Linear Regression

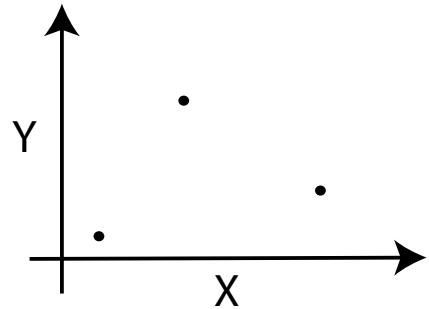
Lecture based on "Dive into Deep Learning" **http://D2L.AI** (Zhang et al., 2020)

Prof. Dr. Christoph Lippert

Digital Health & Machine Learning

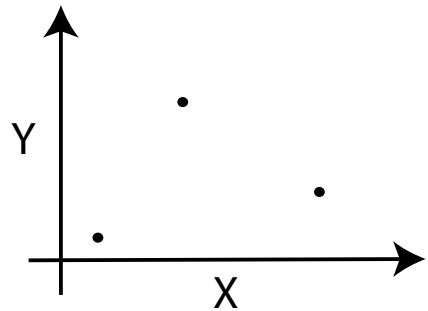
Data

- Let $\mathcal{D}$ denote a **dataset**, consisting of n datapoints $\mathcal{D} = \{ \underbrace{\mathbf{x}_i}_{\text{Inputs}}, \underbrace{y_i}_{\text{Outputs}} \}_{i=1}^{n}$.
- Typical (this course)
  - $\mathbf{x} = \{x_1, \ldots, x_d\}$ multivariate, spanning d features for each observation.
  - y
    - continuous univariate (price, insulin level, etc.).



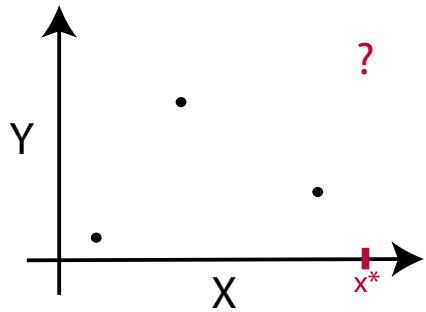
Predictions

- Observed dataset $\mathcal{D} = \{ \underbrace{\mathbf{x}_i}_{\text{Inputs}}, \underbrace{y_i}_{\text{Outputs}} \}_{i=1}^{n}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?



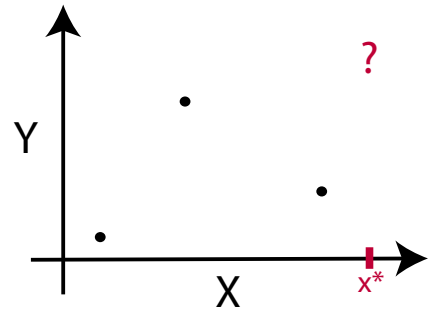
Predictions

- Observed dataset $\mathcal{D} = \{ \underbrace{\mathbf{x}_i}_{\text{Inputs}}, \underbrace{y_i}_{\text{Outputs}} \}_{i=1}^{n}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?



Model

- Observed dataset $\mathcal{D} = \{ \underbrace{\mathbf{x}_i}_{\text{Inputs}}, \underbrace{y_i}_{\text{Outputs}} \}_{i=1}^{n}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?
- To make predictions we need to make assumptions.
- A model $\mathcal{H}$ encodes these assumptions and often depends on some parameters $\boldsymbol{\theta}$.

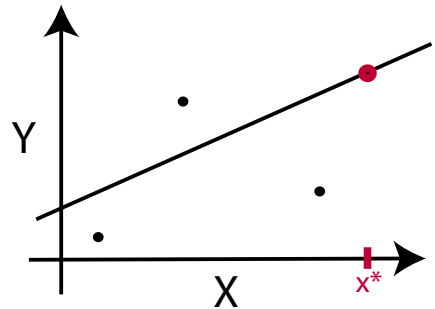


Model

- Observed dataset $\mathcal{D} = \{\underbrace{\mathbf{x}^{(i)}}_{\text{Inputs}}, \underbrace{y^{(i)}}_{\text{Outputs}}\}_{i=1}^{n}$.
- Given $\mathcal{D}$, what can we say about $y^\star$ at an unseen test input $\mathbf{x}^\star$?
- To make predictions we need to make assumptions.
- A model $\mathcal{H}$ encodes these assumptions and often depends on some parameters $\boldsymbol{\theta}$.

- Curve fitting: the model relates $\mathbf{x}$ to y,

$$\hat{y} = f(\mathbf{x}; \underbrace{\boldsymbol{\theta}}_{(w,b)}) = \underbrace{wx + b}_{\text{linear regression}}$$



The **linear model** is expressed as

$$\hat{y} = w_1 \cdot x_1 + \cdots + w_d \cdot x_d + b$$

Equivalently using vectors:

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b,$$

where $\mathbf{x}$ corresponds to a single data point.

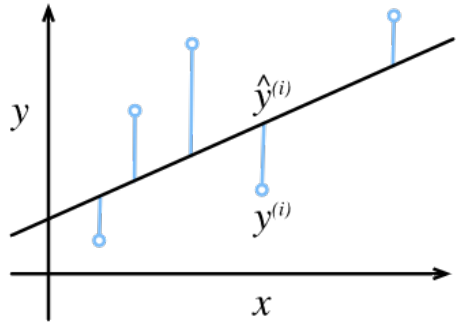
Or, using the **design matrix** $\mathbf{X}$ for an entire data set.

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \dots & x_d^{(n)} \end{pmatrix}$$

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b$$

The **loss function** quantifies the distance between the **real** and **predicted** value of the target.

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2.$$



$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2.$$

$$\mathbf{w}^*, b^* = \underset{\mathbf{w},b}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)} \right)^2 .$$

Equivalently, we can subsume b into $\mathbf{w}$ by appending $1s$ $\mathbf{X}$

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{2n} (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y})$$

where

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \dots & x_d^{(1)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{(n)} & \dots & x_d^{(n)} & 1 \end{pmatrix}, \qquad \mathbf{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_d \\ b \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Analytic Solution

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{2n} (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y})$$

Taking the gradient with respect to $\mathbf{w}$ and setting it to $\mathbf{0}$ yields the analytic solution:

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

Summary

- Key ingredients in a machine learning model are
  - training data
  - a loss function
  - an optimization algorithm
  - the model
- Linear regression with a squared loss has an analytic solution.