Linear Algebra & Convex Optimization – Lecture 7

Text : Introduction to Applied Linear Algebra, S. Boyd: Chapters 14,15.

Example: Advertising Purchase

- m demographic groups of audiences
- n number channels to advertise
- $m \times n$ Matrix R represents the 'Available Data' on Ad views per dollar spent
- v^{des} is the desired viewership from each region
- $m-$ vector $Rs = v$ gives the total viewership from each demographic group
- $n-$ vector s is the dollars invested in each channel for advertisement

Poll: What does $n-$ vector s represents ?
A) Total Views per channel
B) Dollars to be invested in each channel C) Total Views per region D) Dollars to be invested in each region

$$R = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 1200 & 1400 & \\ 2 & 800 & & \\ \vdots & & & \\ 10 & & & \end{array} \qquad b = \begin{bmatrix} 10^6 \\ 10^6 \\ \vdots \\ 10^6 \end{bmatrix}$$

$$\underset{m\times n}{R}\;\underset{n\times 1}{s} = \underset{m\times 1}{v}$$

$$R \begin{bmatrix} \text{Views}/\$ & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \$_1 \\ \$_2 \\ \$_3 \end{bmatrix} = \begin{bmatrix} \text{Views}_1 \\ \vdots \\ \text{Views}_{10} \end{bmatrix}$$

Example: Advertising Purchase

$n = 3$ channels

$m = 10$ demographic groups

units : 1000 views per dollar

$v^{\mathrm{des}} = (10^3)\mathbf{1}$

$\begin{bmatrix} 10^3 \\ \vdots \\ 10^3 \end{bmatrix}$

$$R = \begin{bmatrix} 0.97 & 1.86 & 0.41 \\ 1.23 & 2.18 & 0.53 \\ 0.80 & 1.24 & 0.62 \\ 1.29 & 0.98 & 0.51 \\ 1.10 & 1.23 & 0.69 \\ 0.67 & 0.34 & 0.54 \\ 0.87 & 0.26 & 0.62 \\ 1.10 & 0.16 & 0.48 \\ 1.92 & 0.22 & 0.71 \\ 1.29 & 0.12 & 0.62 \end{bmatrix}$$

Objective:

find s so that $v = Rs = v^{des}$

Solution:

Find $\hat{s}$ that minimizes $\left\|Rs - v^{des}\right\|^2$

$$\hat{s} = \begin{bmatrix} 62 \\ 100 \\ 1443 \end{bmatrix}$$

This Least Square formulation does-not take consider any budgetary constraints

Scalar Input Data : Straight Line Fit

$$y = \overset{\theta_2}{\boldsymbol{m}}x + \overset{\theta_1}{\boldsymbol{c}}$$

$$\hat{f}(x) = \theta_1 + \theta_2 x$$

$$m = \theta_2, c = \theta_1, y = \hat{f}(x)$$

Input	Output

$$\underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}}_{A} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}}_{y^d}$$

$\hat{\theta}$ are the parameters of the line that makes least square error

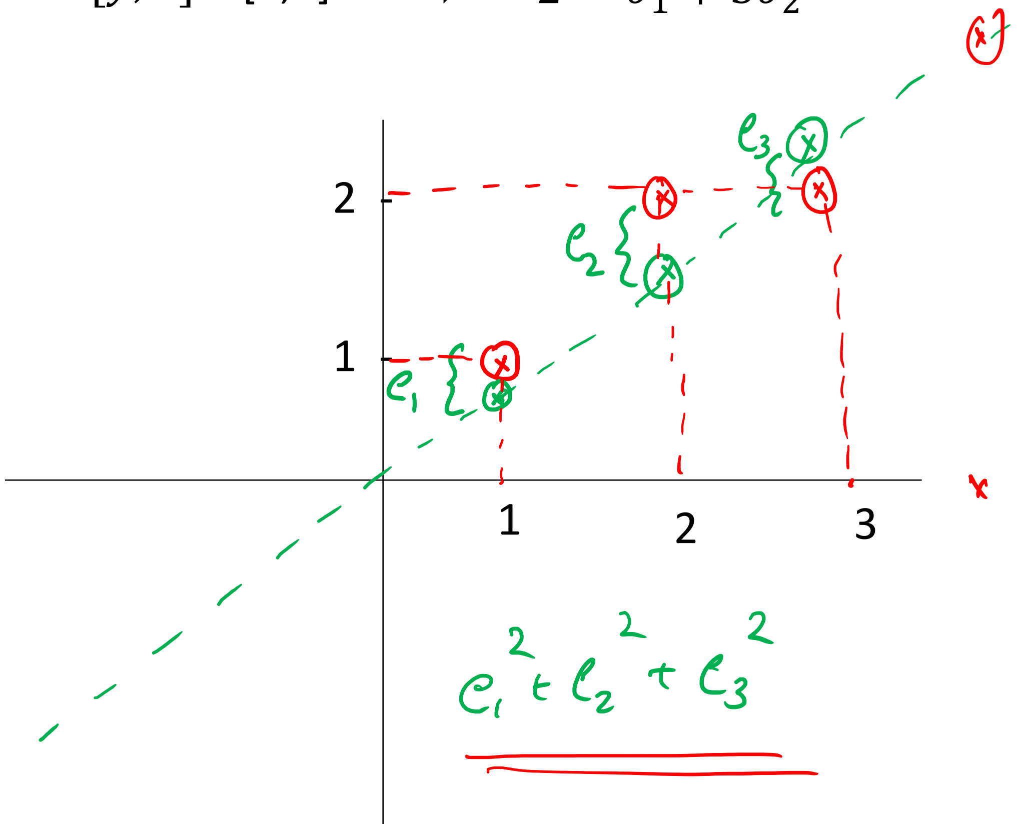
$$\begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix} = (A^T A)^{-1} A^T y^d$$

Example: $y = \theta_1 + \theta_2 x$

$[y, x]$ = [1,1] $\Longrightarrow$ $1 = \theta_1 + \theta_2$

$[y, x]$ = [2,2] $\Longrightarrow$ $2 = \theta_1 + 2\theta_2$

$[y, x]$ = [2,3] $\Longrightarrow$ $2 = \theta_1 + 3\theta_2$



Also known as Linear Regression.

Scalar Input Data : Polynomial Fit

$\hat{f}$ is a polynomial of degree at most $p-1$

$$\hat{f}(x) = \theta_1 + \theta_2 x + \cdots + \theta_p x^{p-1}$$

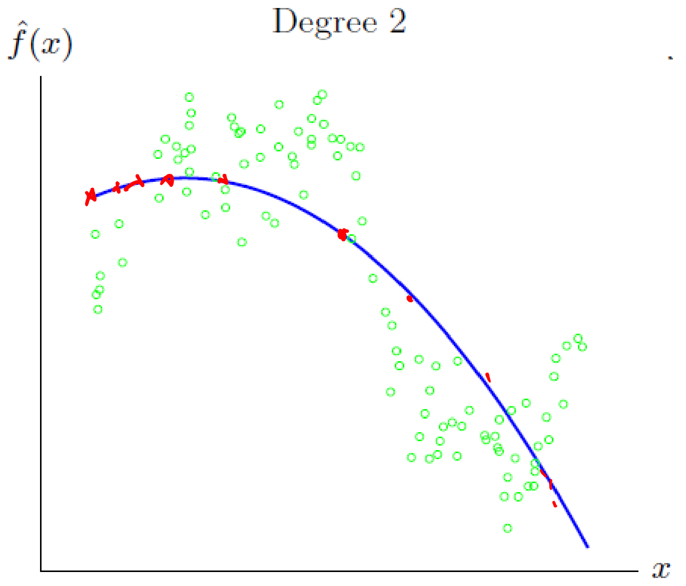
$$= \theta_1 + \theta_2 x + \theta_3 x^2 + \cdots \theta_6 x^5 \text{ (Ex.)}$$

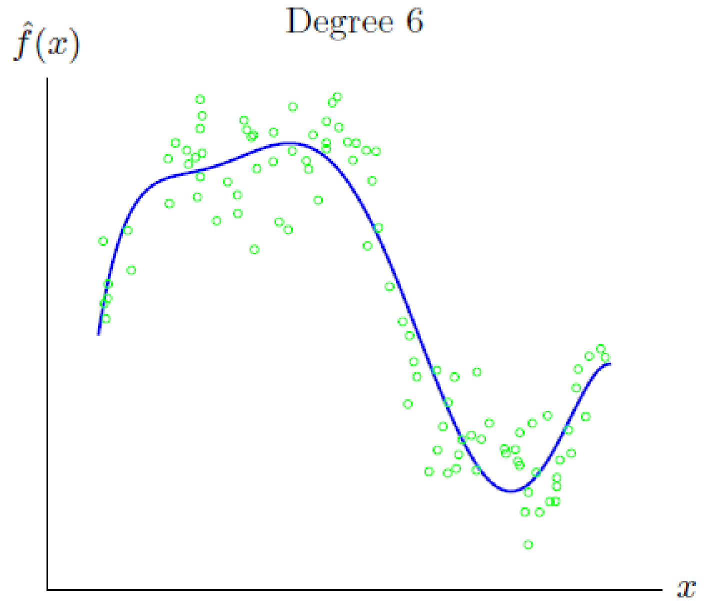
$$A = \begin{bmatrix} 1 & x^{(1)} & \cdots & (x^{(1)})^{p-1} \\ 1 & x^{(2)} & \cdots & (x^{(2)})^{p-1} \\ \vdots & \vdots & & \vdots \\ 1 & x^{(N)} & \cdots & (x^{(N)})^{p-1} \end{bmatrix} = \begin{bmatrix} 1 & x^{(1)} & x^{(1)2} \cdots & x^{(1)5} \\ & & & \\ 1 & x^{(10)} & x^{(10)2} \cdots & x^{(10)5} \end{bmatrix}$$

x^i means the generic scalar value x raised to the ith power

$x^{(i)}$ means the ith observed scalar data value.

Poll: Do you think a polynomial of degree 100 is better suited for the data distribution shown in bottom figure?
A) Yes B) No





Data Fitting

Objective

Given are N input- output (data-prediction) pairs

$$x^{(1)}, \ldots, x^{(N)}, \qquad y^{(1)}, \ldots, y^{(N)}$$

x	y
Input	Output
Data	Prediction
Feature Vector	Label

Common Terminologies

Based on observed data, learn a function $f\colon \mathbb{R}^n \rightarrow \mathbb{R}$ that maps (predicts) any n-vector x to a scalar value

Linear Parameter Model:

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

$f_i\colon \mathbb{R}^n \rightarrow \mathbb{R}$ are pre-defined *basis functions* or *feature mappings*

θ_i are the *model parameters* to be learnt

$f_1(x) = 1$

$f_2(x) = x$

$f_3(x) = x^2$

$f_p(x) = x^{p-1}$

N.N.

F.C.

128

$$\begin{bmatrix} f_1(x^1) & \cdots & f_p(x^1) \\ f_1(x^2) & \cdots & f_p(x^2) \\ f_1(x^N) & \cdots & f_p(x^N) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_p \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$\begin{bmatrix} f_1(x) \\ \vdots \\ f_{128}(x) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{128} \end{bmatrix}$$

Poll: Which of the following does not represent a basis function for fitting a degree – 3 polynomial

A) x B) $3x$ C) x^3 D) x^2

General Regression Model

$$x^{(1)} = [x_1\ x_2 \cdots x_n]$$

1 2 3 4 5

[ ]

Y

[ ]

Regression Model:

$$\hat{y} = x^T\beta + v$$

β is the weight vector

v is the offset

$$\begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & & & \\ 1 & x_1^{(N)} & x_2^{(N)} & \cdots & x_n^{(N)} \end{bmatrix} \begin{bmatrix} v \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$$

$$f_1(x) = 1 \qquad f_i(x) = x_{i-1}, \quad i = 2, \ldots, n+1,$$

$f_1(x) = 1$

$f_2(x) = x_1$

$f_3(x) = x_2$

$\vdots$

Examples:

- Advertising spending on various products vs Total revenue
- Dosages of various drugs vs Blood pressure
- Amount of different fertilizers , water etc. vs crop yield

Least Squares Classifier

Given N Data points and Label for each Data

$$x^{(1)}, \ldots, x^{(N)}, \qquad y^{(1)}, \ldots, y^{(N)}.$$

The outcome y takes only two values : -1 & 1

Steps:

choose basis functions $f_1, \ldots, f_p$,

$$\tilde{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x).$$

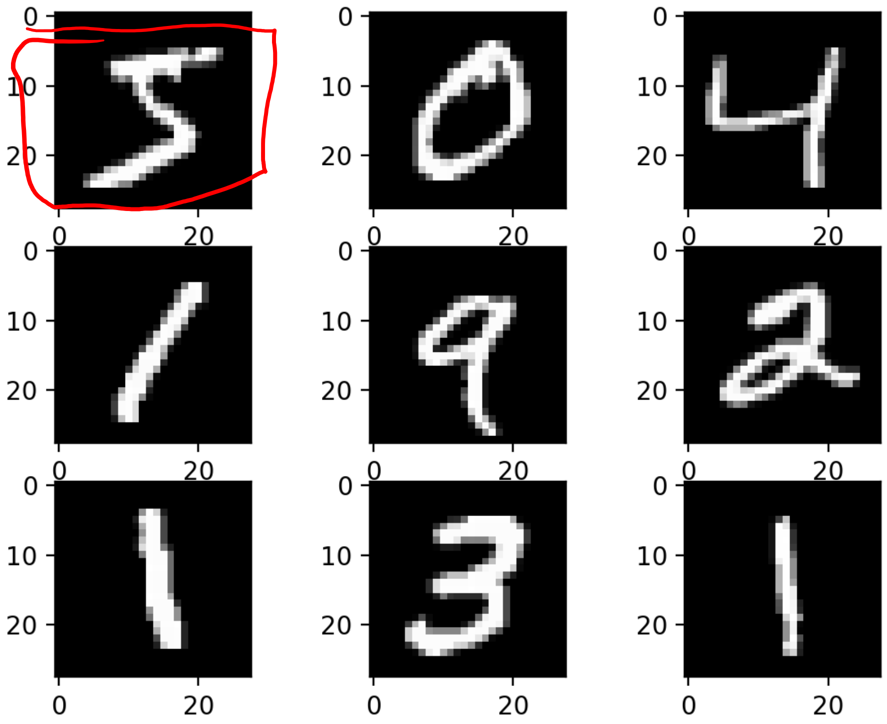
choose the parameters $\theta_1, \ldots, \theta_p$ to minimize the sum squared error

$$(y^{(1)} - \tilde{f}(x^{(1)}))^2 + \cdots + (y^{(N)} - \tilde{f}(x^{(N)}))^2.$$

final classifier is then taken to be

$$\hat{f}(x) = \mathbf{sign}(\tilde{f}(x)).$$

MNIST Classification



MNIST digits Samples

Objective: Train a classifier to classify the digit ‘0’

Data: 60,000 Images (28x28) with labels 0 - 9

Least Squares Classifier for Handwritten Digits 0-9

$f_1(x) = 1$

$f_2(x) = x_1$

$\vdots$

$f_{494}(x) = x_{493}$

The (training) data set contains 60000 images of size 28 by 28.

Pre-processing-steps:

remove the pixels that are nonzero in fewer than 600 training examples.

remaining 493 pixels are shown as the white area

$n = 494$ features

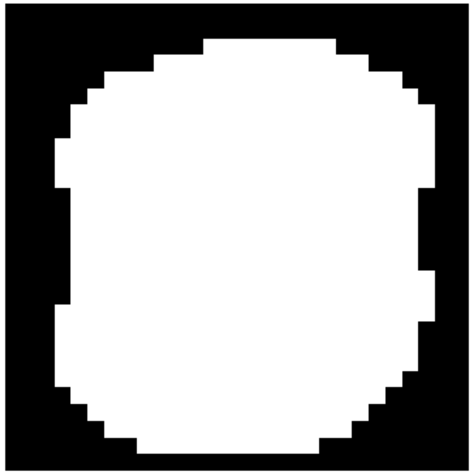
$$\begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_{493}^{(1)} \\ \vdots & & & \vdots \\ 1 & x^{(60,000)} & \cdots & x_{493}^{(60,000)} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{494} \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ \vdots \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

Training Classifier for digit 0: 494

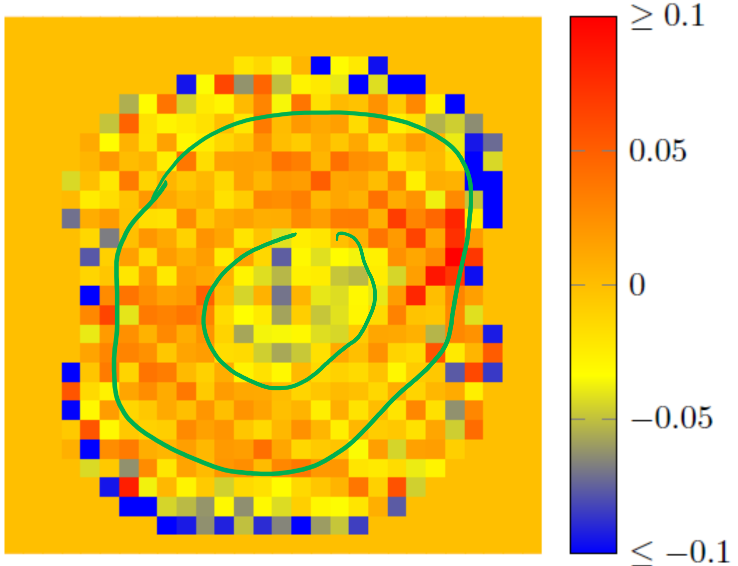
training examples $x^{(i)}$ from class $+1$ (digit zero)

training examples $x^{(i)}$ from class -1 (digits 1–9)

Poll: What is the modified number of weight parameters to be learnt ?
A) 784 B) 493 C) 492
D) 494



Location of the pixels used as features



The coefficients β_k in the least squares classifier that distinguishes the digit zero from the other nine digits.