ECE595 / STAT598: Machine Learning I

Lecture 1.1: Linear Regression

Spring 2020

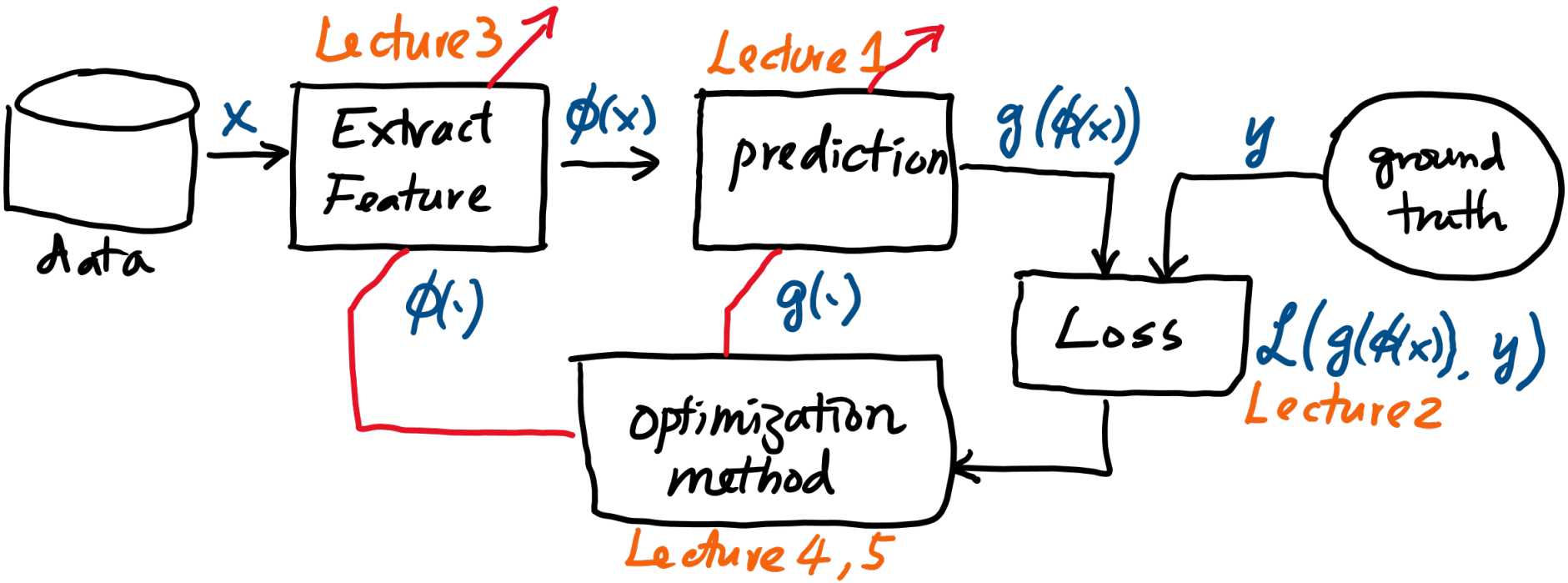
Stanley Chan

School of Electrical and Computer Engineering
Purdue University

PURDUE UNIVERSITY

© Stanley Chan 2020. All Rights Reserved.

Outline



©Stanley Chan 2020. All Rights Reserved.

Outline

Mathematical Background

- Lecture 1: Linear regression: A basic data analytic tool
- Lecture 2: Regularization: Constraining the solution
- Lecture 3: Kernel Method: Enabling nonlinearity

Lecture 1: Linear Regression

- Linear Regression
  - Notation
  - Loss Function
  - Solving the Regression Problem
- Geometry
  - Projection
  - Minimum-Norm Solution
  - Pseudo-Inverse

©Stanley Chan 2020. All Rights Reserved.

Basic Notation

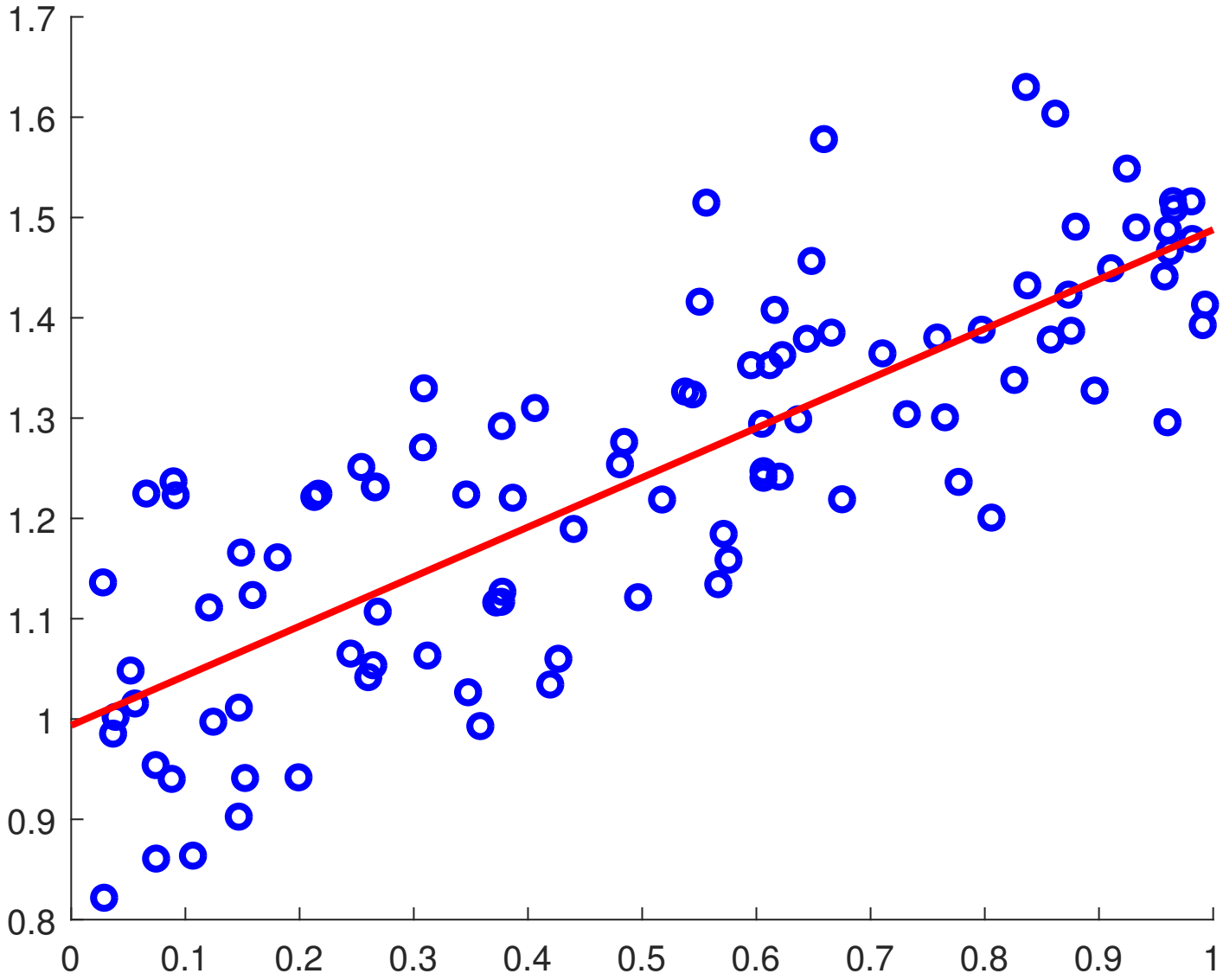
- Scalar: $a, b, c \in \mathbb{R}$
- Vector: $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^d$
- Matrix: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{N \times d}$; Entries are a_{ij} or $[\boldsymbol{A}]_{ij}$.
- Rows and Columns

$$\boldsymbol{A} = \begin{bmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_d \\ | & | & & | \end{bmatrix}, \quad \text{and} \quad \boldsymbol{A} = \begin{bmatrix} — & (\boldsymbol{x}^1)^T & — \\ — & (\boldsymbol{x}^2)^T & — \\ & \vdots & \\ — & (\boldsymbol{x}^N)^T & — \end{bmatrix}.$$

- $\{\boldsymbol{a}_j\}$: The j-th feature. $\{\boldsymbol{x}^n\}$: The n-th sample.
- Identity matrix $\boldsymbol{I}$
- All-one vector $\mathbf{1}$ and all-zero vector $\mathbf{0}$
- Standard basis $\boldsymbol{e}_i$.

©Stanley Chan 2020. All Rights Reserved.

Line Fitting



©Stanley Chan 2020. All Rights Reserved.

Linear Regression

The problem of **regression** can be summarized as:

- Given measurements: y^n, (where $n = 1, \ldots, N$)
- Given inputs: $\boldsymbol{x}^n$
- Given a model: $g_{\boldsymbol{\theta}}(\boldsymbol{x}^n)$ parameterized by $\boldsymbol{\theta}$
- Determine $\boldsymbol{\theta}$ such that $y^n \approx g_{\boldsymbol{\theta}}(\boldsymbol{x}^n)$.

Linear regression is one type of regression:

- Restrict $g_{\boldsymbol{\theta}}(\cdot)$ to a line:

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\theta}$$

- The inputs $\boldsymbol{x}$ and the parameters $\boldsymbol{\theta}$ are

$$\boldsymbol{x} = [x_1, \ldots, x_d]^T \qquad \text{and} \qquad \boldsymbol{\theta} = [\theta_1, \ldots, \theta_d]^T$$

- This is equivalent to

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\theta} = \sum_{j=1}^{d} x_j \theta_j.$$

©Stanley Chan 2020. All Rights Reserved.

Solving the Regression Problem

The (general) regression can be solved via the following logic:

- Define a (Squared-Error) *Loss Function*

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} \mathcal{L}(g_{\boldsymbol{\theta}}(\boldsymbol{x}^n), y^n)$$

$$= \sum_{n=1}^{N} (g_{\boldsymbol{\theta}}(\boldsymbol{x}^n) - y^n)^2, \qquad \text{e.g., } \mathcal{L}(\clubsuit, \spadesuit) \stackrel{\text{def}}{=} (\clubsuit - \spadesuit)^2$$

- Other loss functions can be used, e.g., $\mathcal{L}(\clubsuit, \spadesuit) = |\clubsuit - \spadesuit|$.
- The goal is to solve an optimization

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \;\; J(\boldsymbol{\theta}).$$

- The prediction of a new input $\boldsymbol{x}^{\text{new}}$ is $y^{\text{new}} = g_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{x}^{\text{new}}) = \widehat{\boldsymbol{\theta}}^T \boldsymbol{x}^{\text{new}}$.

©Stanley Chan 2020. All Rights Reserved.

Linear Regression Solution

The linear regression problem is a special case which we can solve analytically.

- Restrict $g_{\boldsymbol{\theta}}(\cdot)$ to a line:

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}^n) = \boldsymbol{\theta}^T \boldsymbol{x}^n$$

- Then the loss function becomes

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} (\boldsymbol{\theta}^T \boldsymbol{x}^n - y^n)^2 = \|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2.$$

- The matrix and vectors are defined as

$$\boldsymbol{A} = \begin{bmatrix} — & (\boldsymbol{x}^1)^T & — \\ — & (\boldsymbol{x}^2)^T & — \\ & \vdots & \\ — & (\boldsymbol{x}^N)^T & — \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}, \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

- $\|\cdot\|^2$ stands for the ℓ_2-norm square. See Tutorial on Linear Algebra.

©Stanley Chan 2020. All Rights Reserved.

Linear Regression Solution

Theorem

For a linear regression problem

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\, J(\boldsymbol{\theta}) \stackrel{def}{=} \|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2,$$

the minimizer is

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}.$$

- Take derivative and setting to zero: (See Tutorial on "Linear Algebra".)

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \left\{\|\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}\|^2\right\} \\ &= 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{y}) = \boldsymbol{0}.\end{aligned}$$

- So solution is $\widehat{\boldsymbol{\theta}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$, assuming $\boldsymbol{A}^T\boldsymbol{A}$ is invertible.

©Stanley Chan 2020. All Rights Reserved.

Examples

Example 1: Second-order polynomial fitting

$$y_n = ax_n^2 + bx_n + c$$

$$\boldsymbol{A} = \begin{bmatrix} x_1^2 & x_1 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

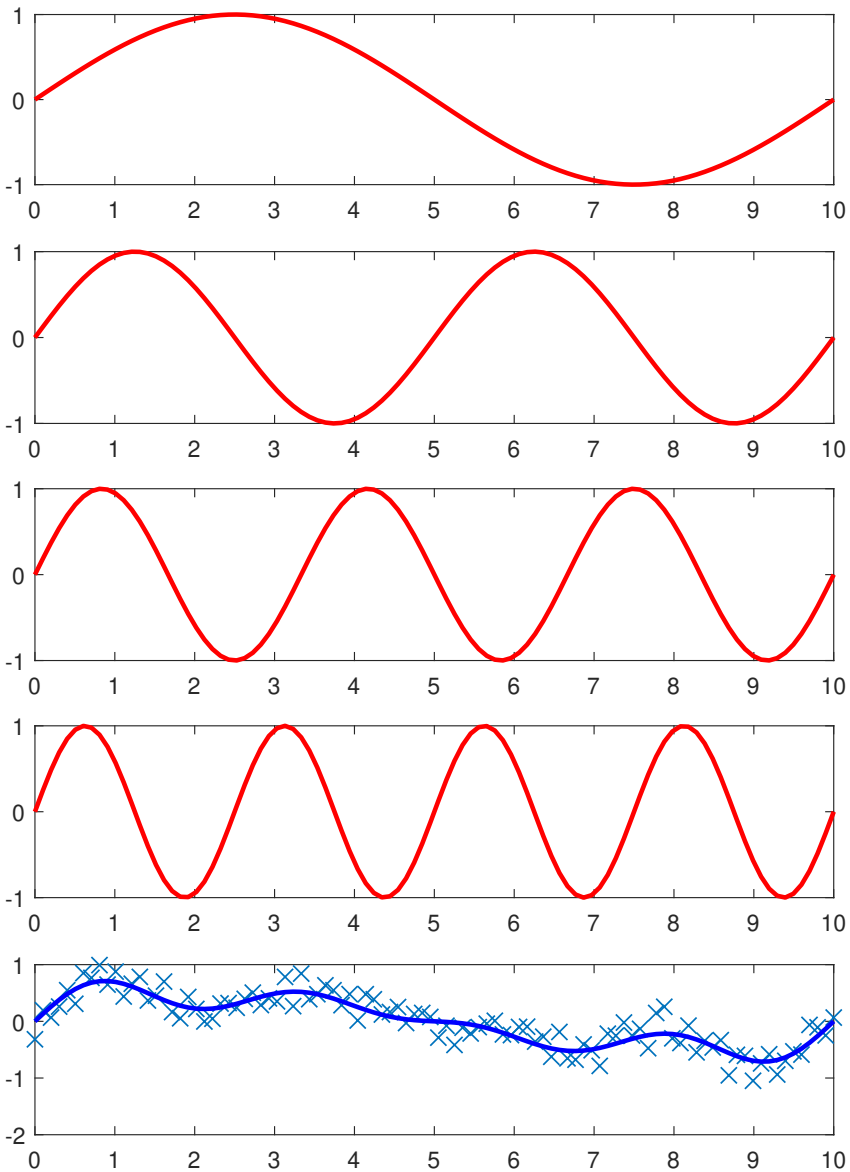
Example 2: Auto-regression

$$y_n = ay_{n-1} + by_{n-2}$$

$$\boldsymbol{A} = \begin{bmatrix} y_2 & y_1 \\ y_3 & y_2 \\ \vdots & \vdots \\ y_{N-1} & y_{N-2} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_3 \\ y_4 \\ \vdots \\ y_N \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} a \\ b \end{bmatrix}$$

©Stanley Chan 2020. All Rights Reserved.

Generalized Linear Regression



- Eg 1: Fourier series

$$\boldsymbol{x}^n = \begin{bmatrix} x_1^n \\ x_2^n \\ \vdots \\ x_d^n \end{bmatrix} = \begin{bmatrix} \sin(\omega_0 t_n) \\ \sin(2\omega_0 t_n) \\ \vdots \\ \sin(K\omega_0 t_n) \end{bmatrix}$$

$$y^n = \boldsymbol{\theta}^T \boldsymbol{x}^n = \sum_{k=1}^{d} \theta_k \sin(k\omega_0 t_n)$$

- θ_k: k-th Fourier coefficient
- $\sin(k\omega_0 t_n)$: k-th Fourier basis at time t_n

©Stanley Chan 2020. All Rights Reserved.