LINEAR REGRESSION FOR NONLINEAR PROBLEMS

CS6140

Predrag Radivojac
KHOURY COLLEGE OF COMPUTER SCIENCES
NORTHEASTERN UNIVERSITY

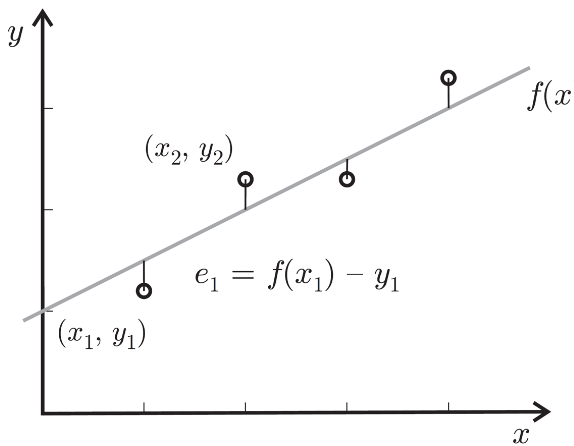
Fall 2024

LINEAR REGRESSION

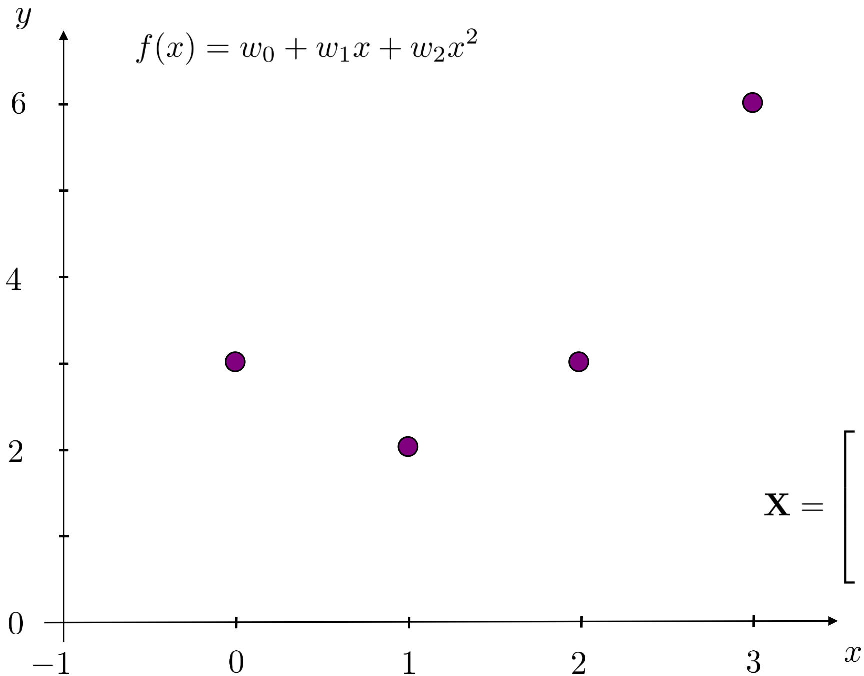
Given: a set of observations $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$

Objective: find best linear approximator $f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{d} w_j x_j$

$\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \mathbb{R}$

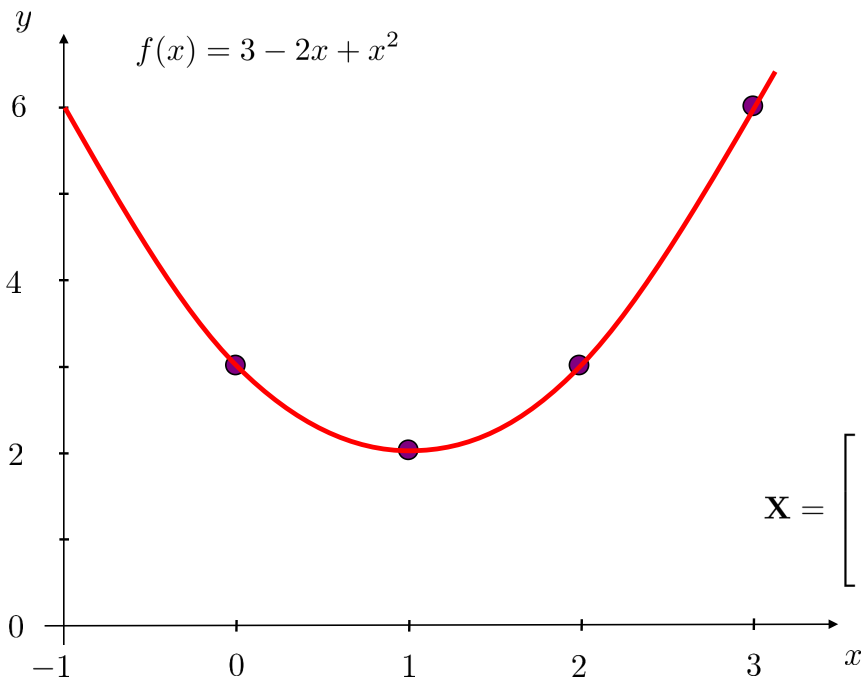


Learning Polynomial Functions



$$
\begin{array}{cc}
 & \begin{array}{ccc} \color{red}{1} & \color{red}{x} & \color{red}{x^2} \\ \color{red}{\downarrow} & \color{red}{\downarrow} & \color{red}{\downarrow} \end{array} \\
\mathbf{X} = & \begin{bmatrix} 1 & 0 & 0^2 \\ 1 & 1 & 1^2 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{bmatrix}
\end{array}
\qquad
\mathbf{y} = \begin{bmatrix} 3 \\ 2 \\ 3 \\ 6 \end{bmatrix}
$$

Learning Polynomial Functions



$$\begin{array}{c} \begin{array}{ccc} 1 & x & x^2 \\ \downarrow & \downarrow & \downarrow \end{array} \\ \mathbf{X} = \begin{bmatrix} 1 & 0 & 0^2 \\ 1 & 1 & 1^2 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{bmatrix} \end{array} \quad \mathbf{y} = \begin{bmatrix} 3 \\ 2 \\ 3 \\ 6 \end{bmatrix}$$

$$e_1^2 + e_2^2 + e_3^2 + e_4^2 = 0$$

Let's Reflect for a Moment

Q: Have we learned a non-linear function using linear regression?

A: Yup!

Q: How did it happen?

A: We mapped the data into a higher-dimension using a nonlinear transformation.

$$\mathbf{x} = (1, x) \xrightarrow{\varphi(\mathbf{x})} \cancel{\mathbf{x} = (1, x, x^2)}$$

$$\mathbf{x} = (1, x, x^2, \ldots, x^p)$$

$p =$ degree of the polynomial

Summary

Original data:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

Intermediate data:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

Still the same solution:

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

What if the Data is Multivariate?

$$\varphi(\mathbf{x})$$

$$\begin{array}{cc} & \\ \mathbf{x}_2^T & \\ & \\ \mathbf{X} = & \left[\begin{array}{ccccc} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & & x_{2d} \\ 1 & x_{31} & x_{32} & & x_{3d} \\ \vdots & & & \ddots & \vdots \\ 1 & x_{n1} & & & x_{nd} \end{array}\right] \end{array} \qquad \mathbf{\Phi} = \begin{bmatrix} 1 & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_p(\mathbf{x}_1) \\ 1 & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & & \phi_p(\mathbf{x}_2) \\ 1 & \phi_1(\mathbf{x}_3) & \phi_2(\mathbf{x}_3) & & \phi_p(\mathbf{x}_3) \\ \vdots & & & \ddots & \vdots \\ 1 & \phi_1(\mathbf{x}_n) & & & \phi_p(\mathbf{x}_n) \end{bmatrix}$$

$$\mathbf{w}^* = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{y}.$$

WHAT IF THE DATA IS MULTIVARIATE?

1. Cluster data into p clusters.
2. Pick p points $\mathbf{c}_1$ .. $\mathbf{c}_p$; e.g., examples or centers
3. Make a transformation for each input example

Centers: $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_p$

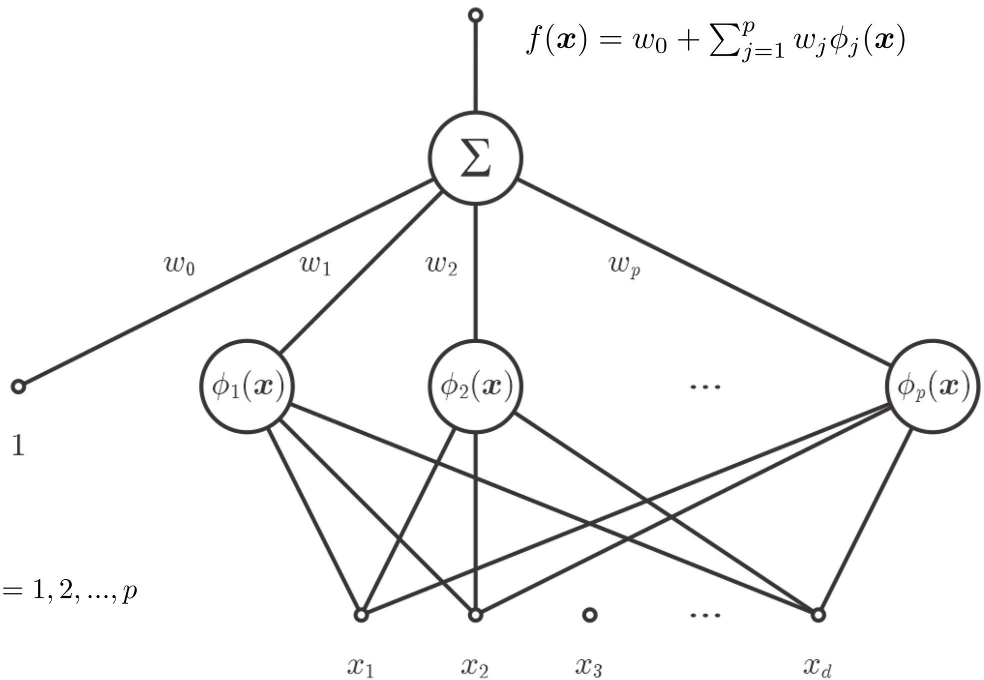
Parameters: $\sigma_1, \sigma_2, \ldots, \sigma_p$

$$\phi_j(\mathbf{x}) = e^{-\frac{(\mathbf{x}-\mathbf{c}_j)^T(\mathbf{x}-\mathbf{c}_j)}{2\sigma_j^2}}$$

Radial basis function

$$j = 1, 2, \ldots, p$$

RADIAL BASIS FUNCTION NETWORK



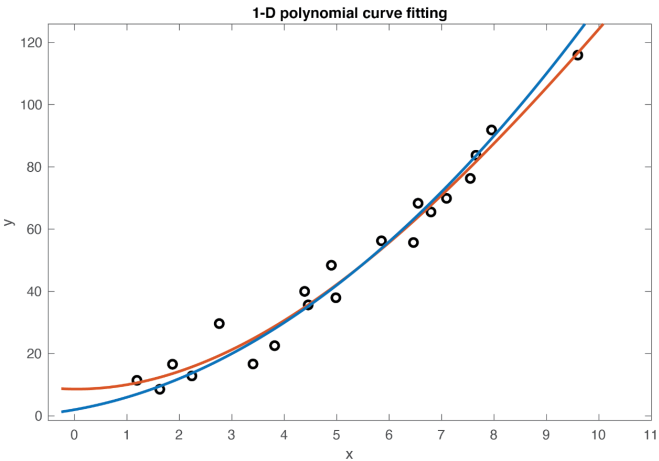
$$\phi_j(\boldsymbol{x}) = e^{-\frac{dist^2(\boldsymbol{x}, \boldsymbol{c}_j)}{2\sigma_j^2}}, \; j = 1, 2, ..., p$$

$$\boldsymbol{x} = (x_1, x_2, ..., x_d)$$

Example: Polynomial Curve Fitting

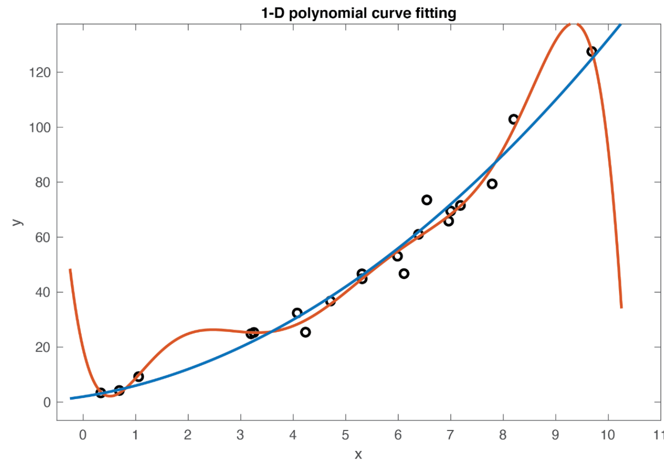
$y = 2 + 3x + x^2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 25)$

$p = 3$



$\hat{\boldsymbol{w}} = (13, -2.6, 2.0 - 0.1)$

$p = 7$



$\hat{\boldsymbol{w}} = (19.4, -80.9, 123.5, -70.3, 19.8, -2.9, 0.2, 0.0)$

Regularization

Idea: modify the objective function

$$\text{Objective} = \underbrace{\text{original objective}}_{\text{sum of squared errors}} + \underbrace{\text{regularization term}}_{\text{a function of } \boldsymbol{w}}$$

Ridge regression:

$\mathbf{w}^T\mathbf{w}$

$$\text{Objective} = \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i)^2 \; + \; \lambda \sum_{j=1}^{d} w_j^2$$

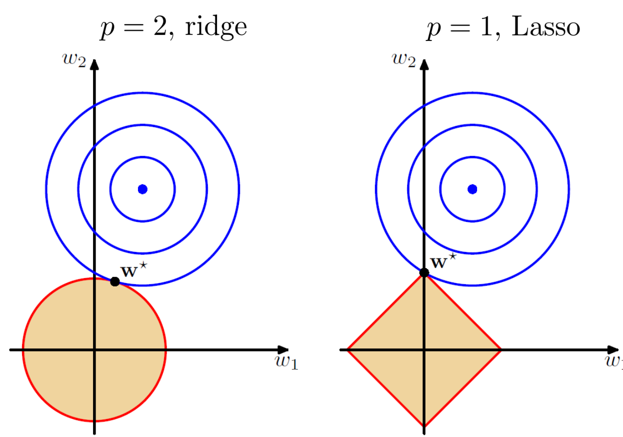
Solution:

$$\mathbf{w}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

Regularization

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{d} w_j x_{ij})^2 \right\}$$

$$\text{subject to } \textstyle\sum_{j=1}^{d} |w_j|^p \leq t$$



Picture from Bishop's textbook (Chapter 3).

MAP Estimation

$$p(\boldsymbol{w}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{w}) \cdot p(\boldsymbol{w})$$

$$p(\mathcal{D}|\boldsymbol{w}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}$$

likelihood

$$p(\boldsymbol{w}) = \frac{1}{(2\pi\alpha^2)^{(d+1)/2}} e^{-\frac{1}{2\alpha^2}\mathbf{w}^T\mathbf{w}}$$

prior

$$\mathbf{w}_{\mathrm{MAP}} = \arg\min_{\mathbf{w}} \left\{ (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{\sigma^2}{\alpha^2}\mathbf{w}^T\mathbf{w} \right\}$$

Performance of Regression Models

Given: training set $\mathcal{D}$, large test set $\mathcal{T}$, and model $f(x)$ trained on $\mathcal{D}$

Goal: estimate accuracy of $f(x)$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2$$

$$\begin{aligned} f(x) &= \text{prediction on } x \\ y &= \text{observed target for } x \\ n &= \text{number of examples in } \mathcal{T} \\ \bar{y} &= \text{mean of the target} \end{aligned}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2}{\sum_{i=1}^{n} \left(\bar{y} - y_i\right)^2}$$

$$\text{MSE} \in [0, \infty)$$

$$R^2 \in (-\infty, 1]$$

R^2 = precentage of variance "explained" by $f(x)$. Target mean is the trivial predictor.

Performance of Regression Models

Given: training set $\mathcal{D}$, large test set $\mathcal{T}$, and model $f(x)$ trained on $\mathcal{D}$

Goal: estimate accuracy of $f(x)$

Pearson's correlation:

$$\rho = \frac{\sum_{i=1}^{n}(f_i - \bar{f})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(f_i - \bar{f})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$\begin{aligned} f_i &= \text{prediction } f(x_i) \\ y_i &= \text{observed target for } x_i \\ n &= \text{number of examples in } \mathcal{T} \\ \bar{y} &= \text{mean of the target} \\ \bar{f} &= \text{mean of the predictions} \end{aligned}$$

Kendall's tau:

$$\tau_K = \frac{2}{n(n-1)} \sum_{i<j} \operatorname{sign}(f(x_i) - f(x_j)) \cdot \operatorname{sign}(y_i - y_j)$$

$$\rho \in [-1, 1]$$

$$\tau_K \in [-1, 1]$$