear function of the unknown parameters
- Have important properties and are called Linear Models

Error Function

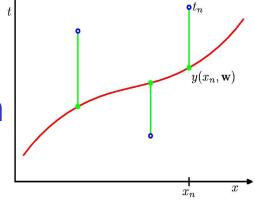
- · We can obtain a fit by minimizing an error function
 - Sum of squares of the errors between the predictions $y(x_n, \mathbf{w})$ for each data point x_n and target value t_n

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

- Factor ½ included for later convenience

Red line is best polynomial fit

 Solve by choosing value of w for which small as possible



Minimization of Error Function

- 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 $y(x, w^*)$

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

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

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

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

- Aw=b
 where A is N × (M+1)
 w is (M+1) × 1: set of weights to be determined
 b is N × 1
- Can be solved using matrix inversion $\mathbf{w} = A^{-1}\mathbf{b}$
- Or by using Gaussian elimination

Srihari Machine Learning

Solving Linear Equations

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

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

Here
$$m=n=M+1$$

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

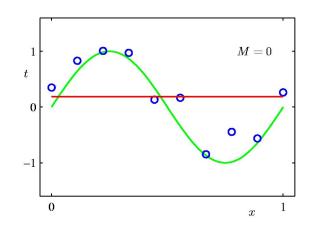
2. Gaussian Elimination followed by back-substitution

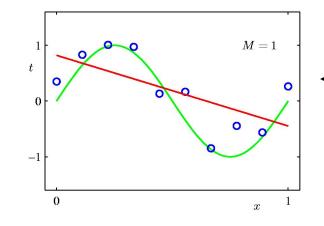
$$x + 3y - 2z = 5$$
$$3x + 5y + 6z = 7$$
$$2x + 4y + 3z = 8$$

Choosing the order of M

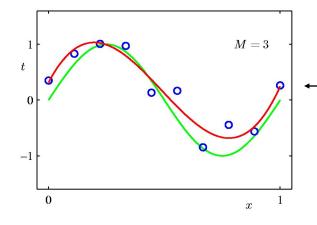
- Model Comparison or Model Selection
- Red lines are best fits with

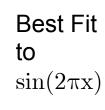
$$- M = 0.1.3.9 \text{ and } N = 10$$

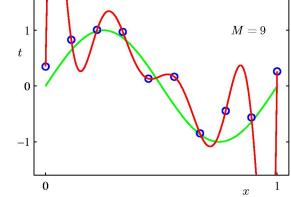




Poor representations of $\sin(2\pi x)$







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

Generalization Performance

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

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

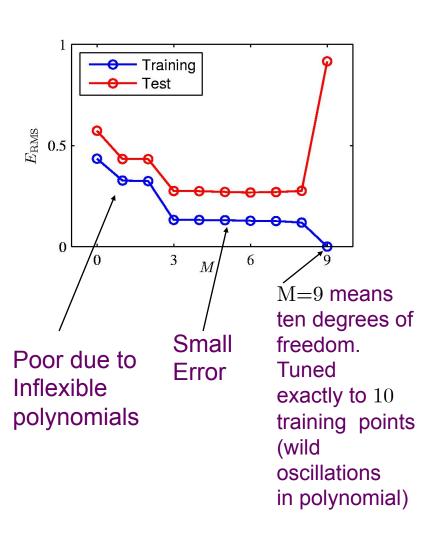
$$y(x, \mathbf{w}^*) = \sum_{j=0}^{M} w_j^* x^j$$

for training data and test data

Use RMS error

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

- 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



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

	M - 0	M _ 1	M C	14. 0
	M = 0	$\underline{M} = \underline{1}$	M=6	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

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

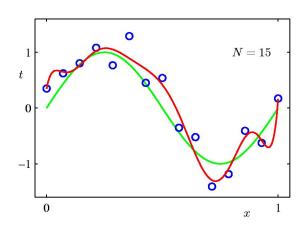
Increasing Size of Data Set

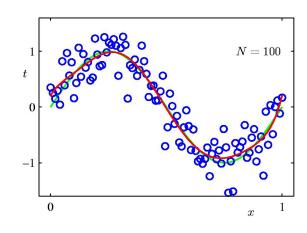
N=15, 100

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



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

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

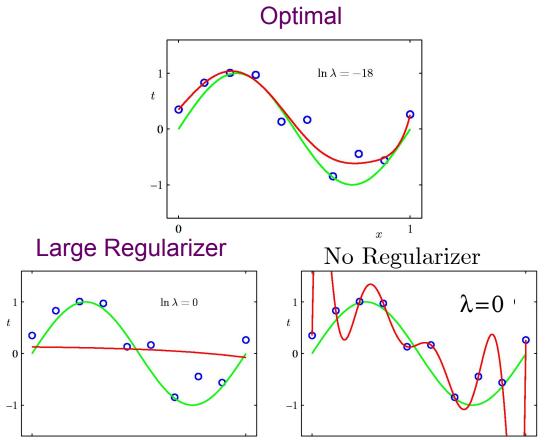
where

$$\mid\mid \mathbf{w}^{2}\mid\mid \equiv w^{T}w = w_{0}^{2} + w_{1}^{2} + .. + w_{M}^{2}$$

- 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

Effect of Regularizer

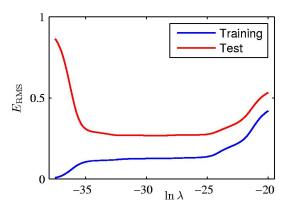
M=9 polynomials using regularized error function



No Reg λ=	gularizer	Large Regularizer λ= 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	
$w_5^{ ilde{\star}}$	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 overfitting
 - Analogous to choice of M
- Suggested Approach:
- Training set
 - to determine coefficients w
 - For different values of $(M \text{ or } \lambda)$
- 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 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