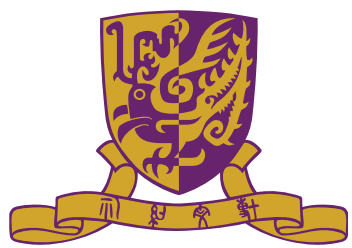
DDA5001 Machine Learning

Linear classification I: The Perceptron & Linear Regression: Least Squares

Xiao Li

School of Data Science
The Chinese University of Hong Kong, Shenzhen

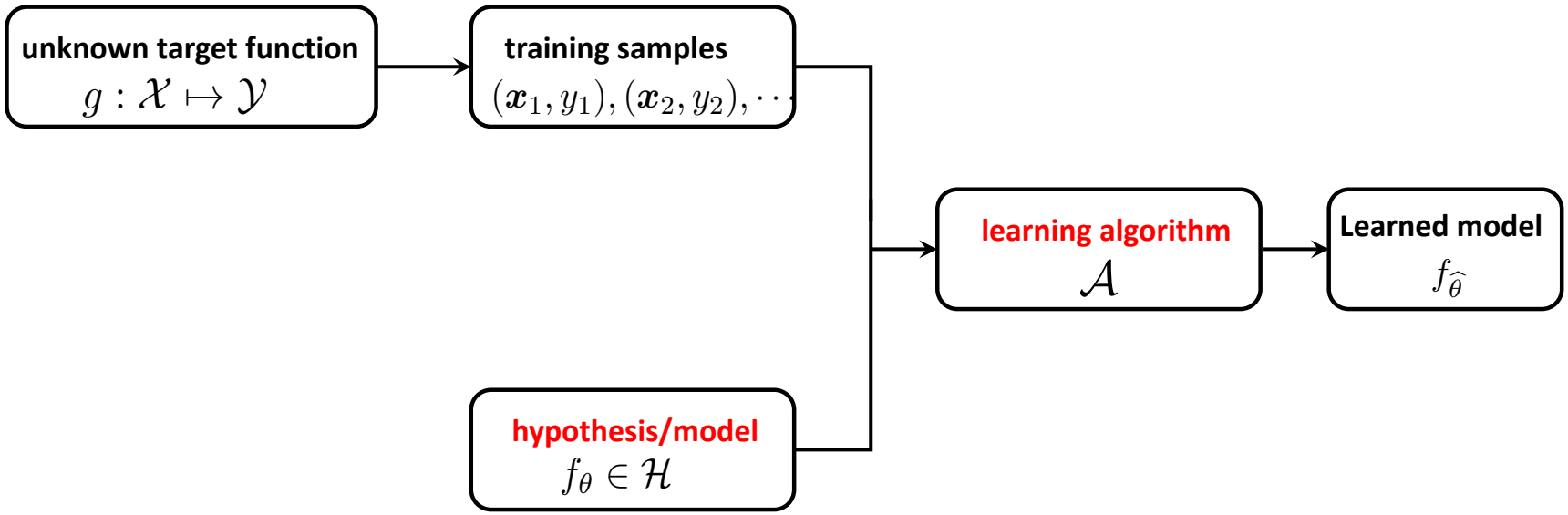


Recap: Supervised Learning Components

Components:

- Target function $g : \mathcal{X} \to \mathcal{Y}$ (underlying credit approval model)
- Training dataset: $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ (historical records)
- Hypothesis space: $\mathcal{H}$ (learning scope to approximate g)
- Hypothesis/model: $f_{\boldsymbol{\theta}}$ (model to be determined)
- Optimization algorithm: $\mathcal{A}$ (learning the model from data)

Recap: Supervised Learning: High-level View



Hopefully,

$$f_{\widehat{\boldsymbol{\theta}}} \approx g$$

Predict/decision: When a new sample data (test data) $\boldsymbol{x}$ comes, the label is predicted as:

$$y \leftarrow f_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{x}).$$

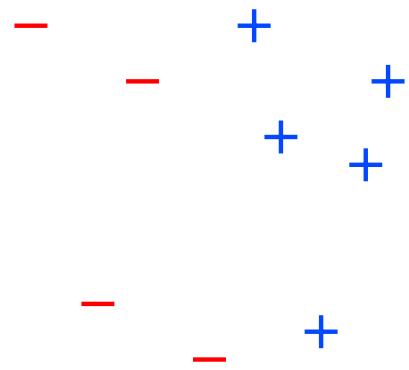
$\rightsquigarrow$ We now discuss linear models for classification and regression.

Linear Classification I — The Perceptron

Linear Regression — Least Squares

Classification

Consider the following illustration of the credit approval example:

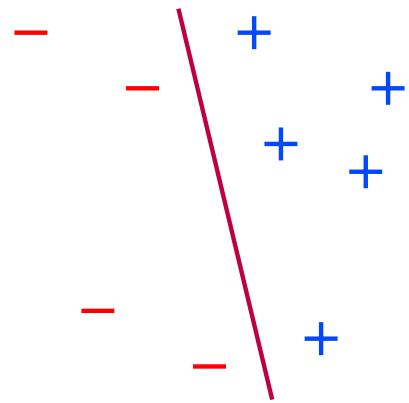


- In the above figure, we can regard each point as one applicant's data (sample). This forms the training data pairs $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ with $y_i = \pm 1$ (approve or deny).
- Classification is to learn a model $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \{-1, +1\}$ based on the training samples to correctly classify the training samples (to approximate the underlying unknown g). This is called binary classification since we only have two classes.

What type of model $f_{\boldsymbol{\theta}}$ can we choose?

Linear Classification

Consider the following illustration of the credit approval example:



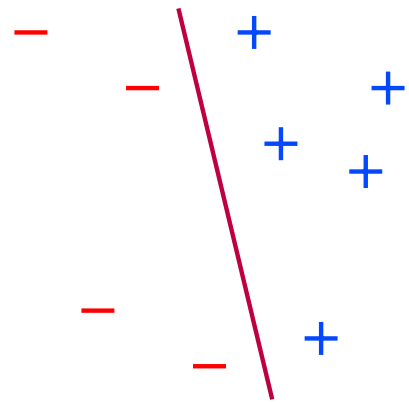
Linear classification: Find a linear model based on the training sample to correctly classify the training samples. Namely, $f_{\boldsymbol{\theta}}$ takes the linear form

$$\boxed{f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} + b.}$$

Correspondingly, the hypothesis space is given by

$$\mathcal{H} := \{f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} + b : \boldsymbol{\theta} \in \mathbb{R}^d, b \in \mathbb{R}\},$$

Linear Classification



The linear model is learned to separate the dataset:

$$\begin{cases} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > 0 & \text{if } y_i = +1, \\ f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) < 0 & \text{if } y_i = -1, \end{cases} \quad \forall\ i = 1, ..., n.$$

The classification result can be represented using the "sign" function, i.e.,

$$y_i = \text{sign}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)),$$

where

$$\text{sign}(z) = \begin{cases} +1, & \text{if } z > 0, \\ -1, & \text{if } z < 0, \\ [-1, +1], & \text{if } z = 0. \end{cases}$$

Absorbing the Bias Term

- The term "b" in the linear model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} + b$ is called bias term.
- Let $\theta_0 = b$ and $x_0 = 1$, and now, we define

$$\widetilde{\boldsymbol{x}} = (x_0, \boldsymbol{x}) \in \mathbb{R}^{d+1}$$

$$\widetilde{\boldsymbol{\theta}} = (\theta_0, \boldsymbol{\theta}) \in \mathbb{R}^{d+1}$$

It then follows that

$$\widetilde{\boldsymbol{\theta}}^\top \widetilde{\boldsymbol{x}} = \boldsymbol{\theta}^\top \boldsymbol{x} + \theta_0 x_0 = \boldsymbol{\theta}^\top \boldsymbol{x} + b.$$

In the sequel, we will use $\boldsymbol{\theta}$ and $\boldsymbol{x}$ to represent $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\boldsymbol{x}}$ to ease notation, respectively. Then, we can compactly write the linear model as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} \quad \text{and} \quad y = \operatorname{sign}(f_{\boldsymbol{\theta}}(\boldsymbol{x})).$$

In this way, the bias term is absorbed and the linear model representation is more compact.

$\rightsquigarrow$ After choosing the model, we need an algorithm to find the classifier.

Linear Classification Algorithm: The Perceptron

- The perceptron was invented in 1943 by McCulloch and Pitts.
- It is a simple linear classification algorithm used to find a separation line based on the training dataset.
- The perceptron was intended to be a machine, rather than a program, and while its first implementation was in software for the IBM 704.
- Later in 1957, it was implemented in hardware-level machine and the machine is called "Mark I perceptron", designed for image recognition.

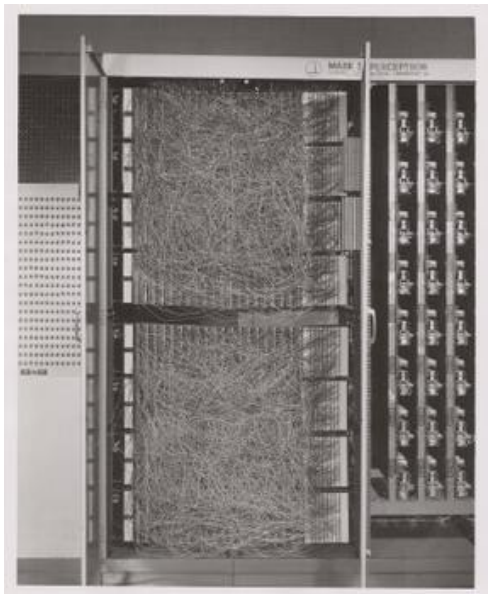
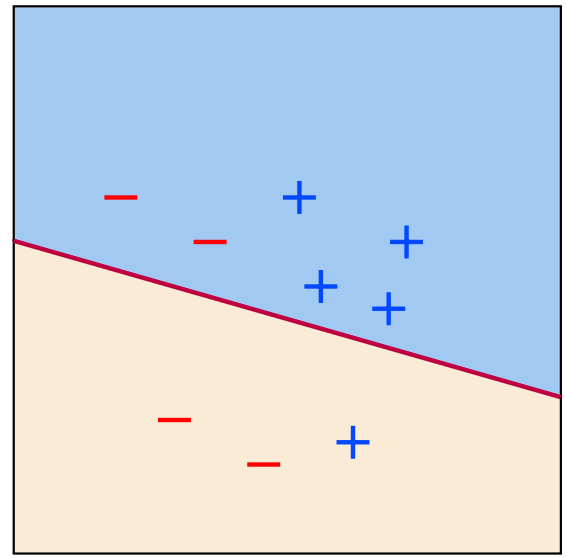


Figure: Mark I Perceptron machine.

The Perceptron: Algorithm Procedure

- Given the training dataset $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, with $y_i = \pm 1$.
- Linear classification model

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}, \quad y = \operatorname{sign}(f_{\boldsymbol{\theta}}(\boldsymbol{x})).$$



- Pick a misclassified data $\boldsymbol{x}_i$ (any misclassified one)

$$\underbrace{\operatorname{sign}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right)}_{=\operatorname{sign}(\boldsymbol{\theta}^\top \boldsymbol{x}_i)} \neq y_i.$$

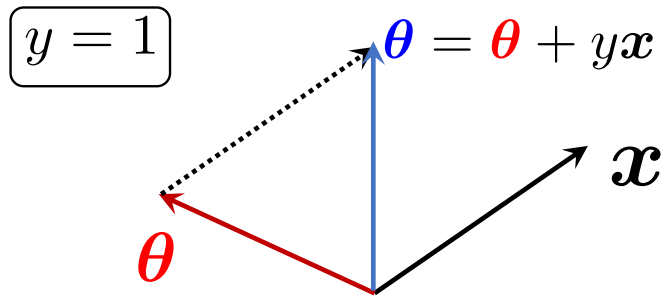
- Update rule

$$\boxed{\boldsymbol{\theta} \quad = \quad \boldsymbol{\theta} \quad + \quad y_i \boldsymbol{x}_i.}$$

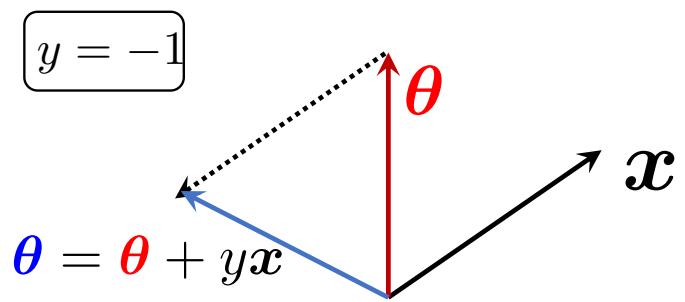
The Perceptron: Interpretation

- Update rule

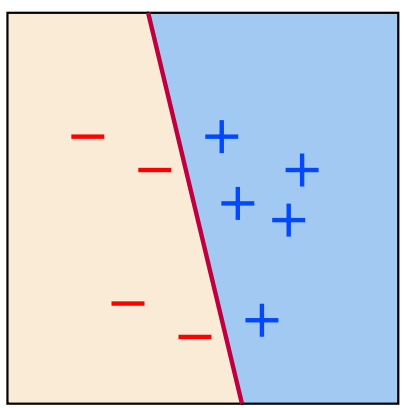
$$\boldsymbol{\theta} = \boldsymbol{\theta} + y_i \boldsymbol{x}_i$$



- This update rule pushes the line to correctly classify the picked data. Iteratively repeat this procedure, until all training samples are correctly classified.



Learned model:



Example

Which statement is true?

(A) The perceptron algorithm is used for unsupervised classification.

(B) The perceptron algorithm can find a classifier (separation line) in one update step.

(C) The perceptron algorithm will stop at the same separation line for every possible runs.

(D) The perceptron is tailored for binary classification.

Answer: (D). For (A), we have label, hence it is supervised. Actually, classification always has label and is always supervised. For (B), it depends on the data. It may take many iterative steps to converge. For (C), it may stop at different separation lines, depending on which misclassified data to choose.

Discussions

From the perceptron learning algorithm, we still have the following questions:

- The learned model successfully classifies the training data points. Does this mean that it will perform well for the future/test data points?
- Is the learned model $f_{\widehat{\boldsymbol{\theta}}}$ optimal in terms of classification?
- Is perceptron guaranteed to converge to a separation line?
- How about the training samples are not linearly separable? Namely, the two classes have an intersection. Does perceptron still work?

The first question will be answered in the training versus testing section.

The second question will be answered in the section of support vector machine.

The third and forth questions will be answered in our Homework 1.

$\rightsquigarrow$ We will have tutorial 2 (Sep 16) to introduce how to implement perceptron, which will be helpful for completing Homework 1.

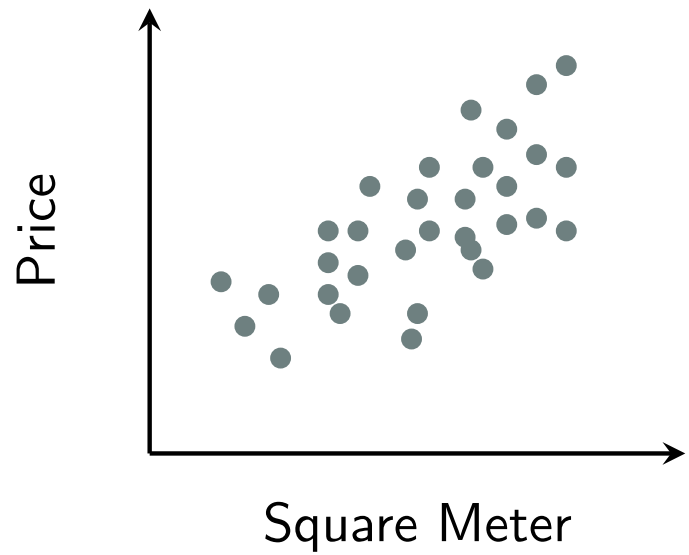
Linear Regression — Least Squares

Regression

Data: Given past sales data $\{(x_i, y_i)\}_{i=1}^n$, where

$x_i \in \mathbb{R}$ represents square meter

y_i represents apartment price



Regression is to learn a model f_θ that captures relationship between x_i and y_i. Given a new apartment with square meter x, predict $y = f_\theta(x)$.

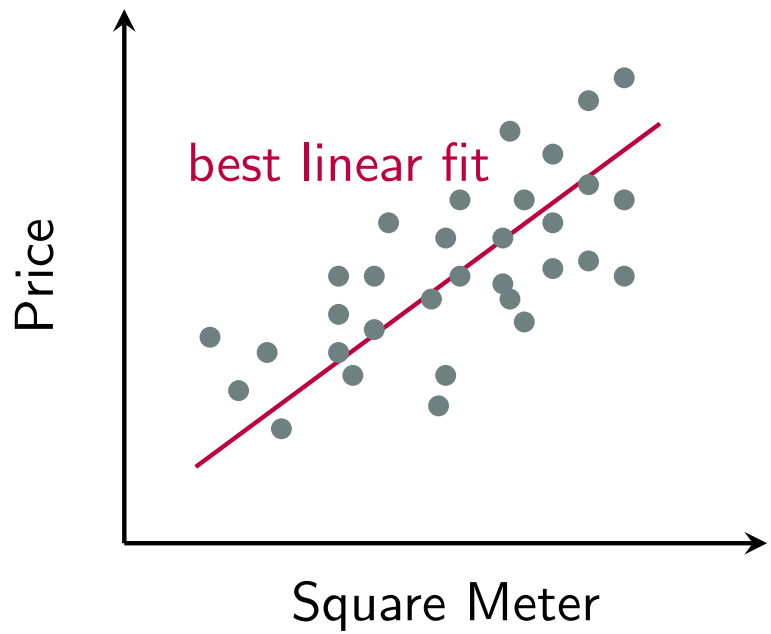
Regression means that y is real-valued (continuous).

What is the difference between regression and classification? In regression, y is continuous, while in classification it is categorical.

Linear Regression

- If we choose the hypothesis space to be linear model, then we have linear regression. That is, we assume that the apartment price is a linear function of square meter.
- For the previous simple example, the linear model can be expressed as

$$y_i \approx f_\theta(x_i) = \theta x_i.$$



Multi-dimensional Linear Regression

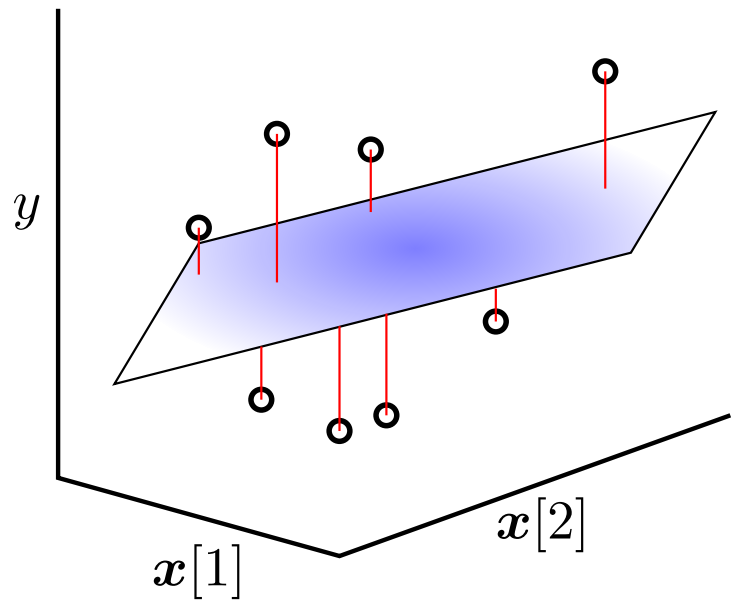
Data: Given past sales data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where

$\boldsymbol{x}_i \in \mathbb{R}^d$ represents square meter, metro, schools...
y_i represents price

Linear model:

$$y_i \approx f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \boldsymbol{\theta}^\top \boldsymbol{x}_i,$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$.



How to find the best linear fit? Least squares.

Least Squares

The learning problem can be formulated as:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i \right)$$

Since our goal is to enforce $y_i \approx f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ and the label y is continuous real-valued, we can choose the most common ℓ_2-loss function:

$$\ell(a, b) = (a - b)^2.$$

The choice of ℓ_2-loss results in the famous least squares (LS) formulation:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\theta}^\top \boldsymbol{x}_i - y_i)^2$$

$\rightsquigarrow$We can also select absolute value loss, leading to robust linear regression formulation (Homework 1).

Least Squares in Matrix Notation

Matrix representation of data and label:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{bmatrix} \in \mathbb{R}^{n \times d} \qquad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

Fact:

$$\begin{bmatrix} \boldsymbol{\theta}^\top \boldsymbol{x}_1 - y_1 \\ \vdots \\ \boldsymbol{\theta}^\top \boldsymbol{x}_n - y_n \end{bmatrix} = \boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}$$

Least squares:

$$\min_{\boldsymbol{\theta}} \ \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\theta}^\top \boldsymbol{x}_i - y_i)^2 \quad \Longleftrightarrow \quad \boxed{\min_{\boldsymbol{\theta}} \ \frac{1}{n} \|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2}$$

$\rightsquigarrow$ Next lecture: Solution of least squares and MLE.