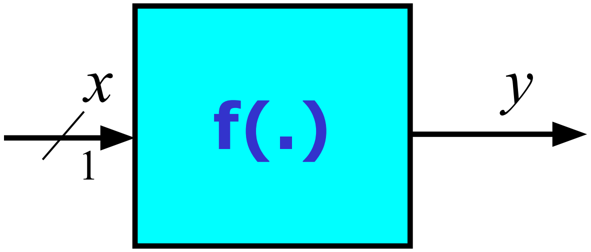
Supervised Machine Learning: Regression

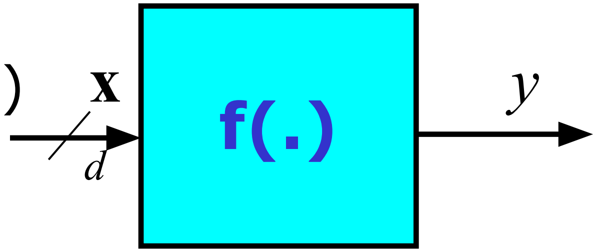
Nonlinear Regression

Nonlinear Regression

- Nonlinear approach to model the relationship between a scalar response, (y) (or dependent variable) and one or more predictor variables, (x *or* $\mathbf{x}$) (or independent variables)
- The response is going to be the nonlinear function of input (one or more independent variables)
- Simple nonlinear regression (Polynomial curve fitting, Neural Network):
  - Single independent variable (x)
  - Single dependent variable (y)
  - *Fitting a curve*



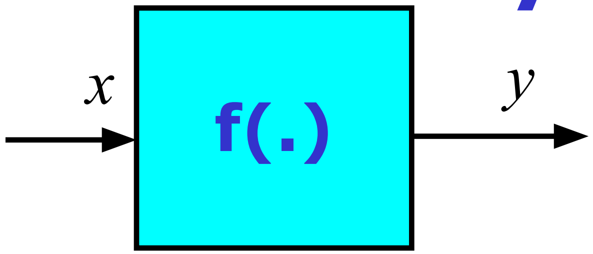
- Multiple nonlinear regression (Polynomial regression, Neural Network):
  - Two or more independent variable ($\mathbf{x}$)
  - Single dependent variable (y)
  - *Fitting a surface*



Supervised Machine Learning: Regression

Polynomial Curve Fitting

Polynomial Curve Fitting



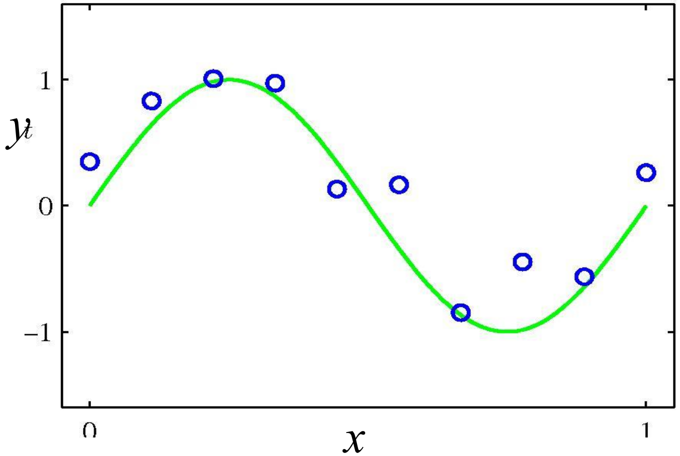
- Given:-Training data:

  $\text{D} = \{x_n, y_n\}_{n=1}^{N}, \; x_n \in \text{R}^1$ and $y_n \in \text{R}^1$

- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(x_n, \mathbf{w}) = w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_p x_n^p = \sum_{j=0}^{p} w_j x_n^j$$

- Here, $1, x_n, x_n^2, x_n^3, \ldots, x_n^p$ are the monomials of polynomial up to degree p



$y = f(x, \mathbf{w})$

- The coefficients $\mathbf{w}=[w_0, w_1, \ldots, w_p]$ are parameters of polynomial curve (regression coefficients) ***- Unknown***

- Polynomial function $f(x_n, \mathbf{w})$ is a nonlinear function of x_n and
- Function $f(x_n, \mathbf{w})$ is a linear function of coefficients $\mathbf{w}$
  - ***Linear model for regression***

Polynomial Curve Fitting: Training Phase

- Given:- Training data: $\mathrm{D} = \{x_n, y_n\}_{n=1}^{N}$, $x_n \in \mathrm{R}^1$ and $y_n \in \mathrm{R}^1$
- **Method of least squares**: Minimizes the sum of the squared error between
  - all the actual data (y_n) i.e. actual dependent variable and
  - the estimate of line (predicted dependent variable ($\hat{y}_n$)) i.e. the function $f(x_n, w, w_0)$, in the training set for any given value of $\mathbf{w}$

$$\hat{y}_n = f(x_n, \mathbf{w}) = w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_p x_n^p$$

$$\underset{\mathbf{w}}{\text{minimize}}\, E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(\hat{y}_n - y_n\right)^2$$

- Minimize the error such that the coefficients $\mathbf{w}$ represent the parameter of polynomial curve that best fit the training data

Polynomial Curve Fitting: Training Phase

- Given:- Training data: $D = \{x_n, y_n\}_{n=1}^{N}$, $x_n \in R^1$ and $y_n \in R^1$
- **Method of least squares**: Minimizes the sum of the squared error between
  - all the actual data (y_n) i.e. actual dependent variable and
  - the estimate of line (predicted dependent variable ($\hat{y}_n$)) i.e. the function $f(x_n, w, w_0)$, in the training set for any given value of $\mathbf{w}$

$$\hat{y}_n = f(x_n, \mathbf{w}) = w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_p x_n^p$$

$$\underset{\mathbf{w}}{\text{minimize}}\, E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left(\hat{y}_n - y_n\right)^2$$

- The error function is a
  - quadratic function of the coefficients $\mathbf{w}$ and
  - The derivatives of error function with respect to the coefficients will be linear in the elements of $\mathbf{w}$
- Hence the minimization of the error function has unique solution and found in closed form

Polynomial Curve Fitting: Training Phase

$$\hat{y}_n = f(x_n, \mathbf{w}) = w_0 + w_1 x_n + w_2 x_n^2 + \ldots + w_p x_n^p = \sum_{j=0}^{p} w_j x_n^j$$

- Let's consider: $x_n \quad x_n^2 \quad x_n^3 \quad \quad x_n^p$ $\quad p$ is degree of polynomial

$$\begin{array}{ccccc} x_n & x_n^2 & x_n^3 & & x_n^p \\ \downarrow & \downarrow & \downarrow & \cdots & \downarrow \\ z_{n1} & z_{n2} & z_{n3} & & z_{np} \end{array}$$

$$\hat{y}_n = f(\mathbf{z}_n, \mathbf{w}) = w_0 + w_1 z_{n1} + w_2 z_{n2} + \ldots + w_p z_{np}$$

$$\hat{y}_n = f(\mathbf{z}_n, \mathbf{w}) = \sum_{j=0}^{p} w_j z_{nj} = \mathbf{w}^{\mathsf{T}} \mathbf{z}_n$$

where $\mathbf{w} = [w_0, w_1, \ldots, w_p]^{\mathsf{T}}$ and $\mathbf{z}_n = [1, z_{n1}, \ldots, z_{np}]^{\mathsf{T}}$

Polynomial Curve Fitting: Training Phase

- Cost function for optimization:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(f(\mathbf{z}_n, \mathbf{w}) - y_n\right)^2$$

- Conditions for optimality: $\dfrac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \bar{0}$

- Application of optimality conditions gives optimal $\hat{\mathbf{w}}$:

$$\frac{\partial \frac{1}{2}\sum_{n=1}^{N}\left(\sum_{j=0}^{p} w_j z_{nj} - y_n\right)^2}{\partial \mathbf{w}} = \mathbf{0}$$

$$\frac{\partial \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{w}^{\top}\mathbf{z}_n - y_n\right)^2}{\partial \mathbf{w}} = \mathbf{0}$$

Polynomial Curve Fitting: Training Phase

- Cost function for optimization:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(f(\mathbf{z}_n, \mathbf{w}) - y_n\right)^2$$

- Conditions for optimality: $\dfrac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = 0$

- Application of optimality conditions gives optimal $\hat{\mathbf{w}}$:

$$\frac{\partial \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{w}^\top \mathbf{z}_n - y_n\right)^2}{\partial \mathbf{w}} = \mathbf{0}$$

$$\boxed{\hat{\mathbf{w}} = \left(\mathbf{Z}^\top \mathbf{Z}\right)^{-1} \mathbf{Z}^\top \mathbf{y}}$$

  - Assumption: $p < N$

$\mathbf{Z}$ is Vandermonde matrix

$$\mathbf{Z} = \begin{bmatrix} 1 & z_{11} & z_{12} & \dots & z_{1p} \\ 1 & z_{21} & z_{22} & \dots & z_{2p} \\ - & - & - & - & - \\ 1 & z_{n1} & z_{n2} & \dots & z_{np} \\ - & - & - & - & - \\ 1 & z_{N1} & z_{N2} & \dots & z_{Np} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ - \\ y_n \\ - \\ y_N \end{bmatrix}$$

where, $z_{nj} = x_n^j$

Polynomial Curve Fitting: Testing

- Optimal coefficient vector $\mathbf{w}$ is given by

$$\hat{\mathbf{w}} = \left(\mathbf{Z}^{\mathsf{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{\mathsf{T}}\mathbf{y}$$

$$\hat{\mathbf{w}} = \mathbf{Z}^{+}\mathbf{y}$$

where $\mathbf{Z}^{+} = \left(\mathbf{Z}^{\mathsf{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{\mathsf{T}}$ is the pseudo inverse of matrix $\mathbf{Z}$

- For any test example x, the predicted value is given by:

$$\hat{y} = f(x, \hat{\mathbf{w}}) = \hat{\mathbf{w}}^{\mathsf{T}}\mathbf{z} = \sum_{j=0}^{p} \hat{w}_i x^j$$

- The prediction accuracy is measured in terms of squared error: $E = \left(\hat{y} - y\right)^2$
- Let N_t be the total number of test samples
- The prediction accuracy of regression model is measured in terms of root mean squared error:

$$E_{\text{RMS}} = \sqrt{\frac{1}{N_t}\sum_{n=1}^{N_t}\left(\hat{y}_n - y_n\right)^2}$$

Determining p, Degree of Polynomial

- This is determined experimentally
- Starting with p=1, test set is used to estimate the accuracy, in terms of error, of the regression model
  - Note: The polynomial degree p=1 is equivalent to simple linear regression (straight-line regression)
- This process is repeated each time by incrementing p
- The regression model with p that gives the minimum error on test set may be selected

Illustration of Polynomial Curve Fitting: Humidity Prediction - Training

Temp (x)	Humidity (y)
25.47	82.19
26.19	83.15
25.17	85.34
24.30	87.69
24.07	87.65
21.21	95.95
23.49	96.17
21.79	98.59
25.09	88.33
25.39	90.43
23.89	94.54
22.51	99.00
22.90	98.00
21.72	99.00
23.18	98.97

- Degree of polynomial p : 1

$$\hat{\mathbf{w}} = \left(\mathbf{Z}^\top \mathbf{Z}\right)^{-1} \mathbf{Z}^\top \mathbf{y} \quad \mathbf{Z} \text{ is } 15 \times 2 \text{ matrix}$$

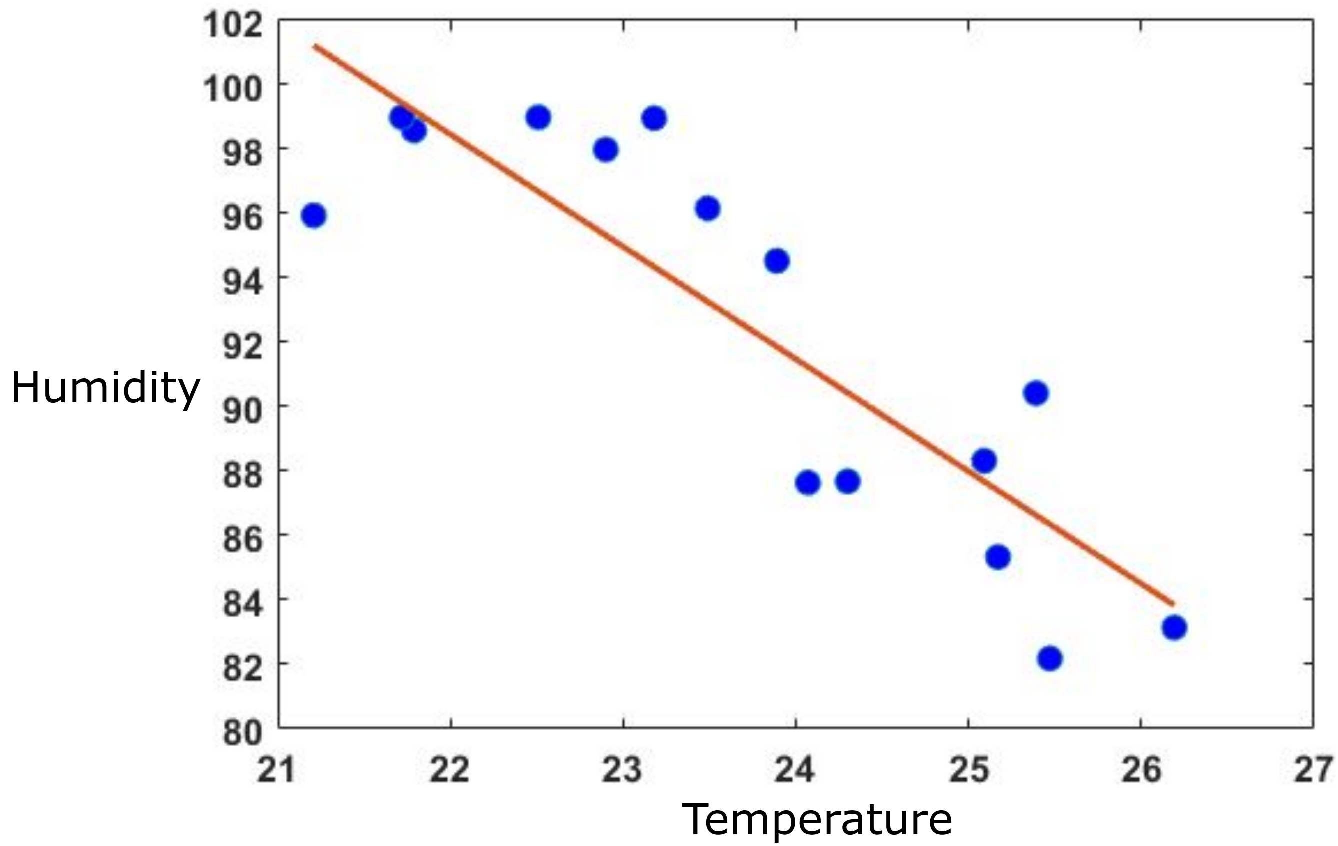
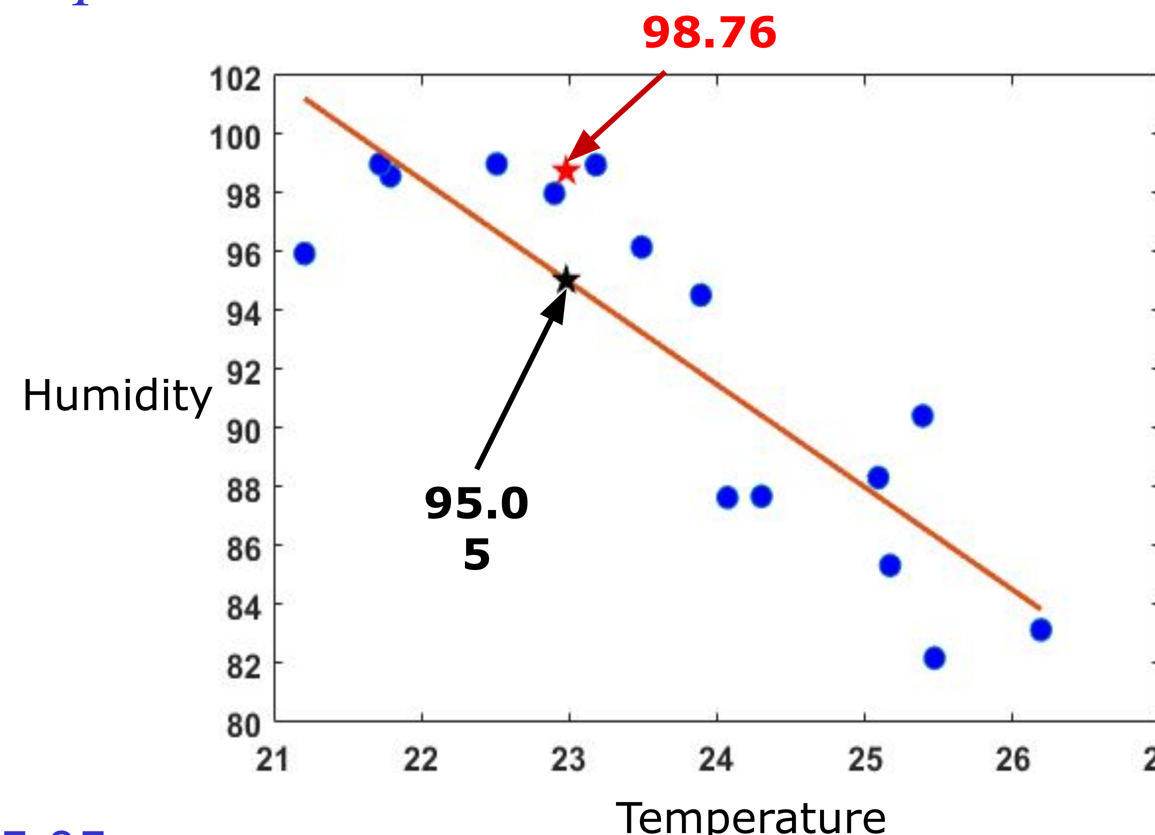


Illustration of Polynomial Curve Fitting: Humidity Prediction - Test

- Degree of polynomial p : 1

Temp (x)	Humidity (y)
22.98	--



- Predicted humidity: 95.05
- Actual humidity: 98.76
- Squared error: **13.77**

Illustration of Polynomial Curve Fitting: Humidity Prediction - Training

Temp (x)	Humidity (y)
25.47	82.19
26.19	83.15
25.17	85.34
24.30	87.69
24.07	87.65
21.21	95.95
23.49	96.17
21.79	98.59
25.09	88.33
25.39	90.43
23.89	94.54
22.51	99.00
22.90	98.00
21.72	99.00
23.18	98.97

- Degree of polynomial p : 2

$$\hat{\mathbf{w}} = \left(\mathbf{Z}^\top \mathbf{Z}\right)^{-1} \mathbf{Z}^\top \mathbf{y}$$ $\mathbf{Z}$ is 15 x 3 matrix

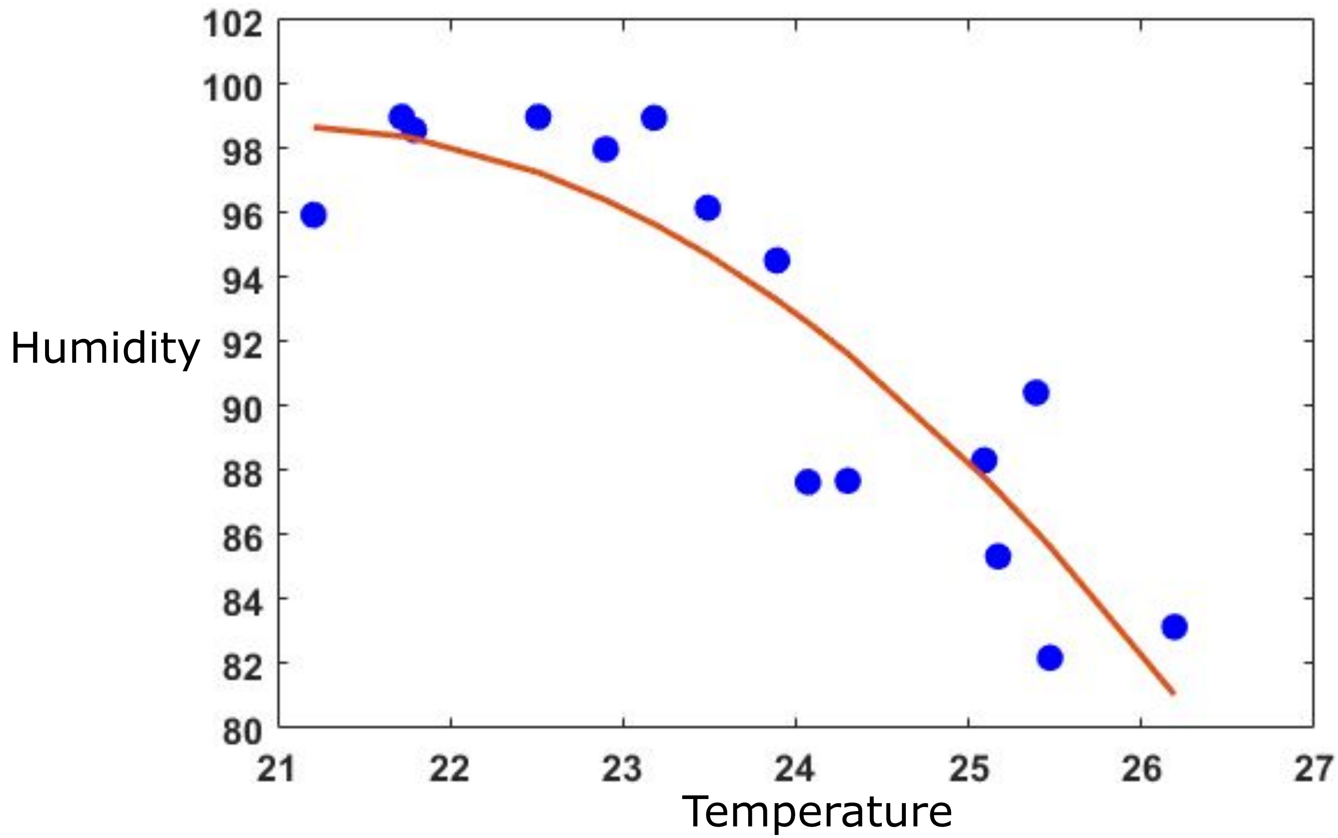
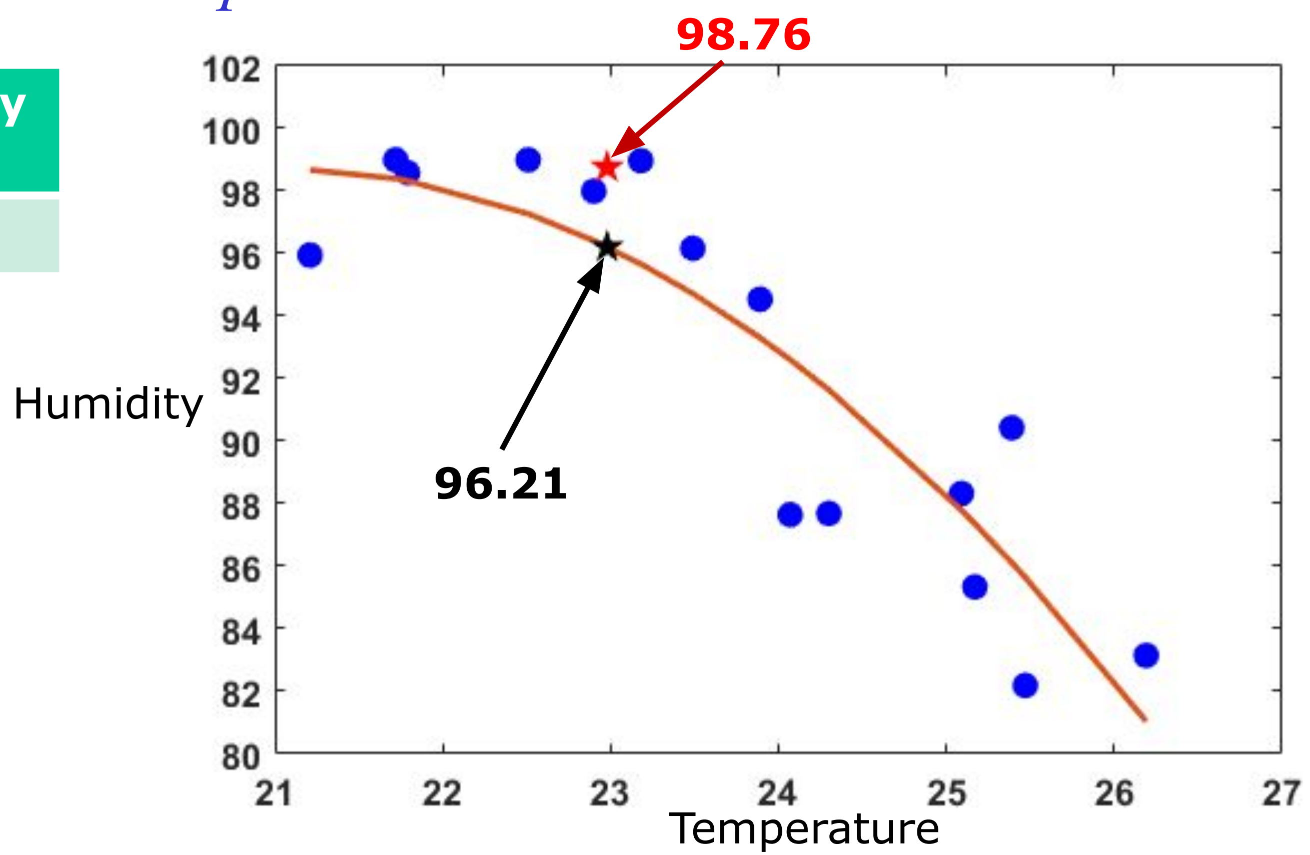


Illustration of Polynomial Curve Fitting: Humidity Prediction - Test

- Degree of polynomial p : 2

Temp (x)	Humidity (y)
22.98	--



- Predicted humidity: 96.21
- Actual humidity: 98.76
- Squared error: **06.49**

Illustration of Polynomial Curve Fitting: Humidity Prediction - Training

Temp (x)	Humidity (y)
25.47	82.19
26.19	83.15
25.17	85.34
24.30	87.69
24.07	87.65
21.21	95.95
23.49	96.17
21.79	98.59
25.09	88.33
25.39	90.43
23.89	94.54
22.51	99.00
22.90	98.00
21.72	99.00
23.18	98.97

- Degree of polynomial p : 3

$$\hat{\mathbf{w}} = \left(\mathbf{Z}^\top \mathbf{Z}\right)^{-1} \mathbf{Z}^\top \mathbf{y}$$

$\mathbf{Z}$ is 15 x 4 matrix

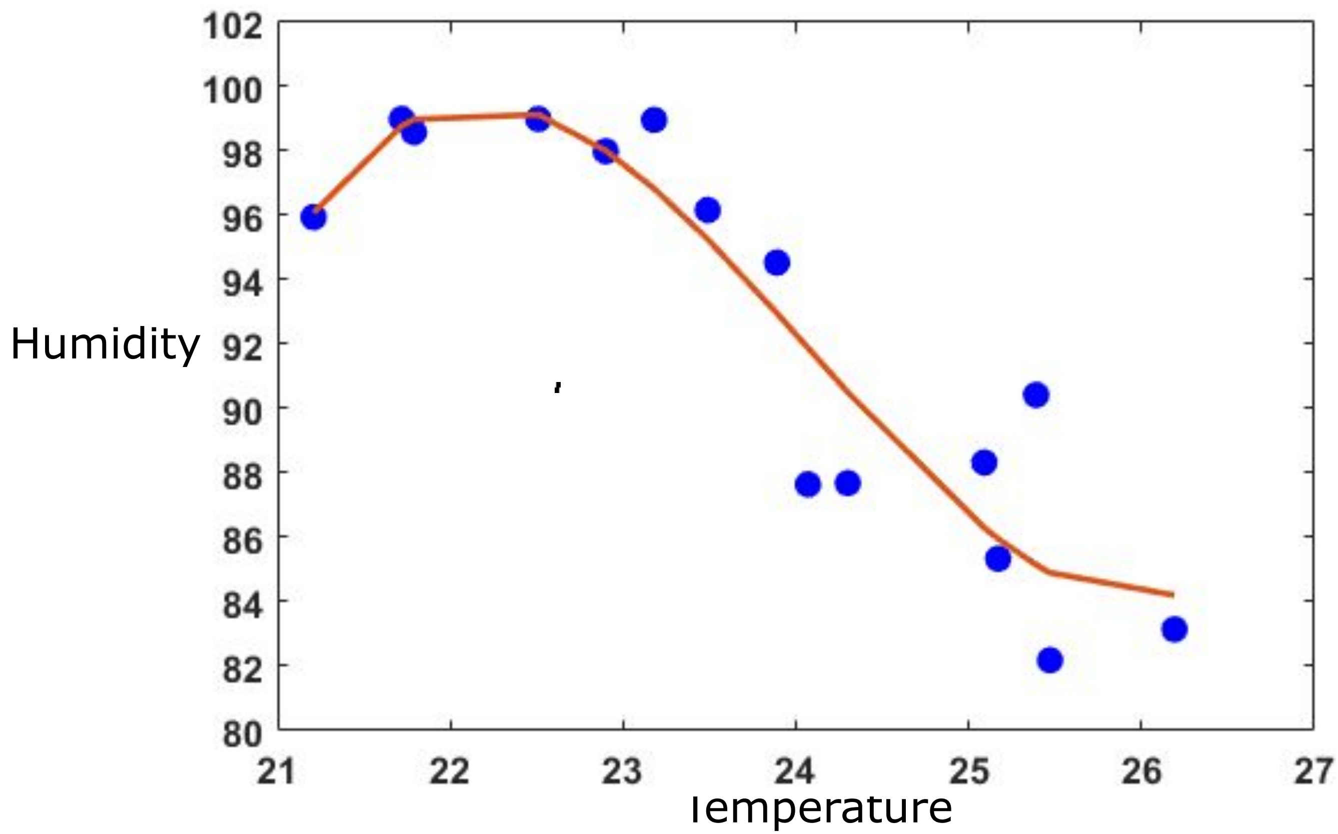
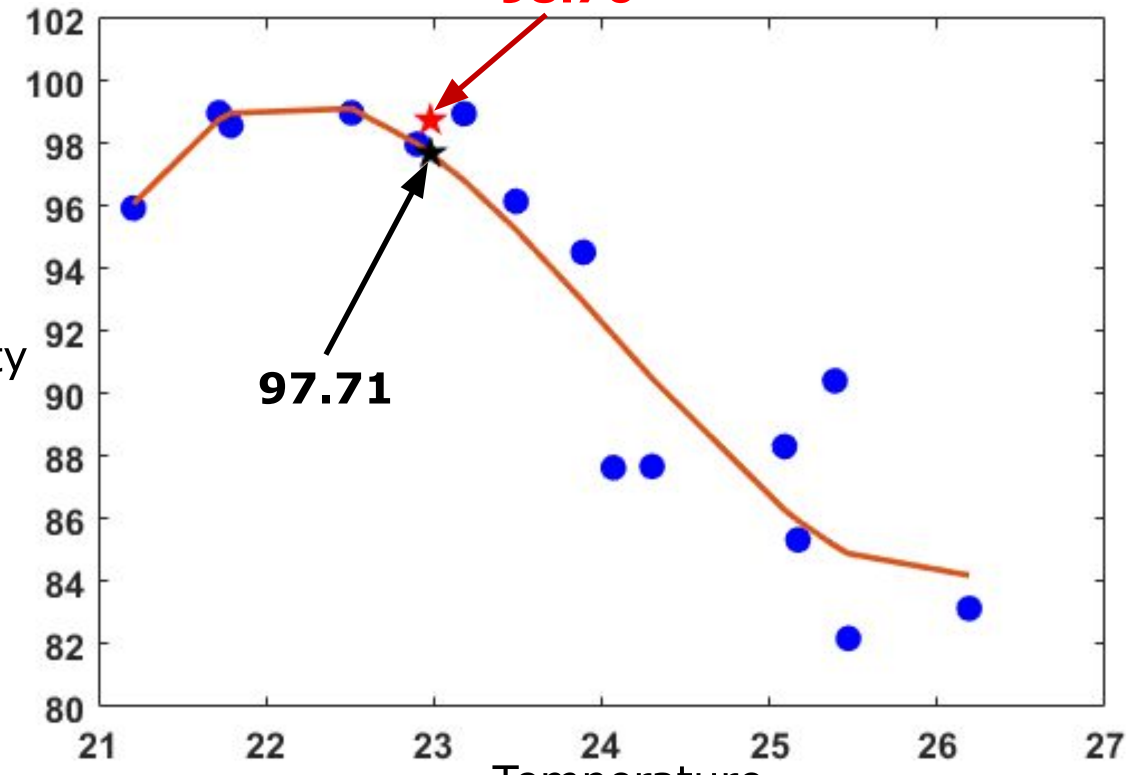


Illustration of Polynomial Curve Fitting: Humidity Prediction - Test

- Degree of polynomial p : 3

Temp (x)	Humidity (y)
22.98	--

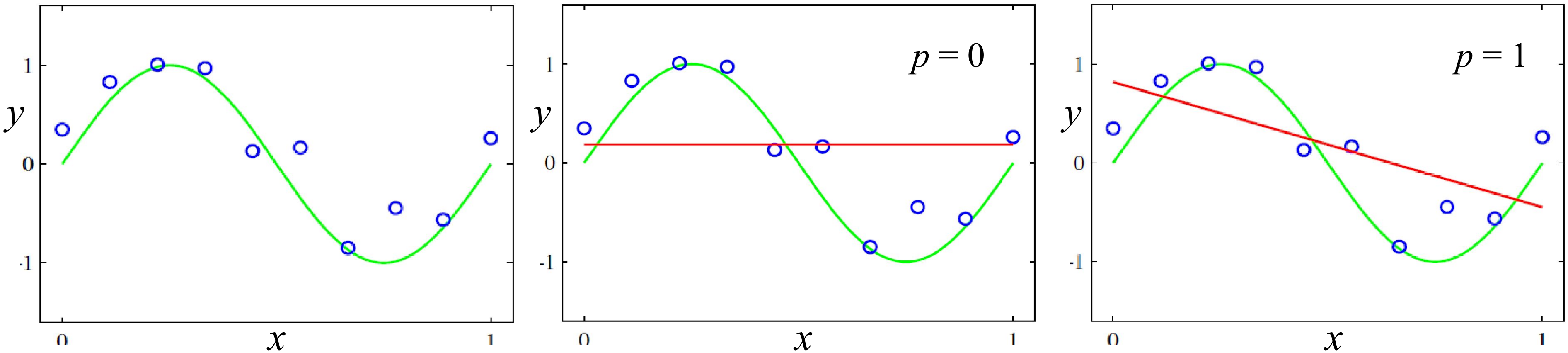


- Predicted humidity: 97.71
- Actual humidity: 98.76
- Squared error: **01.11**

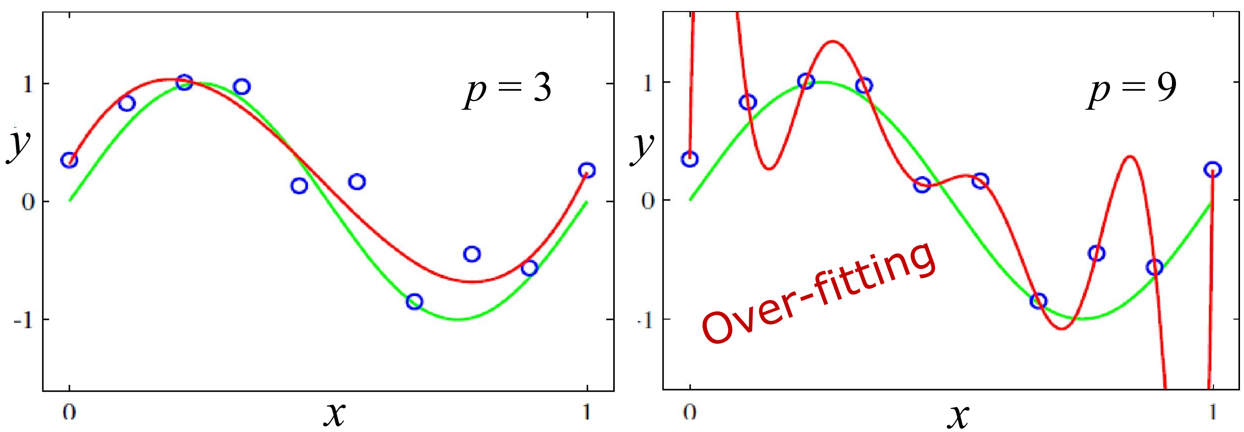
$p < N$

$p > N$

Illustration: Polynomial Curve Fitting



- Condition: Less number of training examples (N=10)
- Effect of increasing the degree of polynomial (p)



C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

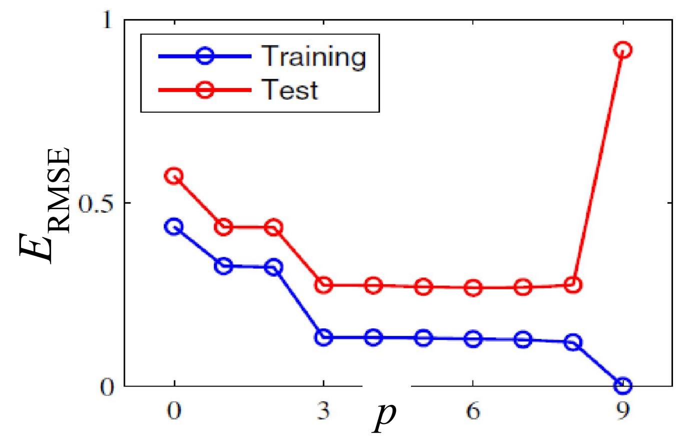
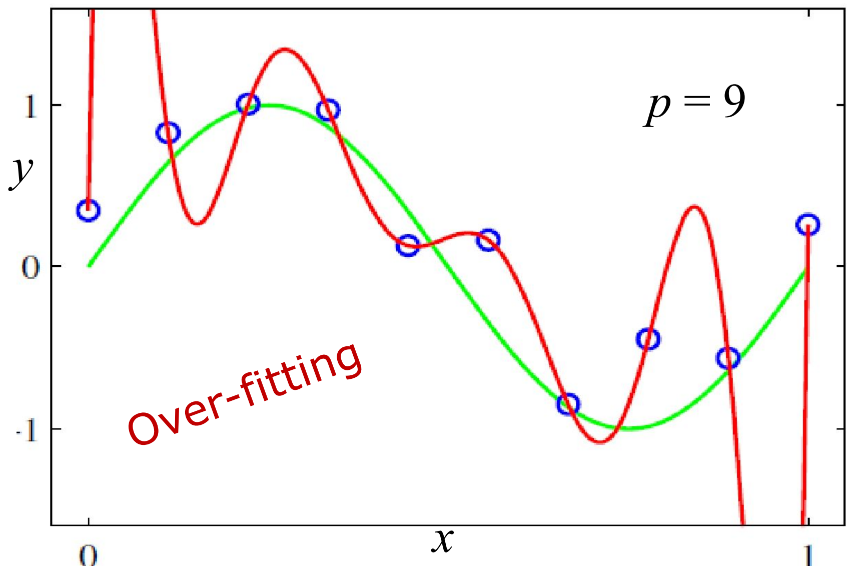
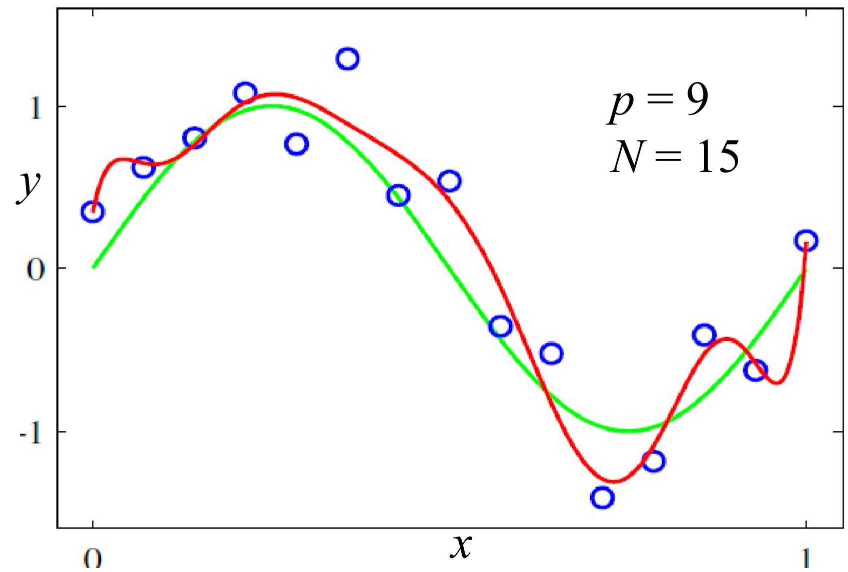
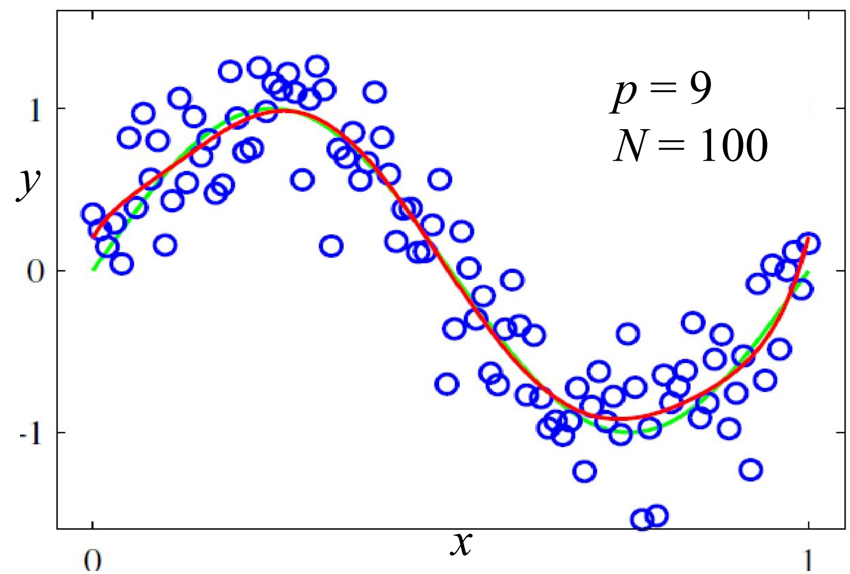


Illustration: Polynomial Curve Fitting





- Increasing the size of the data set reduces the over-fitting problem

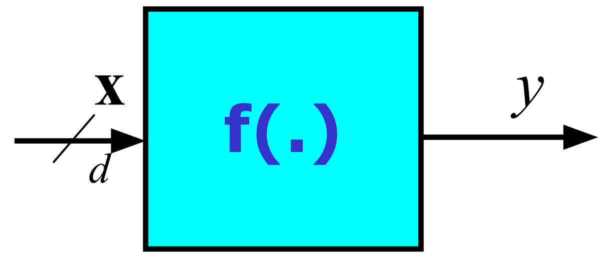


C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Supervised Machine Learning: Regression

Polynomial Regression

Nonlinear Regression: Polynomial Regression



- Polynomial regression:
  - Two or more independent variable ($\mathbf{x}$)
  - Single dependent variable (y)
- Given:- Training data: $\mathrm{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathrm{R}^d$ and $y_n \in \mathrm{R}^1$
- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

  - m is the number of monomials of polynomial up to degree p
  - $\varphi_j(\mathbf{x}_n)$ is the jth monomial of degree p for $\mathbf{x}_n$
- For 2-dimensional input, $\mathbf{x}_n = [x_{n1}, x_{n2}]^{\mathsf{T}}$ and degree, $p=2$

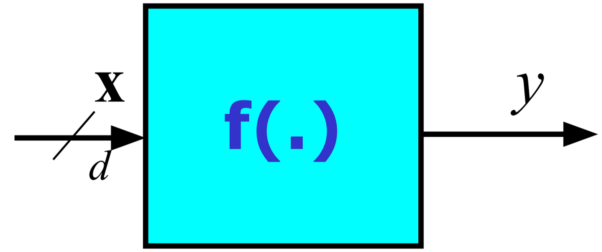
$$\boldsymbol{\varphi}(\mathbf{x}_n) = [\varphi_0(\mathbf{x}_n),\ \varphi_1(\mathbf{x}_n),\ \varphi_2(\mathbf{x}_n),\ \varphi_3(\mathbf{x}_n),\ \varphi_4(\mathbf{x}_n),\ \varphi_5(\mathbf{x}_n)]^{\mathsf{T}}$$

$$\boldsymbol{\varphi}(\mathbf{x}_n) = [\ 1,\ x_{n1},\ x_{n2},\ x_{n1}^2,\ x_{n2}^2,\ x_{n1}x_{n2}\ ]^{\mathsf{T}}$$

$m = 6$

Nonlinear Regression: Polynomial Regression

- Polynomial regression:
  - Two or more independent variable ($\mathbf{x}$)
  - Single dependent variable (y)



- Given:- Training data: D $= \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathrm{R}^d$ and $y_n \in \mathrm{R}^1$
- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

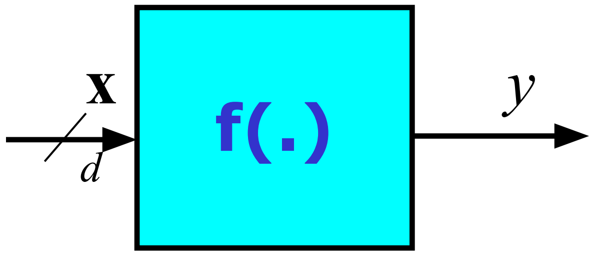
  - m is the number of monomials of polynomial up to degree p
  - $\varphi_j(\mathbf{x}_n)$ is the jth monomial of degree p for $\mathbf{x}_n$
- For 2-dimensional input, $\mathbf{x}_n=[x_{n1}, x_{n2}]^{\mathsf{T}}$ and degree, $p=2$

$$y_n = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = w_0 + w_1 x_{n1} + w_2 x_{n2} + w_3 x_{n1}^2 + w_4 x_{n2}^2 + w_5 x_{n1} x_{n2}$$

$$y_n = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = w_0\varphi_0(\mathbf{x}_n) + w_1\varphi_1(\mathbf{x}_n) + w_2\varphi_2(\mathbf{x}_n) + w_3\varphi_3(\mathbf{x}_n) + w_4\varphi_4(\mathbf{x}_n) + w_5\varphi_5(\mathbf{x}_n)$$

Nonlinear Regression: Polynomial Regression

- Polynomial regression:
  - Two or more independent variable ($\mathbf{x}$)
  - Single dependent variable (y)



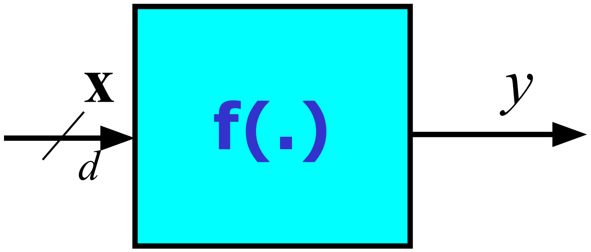
- Given:- Training data: D $= \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathrm{R}^d$ and $y_n \in \mathrm{R}^1$
- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

  - m is the number of monomials of polynomial up to degree p
  - $\varphi_j(\mathbf{x}_n)$ is the jth monomial of degree p for $\mathbf{x}_n$
- For 2-dimensional input, $\mathbf{x}=[x_1, x_2]^{\mathsf{T}}$ and degree, $p=3$

Nonlinear Regression: Polynomial Regression

- Polynomial regression:
  - Two or more independent variable ($\mathbf{x}$)
  - Single dependent variable (y)



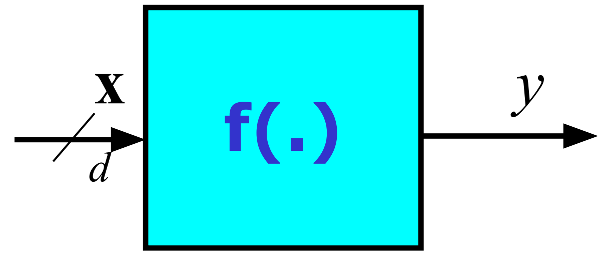
- Given:- Training data: D $= \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \mathbb{R}^1$
- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

  - m is the number of monomials of polynomial up to degree p
  - $\varphi_j(\mathbf{x}_n)$ is the jth monomial of degree p for $\mathbf{x}_n$
- For 3-dimensional input, $\mathbf{x}=[x_1, x_2, x_3]^{\mathsf{T}}$ and degree, $p=2$

Nonlinear Regression: Polynomial Regression

- Polynomial regression:
  - One or more independent variable ($\mathbf{x}$)
  - Single dependent variable (y)



- Given:- Training data: D $= \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, $\mathbf{x}_n \in$ R^d and $y_n \in$ R^1
- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

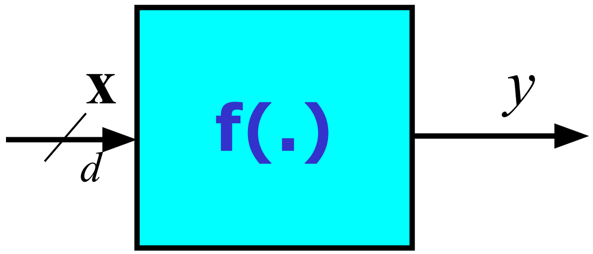
  - m is the number of monomials of polynomial up to degree p
  - $\varphi_j(\mathbf{x}_n)$ is the jth monomial of degree p for $\mathbf{x}_n$

The number of monomials m for the polynomial of degree p and the dimension of d is given by

$$m = \frac{(d+p)!}{d!\,p!}$$

Nonlinear Regression: Polynomial Regression

- Polynomial regression:
  - Two or more independent variable ($\mathbf{x}$)
  - Single dependent variable (y)



- Given:- Training data: D $= \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathrm{R}^d$ and $y_n \in \mathrm{R}^1$
- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

  - m is the number of monomials of polynomial up to degree p
  - $\varphi_j(\mathbf{x}_n)$ is the jth monomial of degree p for $\mathbf{x}_n$

$$m = \frac{(d+p)!}{d!\,p!}$$

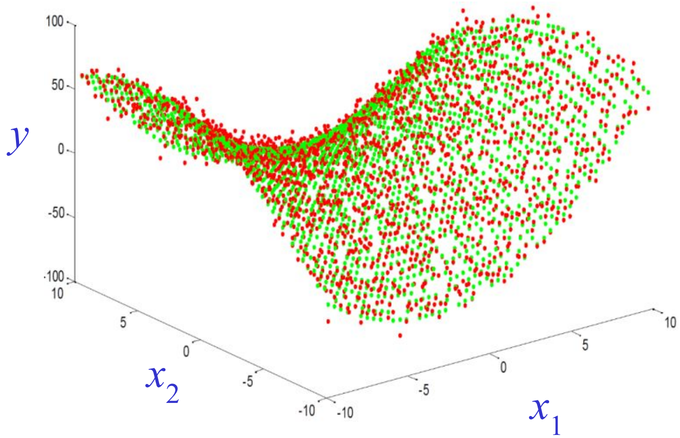
Example: Let the dimension of input variable is $d=6$ and the polynomial of degree $p=3$

- The number of monomials $m = \mathbf{84}$

Nonlinear Regression: Polynomial Regression

- Given:- Training data: $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, $\mathbf{x}_n \in R^d$ and $y_n \in R^1$
- Function governing the relationship between input and output given by a polynomial function of degree p:

$$y_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$



$y = f(\mathbf{x}_n, \mathbf{w})$

$\mathbf{x} = [x_1, x_2]^{\mathsf{T}}$

Fitting a surface

- The coefficients $\mathbf{w}=[w_0, w_1, \ldots, w_{m-1}]$ are parameters of surface (polynomial function) (regression coefficients) ***- Unknown***

- Polynomial function $f(\mathbf{x}_n,\mathbf{w})$ is a nonlinear function of $\mathbf{x}_n$ and
- Function $f(\mathbf{x}_n,\mathbf{w})$ is a linear function of coefficients $\mathbf{w}$
  – ***Linear model for regression***

Polynomial Regression: Training Phase

- The values for the coefficients will be determined by fitting the linear function to the training data
- Given:- Training data: $\mathrm{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, $\mathbf{x}_n \in \mathrm{R}^d$ and $y_n \in \mathrm{R}^1$
- **Method of least squares**: Minimizes the sum of the squared error between
  - all the actual data (y_n) i.e. actual dependent variable and
  - the estimate of line (predicted dependent variable ($\hat{y}_n$)) i.e. the function $f(x_n, \mathbf{w})$, in the training set for any given value of $\mathbf{w}$

$$\hat{y}_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

$$\underset{\mathbf{w}}{\text{minimize}}\, E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(\hat{y}_n - y_n\right)^2$$

- Minimize the error such that the coefficients $\mathbf{w}$ represent the parameter of polynomial curve that best fit the training data

Polynomial Regression: Training Phase

- The values for the coefficients will be determined by fitting the linear function to the training data
- Given:- Training data: $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, $\mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \mathbb{R}^1$
- **Method of least squares**: Minimizes the sum of the squared error between
  - all the actual data (y_n) i.e. actual dependent variable and
  - the estimate of line (predicted dependent variable ($\hat{y}_n$)) i.e. the function $f(x_n, \mathbf{w})$, in the training set for any given value of $\mathbf{w}$

$$\hat{y}_n = f(\mathbf{x}_n, \mathbf{w}) = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

$$\underset{\mathbf{w}}{\text{minimize}}\, E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(\hat{y}_n - y_n\right)^2$$

- The error function is a
  - quadratic function of the coefficients $\mathbf{w}$ and
  - The derivatives of error function with respect to the coefficients will be linear in the elements of $\mathbf{w}$
- Hence the minimization of the error function has unique solution and found in closed form

Polynomial Regression : Training Phase

$$\hat{y}_n = f(\mathbf{x}_n, \mathbf{w})$$

$$\hat{y}_n = f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w})$$

$$\hat{y}_n = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n)$$

$$\hat{y}_n = \mathbf{w}^{\mathsf{T}} \boldsymbol{\varphi}(\mathbf{x}_n)$$

where $\mathbf{w} = [w_0, w_1, \ldots, w_{m-1}]^{\mathsf{T}}$ and

$$\boldsymbol{\varphi}(\mathbf{x}_n) = [\varphi_0(\mathbf{x}_n), \varphi_1(\mathbf{x}_n), \varphi_2(\mathbf{x}_n), \ldots, \varphi_{m-1}(\mathbf{x}_n)]^{\mathsf{T}}$$

Polynomial Regression : Training Phase

- Cost function for optimization:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) - y_n\right)^2$$

- Conditions for optimality: $\dfrac{\partial\, E(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{0}$

- Application of optimality conditions gives optimal $\hat{\mathbf{w}}$:

$$\frac{\partial\, \frac{1}{2}\sum_{n=1}^{N}\left(\sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x}_n) - y_n\right)^2}{\partial \mathbf{w}} = \mathbf{0}$$

$$\frac{\partial\, \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{w}^{\mathsf{T}}\boldsymbol{\varphi}(\mathbf{x}_n) - y_n\right)^2}{\partial \mathbf{w}} = \mathbf{0}$$

Polynomial Regression : Training Phase

- Cost function for optimization:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(f(\boldsymbol{\varphi}(\mathbf{x}_n), \mathbf{w}) - y_n\right)^2$$

- Conditions for optimality: $\dfrac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = 0$

- Application of optimality conditions gives optimal $\hat{\mathbf{w}}$:

$$\frac{\partial \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{w}^{\top}\boldsymbol{\varphi}(\mathbf{x}_n) - y_n\right)^2}{\partial \mathbf{w}} = \mathbf{0}$$

$$\boxed{\hat{\mathbf{w}} = \left(\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\top}\mathbf{y}}$$

  – Assumption: $m < N$

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi_0(\mathbf{x}_1) & \varphi_1(\mathbf{x}_1) & \ldots & \varphi_{m-1}(\mathbf{x}_1) \\ \varphi_0(\mathbf{x}_2) & \varphi_1(\mathbf{x}_2) & \ldots & \varphi_{m-1}(\mathbf{x}_2) \\ - & - & - & - \\ \varphi_0(\mathbf{x}_n) & \varphi_1(\mathbf{x}_n) & \ldots & \varphi_{m-1}(\mathbf{x}_n) \\ - & - & - & - \\ \varphi_0(\mathbf{x}_N) & \varphi_1(\mathbf{x}_N) & \ldots & \varphi_{m-1}(\mathbf{x}_N) \end{bmatrix}$$

Polynomial Regression: Testing

- Optimal coefficient vector $\mathbf{w}$ is given by

$$\hat{\mathbf{w}} = \left(\mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}$$

$$\hat{\mathbf{w}} = \mathbf{\Phi}^{+} \mathbf{y}$$

where $\mathbf{\Phi}^{+} = \left(\mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top$ is the pseudo inverse of matrix $\mathbf{\Phi}$

- For any test example $\mathbf{x}$, the predicted value is given by:

$$\hat{y} = f(\mathbf{x}, \hat{\mathbf{w}}) = \hat{\mathbf{w}}^\top \boldsymbol{\varphi}(\mathbf{x}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x})$$

- The prediction accuracy is measured in terms of squared error: $E = \left(\hat{y} - y\right)^2$
- Let N_t be the total number of test samples
- The prediction accuracy of regression model is measured in terms of root mean squared error:

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{N_t} \sum_{n=1}^{N_t} \left(\hat{y}_n - y_n\right)^2}$$

Determining p, Degree of Polynomial

- This is determined experimentally
- Starting with p=1, test set is used to estimate the accuracy, in terms of error, of the regression model
  - Note: The polynomial degree p=1 is equivalent to multiple linear regression
- This process is repeated each time by incrementing p
- The regression model with p that gives the minimum error on test set may be selected

Illustration of Polynomial Regression: Temperature Prediction

Humidity (x_1)	Pressure (x_2)	Temp (y)
82.19	1036.35	25.47
83.15	1037.60	26.19
85.34	1037.89	25.17
87.69	1036.86	24.30
87.65	1027.83	24.07
95.95	1006.92	21.21
96.17	1006.57	23.49
98.59	1009.42	21.79
88.33	991.65	25.09
90.43	1009.66	25.39
94.54	1009.27	23.89
99.00	1009.80	22.51
98.00	1009.90	22.90
99.00	996.29	21.72
98.97	800.00	23.18

$d=2$

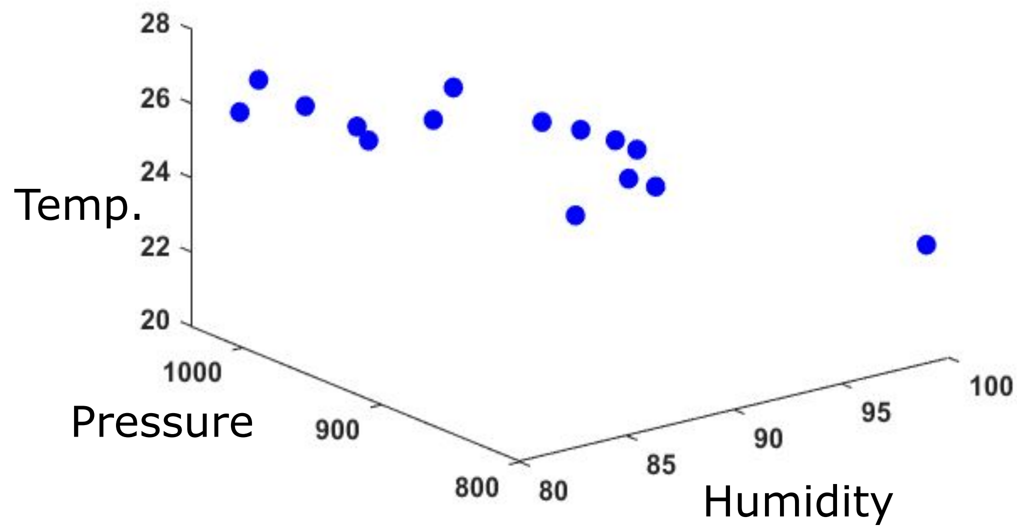


Illustration of Polynomial Regression: Temperature Prediction

Humidity (x_1)	Pressure (x_2)	Temp (y)
82.19	1036.35	25.47
83.15	1037.60	26.19
85.34	1037.89	25.17
87.69	1036.86	24.30
87.65	1027.83	24.07
95.95	1006.92	21.21
96.17	1006.57	23.49
98.59	1009.42	21.79
88.33	991.65	25.09
90.43	1009.66	25.39
94.54	1009.27	23.89
99.00	1009.80	22.51
98.00	1009.90	22.90
99.00	996.29	21.72
98.97	800.00	23.18

- **Training**:
- Polynomial Degree $p = 3$

$$\hat{\mathbf{w}} = \left(\mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}$$

$\mathbf{\Phi}$ is 15 x 10 matrix

$$\text{Number of monomials, } m = \frac{(d+p)!}{d!\,p!} = \frac{(2+3)!}{2! * 3!} = 10$$

Illustration of Polynomial Regression: Temperature Prediction

Humidity (x_1)	Pressure (x_2)	Temp (y)
82.19	1036.35	25.47
83.15	1037.60	26.19
85.34	1037.89	25.17
87.69	1036.86	24.30
87.65	1027.83	24.07
95.95	1006.92	21.21
96.17	1006.57	23.49
98.59	1009.42	21.79
88.33	991.65	25.09
90.43	1009.66	25.39
94.54	1009.27	23.89
99.00	1009.80	22.51
98.00	1009.90	22.90
99.00	996.29	21.72
98.97	800.00	23.18

- **Training**:
- Polynomial Degree $p = 3$

$$\hat{\mathbf{w}} = \left(\mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}$$

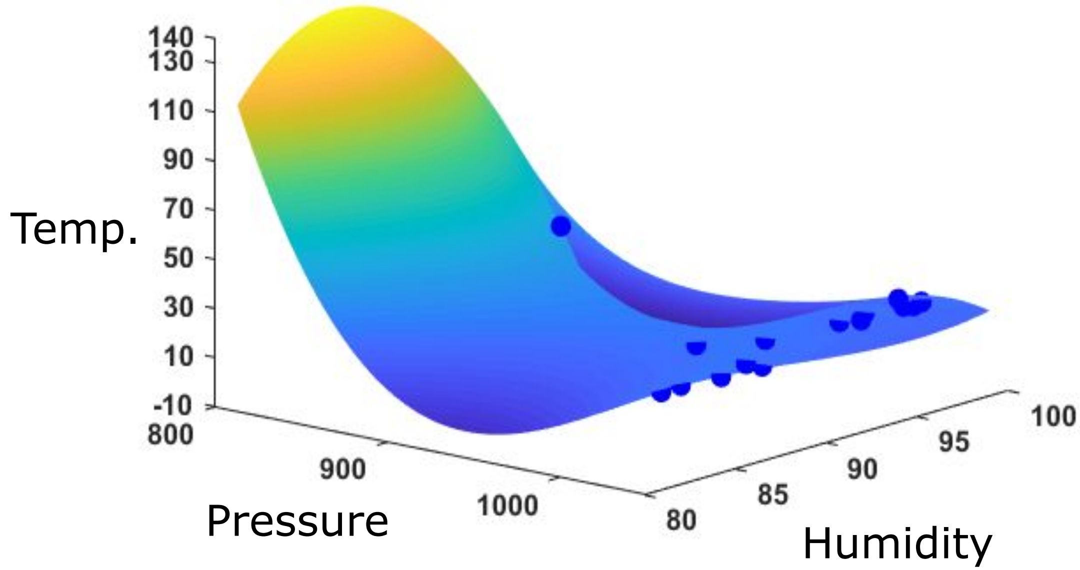


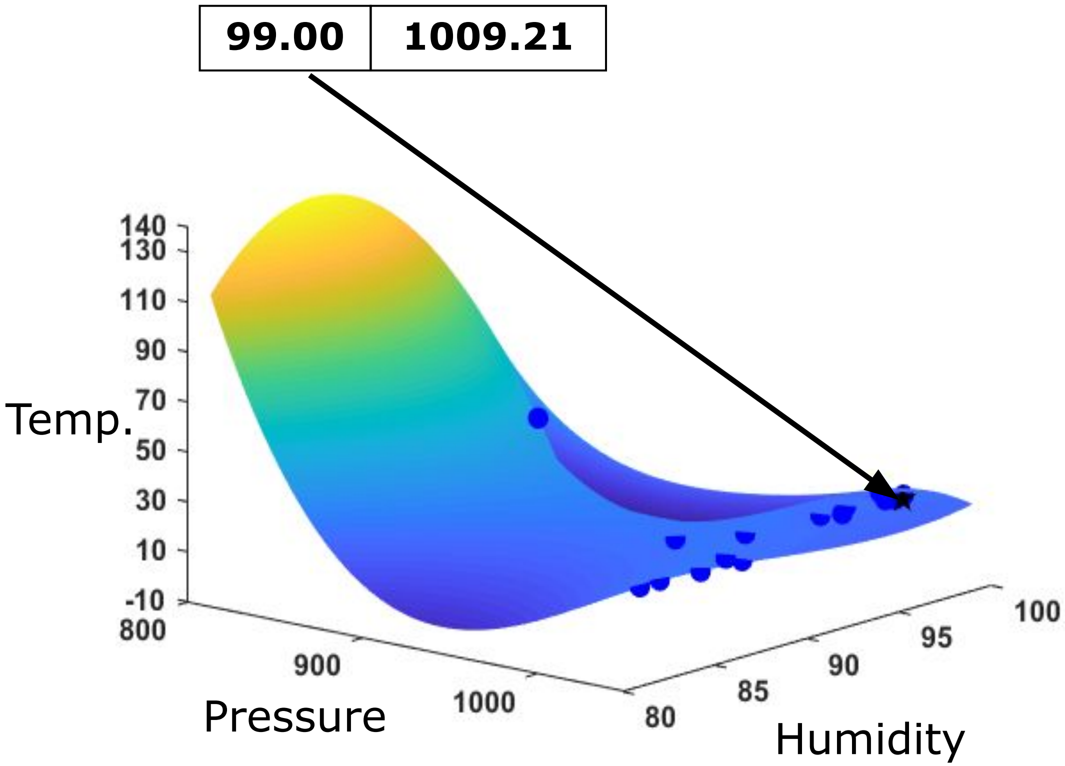
Illustration of Polynomial Regression: Temperature Prediction - Test

- Degree of polynomial $p = 3$

$$\hat{\mathbf{w}} = \left(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathsf{T}}\mathbf{y}$$

Humidity (x_1)	Pressure (x_2)	Temp (y)
99.00	1009.21	-

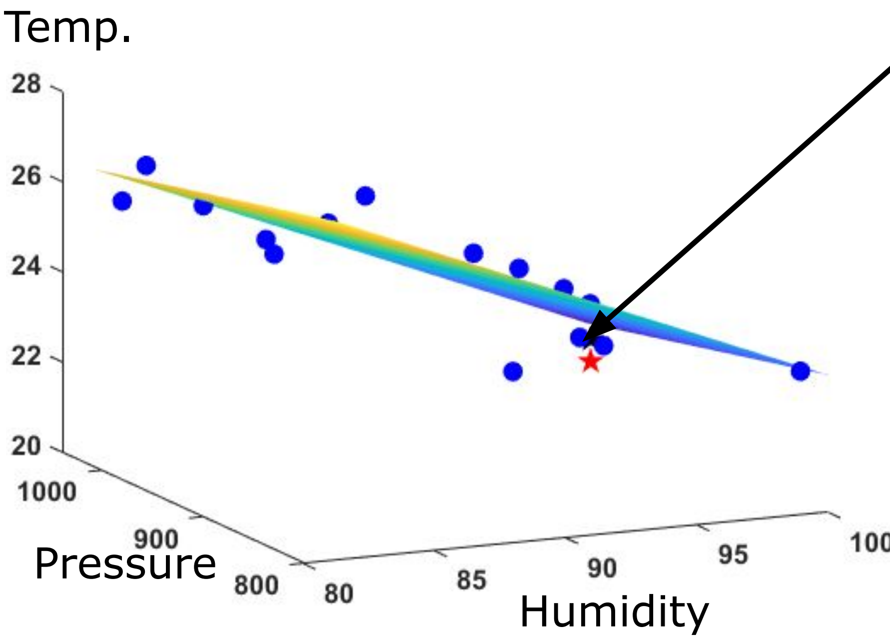
$$\hat{y} = f(\mathbf{x}, \hat{\mathbf{w}}) = \hat{\mathbf{w}}^{\mathsf{T}}\boldsymbol{\varphi}(\mathbf{x}) = \sum_{j=0}^{m-1} w_j \varphi_j(\mathbf{x})$$



- Predicted Temperature: 21.05
- Actual Temperature: 21.24
- Squared error: **0.035**

Multiple Linear Regression vs Polynomial Regression Temperature Prediction

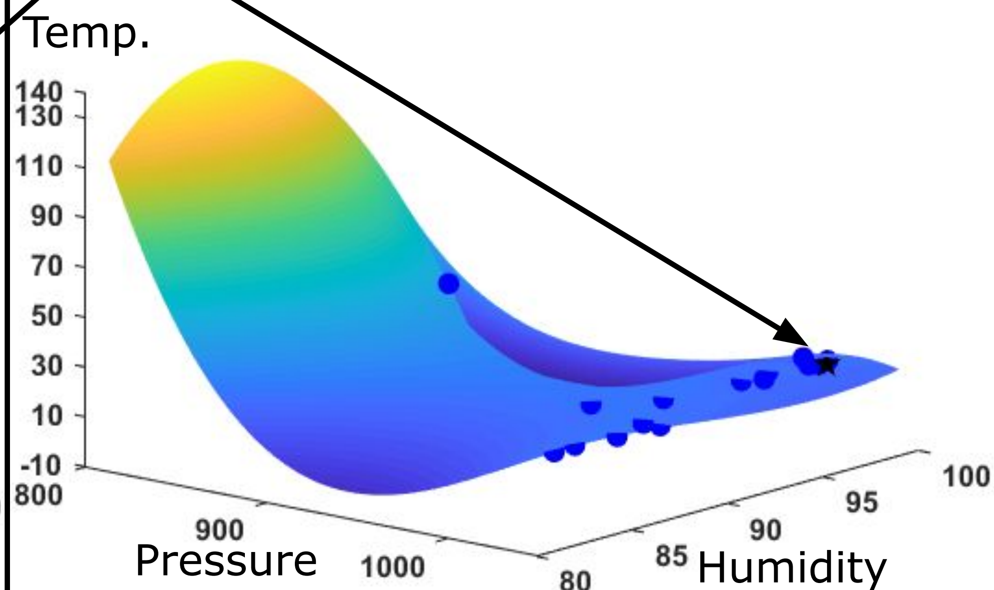
- Multiple Linear Regression



- Predicted Temperature: 21.72
- Actual Temperature: 21.24
- Squared error: **0.2347**

- Polynomial Regression
  - Degree of polynomial $p = 3$

99.00	**1009.21**



- Predicted Temperature: 21.05
- Actual Temperature: 21.24
- Squared error: **0.035**

Summary: Regression

- Regression analysis is used to model the relationship between one or more independent (predictor) variable and a dependent (response) variable
- Response is some function of one or more input variables
- Linear regression: Response is linear function of one or more input variables
  - If the response is linear function of one input variable, then it is simple linear regression (straight-line fitting)
  - If the response is linear function of two or more input variable, then it is multiple linear regression (linear surface fitting or hyperplane fitting)
- Nonlinear regression: Response is nonlinear function of one or more input variables
  - Polynomial regression: Response is nonlinear function approximated using polynomial function up to degree p of one or more input variables
  - When the degree of polynomial (p) is 1, then it is linear regression

Text Books

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.

2. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.