

Basic Concepts in Machine Learning

Sargur N. Srihari

Introduction to ML: Topics

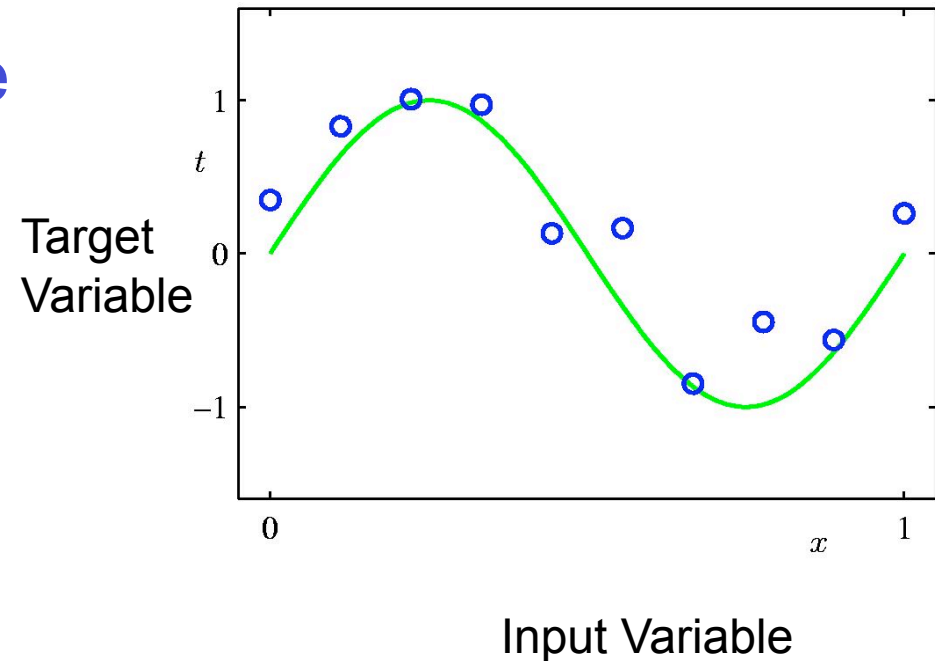
1. Polynomial Curve Fitting
2. Probability Theory of multiple variables
3. Maximum Likelihood
4. Bayesian Approach
5. Model Selection
6. Curse of Dimensionality

Polynomial Curve Fitting

Sargur N. Srihari

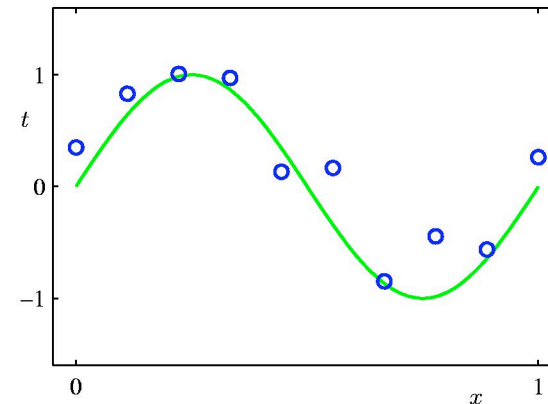
Simple Regression Problem

- Observe Real-valued input variable x
- Use x to predict value of target variable t
- Synthetic data generated from $\sin(2\pi x)$
- Random noise in target values



Notation

- N observations of x
 $\mathbf{x} = (x_1, \dots, x_N)^\top$
 $\mathbf{t} = (t_1, \dots, t_N)^\top$
- Goal is to exploit training set to predict value of \hat{t} from \mathbf{x}
- Inherently a difficult problem
- Probability theory allows us to make a prediction



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

Polynomial Curve Fitting

- 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 denoted by vector \mathbf{w}
- Nonlinear function of x , linear function of coefficients \mathbf{w}
- Called Linear Models

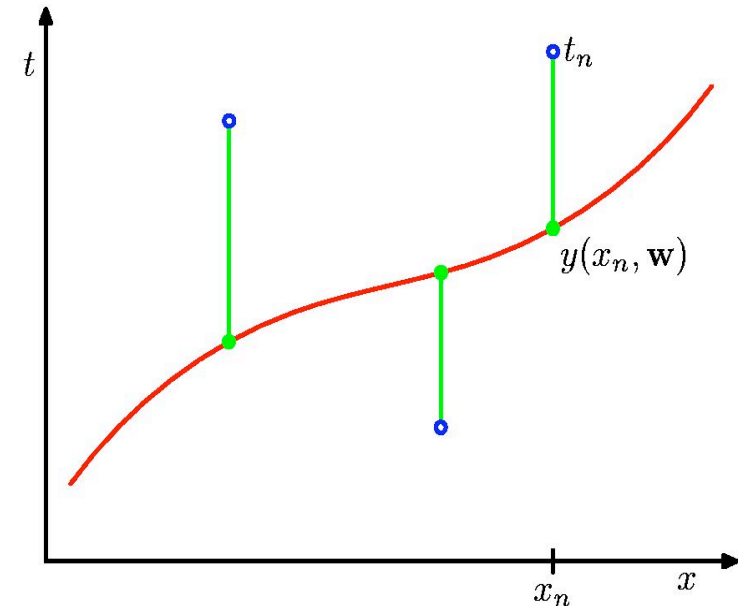
Error Function

- 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 $E(\mathbf{w})$ is as small as possible

Red line is best polynomial fit



Minimization of Error Function

- Error function is a quadratic in coefficients \mathbf{w}
- Derivative with respect to coefficients will be linear in elements of \mathbf{w}
- Thus error function has a 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 $y(x, \mathbf{w}) = \sum_{j=0}^M w_j x^j$

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

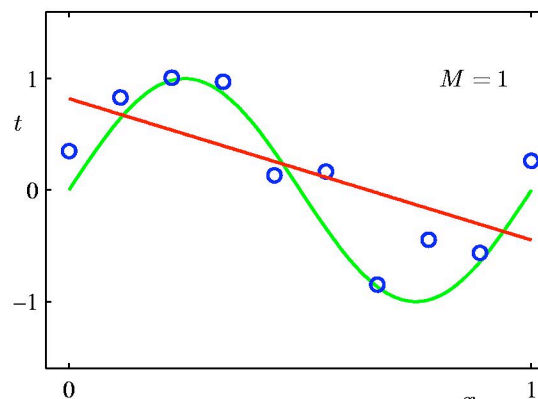
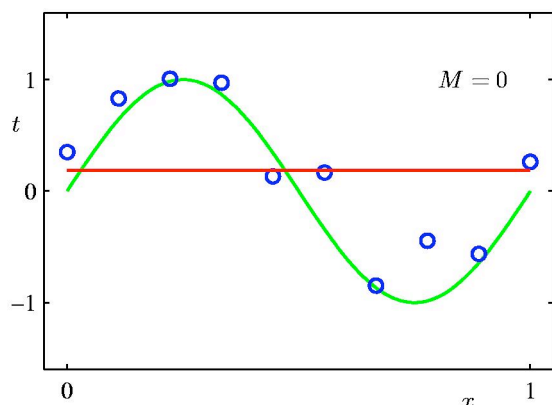
Setting equal to zero

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

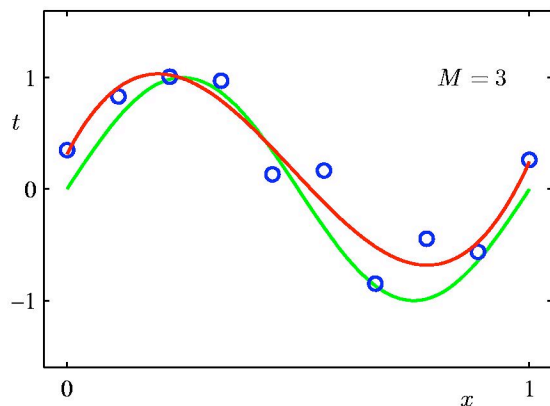
Set of $M+1$ equations ($i = 0, \dots, M$) are solved to get elements of \mathbf{w}^*

Choosing the order of M

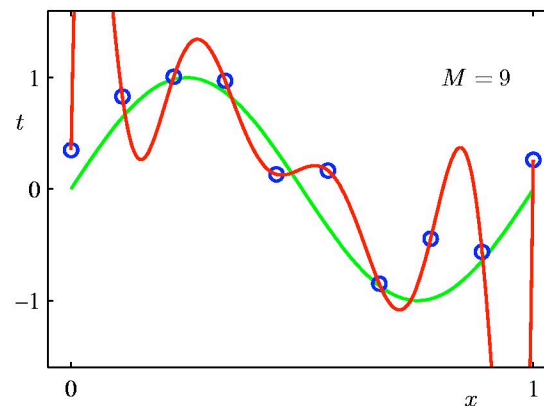
- Model Comparison or Model Selection
- Red lines are best fits with
 - $M = 0, 1, 3, 9$ and $N=10$



← 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

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

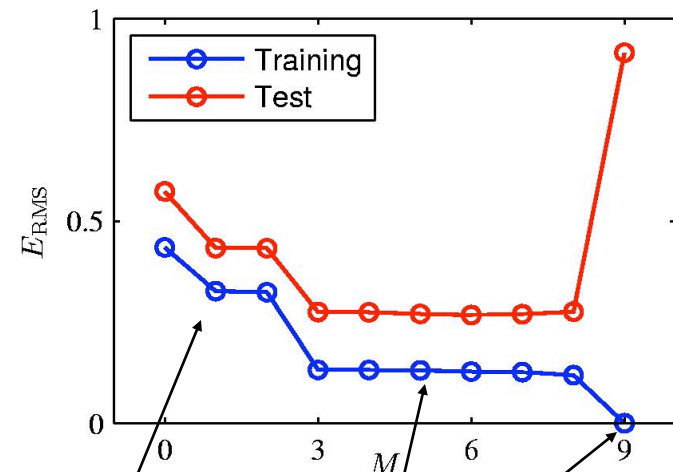
$$y(x, w^*) = \sum_{j=0}^M w_j^* x^j$$

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



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

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
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

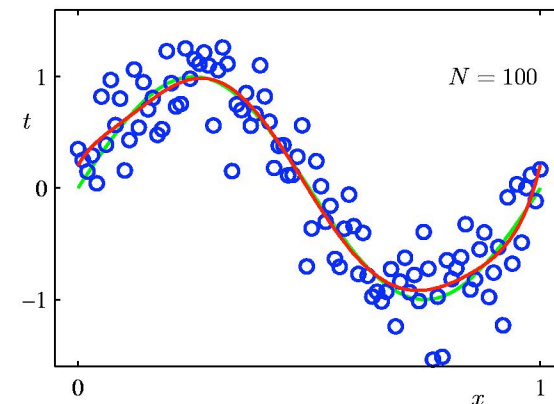
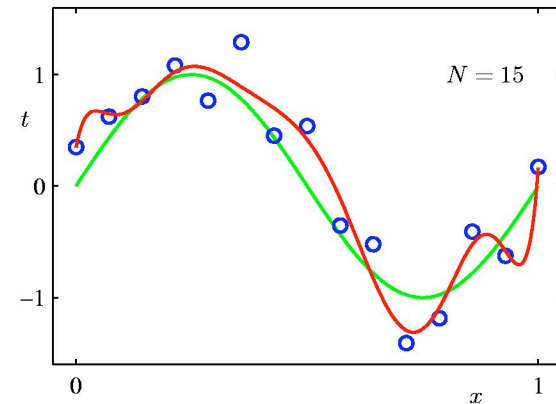
Increasing Size of Data Set

$N=15, 100$

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

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

where

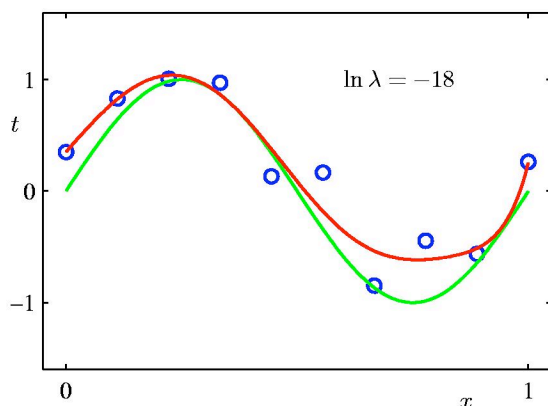
$$\|\mathbf{w}\|^2 \equiv \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$$

- 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

Optimal

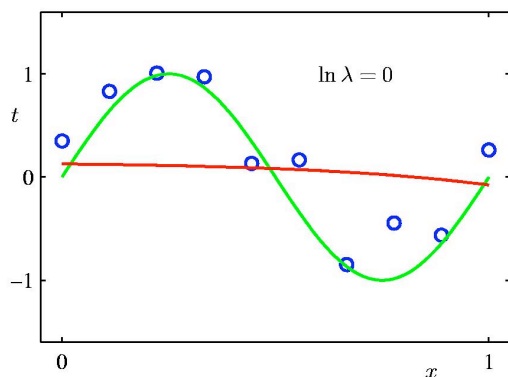


No
Regularizer
 $\lambda = 0$

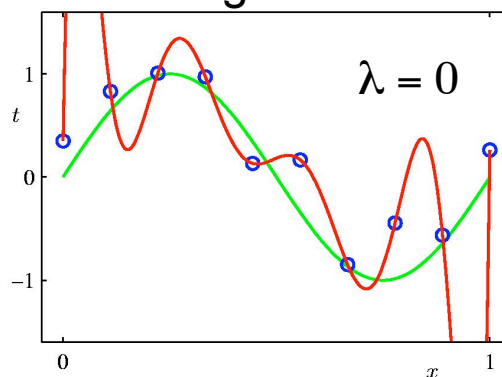
Large
Regularizer
 $\lambda = 1$

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
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

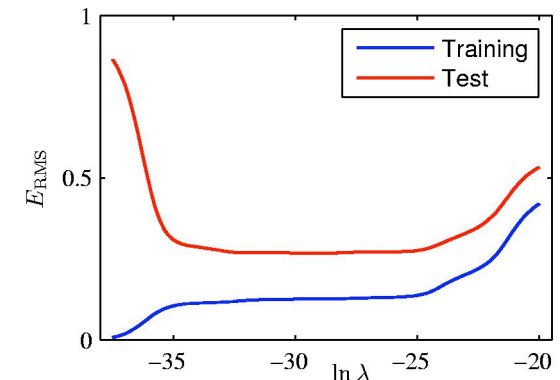


No Regularizer



Impact of Regularization on Error

- 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 w
 - For different values of (M or λ)
- Validation set (holdout)
 - to optimize model complexity (M or λ)



$M=9$ polynomial

Concluding Remarks on Regression

- Approach suggests partitioning data into training set to determine coefficients \mathbf{w}
- 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