Concepts of Learning From Data

Ruwen Qin
Stony Brook University

August 21, 2025

1 An Example

The following is an example of image data with two classes: with and without crack.

Figure 1: Image data of concrete surface with and without crack

Can you tell the class of each image below: with or without crack? Everyone including those having no domain knowledge can finish this task easily after reviewing the examples in Figure 1.

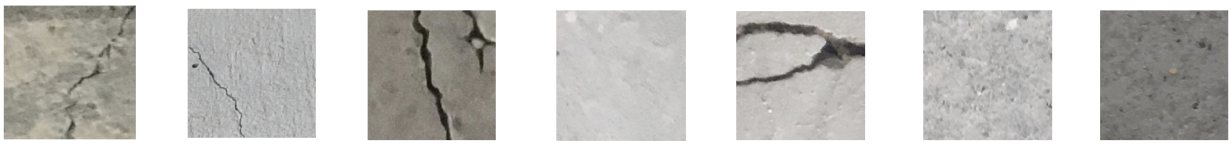

Figure 2: Images to be classified

Can we teach machine (e.g., a computer) to perform the same task as you did just now? Learning from examples or comparison is a common way of gaining new knowledge. Machine learning is possible when a pattern exists but we cannot pin it down mathematically. If we have data, we can discover the pattern from the data.

This module, mainly developed based on the reference [1], introduces the basic concepts of machine learning to prepare you for studying related modules. Interested students can refer to that reference for details.

2 Learning Diagram

Figure 3 illustrates major components of (supervised) learning and solution:

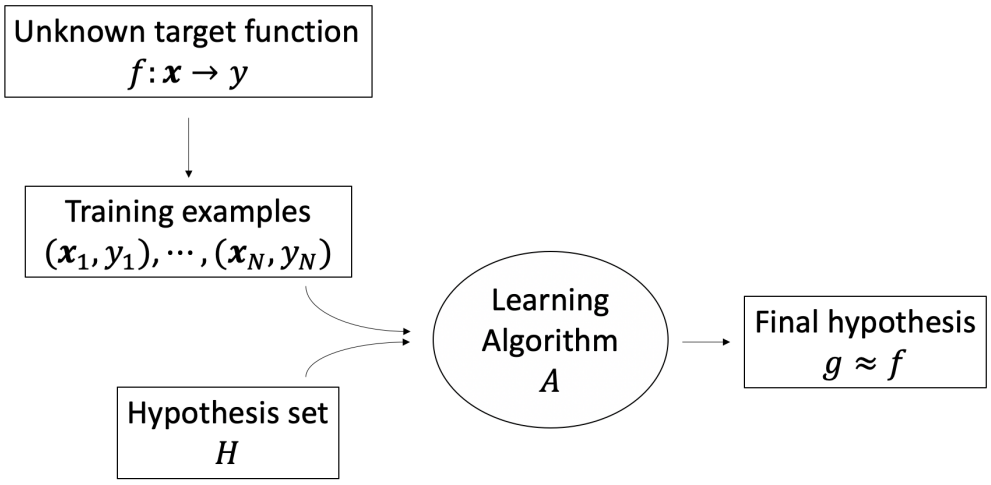

Figure 3: Learning and solution components

2.1 Learning Components

- Input, $\boldsymbol{x}$
- output, y
- Target function, f
- Training data: $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$
- Hypothesis: $g : \boldsymbol{x} \rightarrow y$

Regarding the example in Section 1, the input $\boldsymbol{x}$ is image data, the output y is image class, the target function f is unknown and we would like to infer, and the training data $\{\boldsymbol{x}_n, y_n\}$ are illustrated in Figure 1. Given the training data, we will learn and find the final hypothesis g that predicts the output for an given input like those images in Figure 2.

2.2 Solution Components

The two solution components are:

- Hypothesis set: H
- Learning algorithm: A

In Figure 3, the hypothesis set and learning algorithm together are referred to as the learning model.

2.3 A Simple Hypothesis Set - Perceptron

Using the image classification problem as an exmaple. For an input, $\boldsymbol{x} = [x_1, \ldots, x_d]$ (pixel values of an image),

$$y = \begin{cases} \text{crack}, & \sum_{i=1}^{d} w_i x_i + w_0 > 0) \\ \text{no-crack}, & \sum_{i=1}^{d} w_i x_i + w_0 \leq 0) \end{cases} \tag{1}$$

The function $h \in H$ can be written as

$$h = \text{sign}(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}) \tag{2}$$

where $\boldsymbol{w} = [w_0, w_1, \ldots, w_d]^{\mathrm{T}}$ are learnable parameters and $\boldsymbol{x} = [1, x_1, \ldots, x_d]^{\mathrm{T}}$ are the input.

2.4 Learning Algorithm

Each hypothesis is a hyper-plane in the space of input $\boldsymbol{x}$, which separates observations into two groups. The learning algorithm picks $g \approx h$ from the hypothesis set H. The final hypothesis g is the one that minimizes the number of misclassified points, as Figure 4 shows.

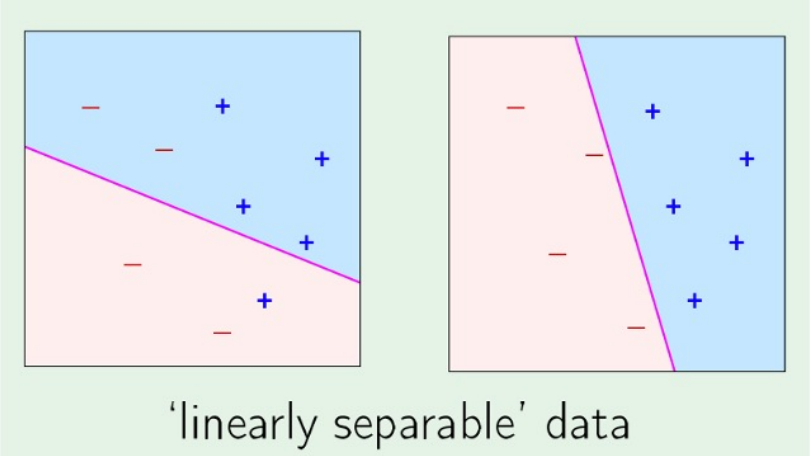

Figure 4: Linearly separable data

3 Learning Diagram with Error Measure

3.1 Error Measure

How well does a hypothesis $h(\boldsymbol{x})$ approximate y? Point wise, the prediction error is the difference between the predicted output, $h(\boldsymbol{x}_n)$, and the correct output (i.e., ground truth), y_n:

$$e_n = h(\boldsymbol{x}_n) - y_n = h(\boldsymbol{x}_n) - f(\boldsymbol{x}_n). \tag{3}$$

A loss function measures the overall error. The in-sample loss function is

$$L_{in}(h) = \sum_{n=1}^{N} l(h(\boldsymbol{x}_n), f(\boldsymbol{x}_n)), \tag{4}$$

and the out-sample loss is

$$L_{out}(h) = \mathrm{E}_{\boldsymbol{x}}[l(h(\boldsymbol{x}), f(\boldsymbol{x}))]. \tag{5}$$

There are various choices of the function l that penalizes the prediction error. For example, if l is the first norm of errors, $\boldsymbol{e} = [e_1, \ldots, e_N] = h(\boldsymbol{x}) - f(\boldsymbol{x})$, and then the loss function L is the sum of absolute errors.

3.2 The Role of Loss Function in Learning Algorithm

The loss function l provides a pointwise error measure of a hypothesis $h(\boldsymbol{x})$ with the ground truth $f(\boldsymbol{x})$. The learning algorithm uses the error function as a reference for searching the final hypothesis $g(\boldsymbol{x})$, as shown in Figure 5. The final hypothesis is the one that minimizes the loss function.

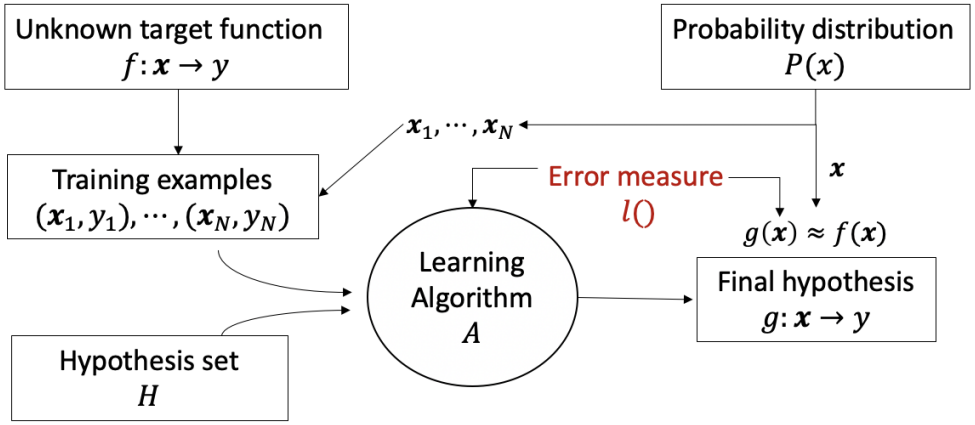

Figure 5: Learning diagram with error measure

4 Learning Diagram with a Noisy Target Function

The target function is not always a function[1]. Mathematically speaking, two identical inputs may have different outputs. Therefore, we specify target distribution instead:

$$P(y|\boldsymbol{x}) \tag{6}$$

$(\boldsymbol{x}, y)$ is generated from the joint distribution

$$P(y|\boldsymbol{x})P(\boldsymbol{x}). \tag{7}$$

Noisy target is the deterministic target function, $f(x) = E[y|\boldsymbol{x}]$, plus noise, $y - f(x)$. The deterministic target function is a special case of the noisy target.

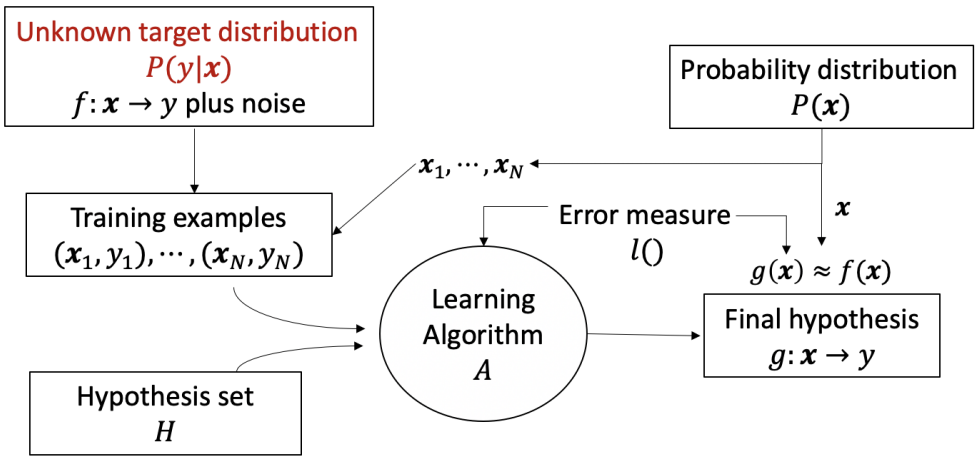

Figure 6: Learning diagram with noise

The target distribution $P(y|\boldsymbol{x})$ is what we are trying to learn. The input distribution $P(\boldsymbol{x})$ quantifies relative importance of $\boldsymbol{x}$. They both convey probabilistic aspects of input $\boldsymbol{x}$ and output y.

[1]The domain of a function, X, is a set of all values for which the function is defined. The codomain of the function Y, is the set into which all of the output of the function is constrained to fall. The function $f : X \rightarrow Y$ is a one-to-one maping relationship between an element in the domain and an element in the codomain. That is, $\forall \boldsymbol{x} \in X$, f maps $\boldsymbol{x}$ to one and only one $y \in Y$.

5 Nonlinear Hypothesis

Figure 7 contains two classes of data points, which are indicated by red cross and blue circle symbols. Can we separate those two classes of data points? Certainly we can, but the hypothesis that separates them is nonlinear.

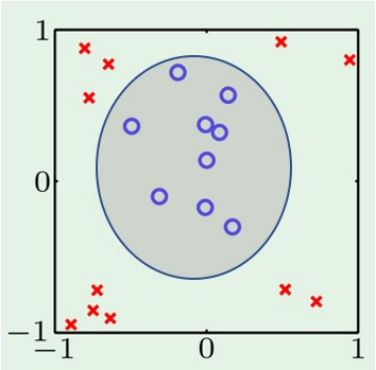

Figure 7: A Nonlinear Hypothesis

For some cases, we may transform the data so that we can separate them using a hyperplane.

- original input data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$
- transform the data: $\boldsymbol{z}_n = \Phi(\boldsymbol{x_n})$, for $n = 1, \ldots, N$
- separate data in Z-space with the hypothesis $\tilde{g}(\boldsymbol{z}) = \text{sign}(\tilde{\boldsymbol{w}}^{\mathrm{T}} \boldsymbol{z})$
- The hypothesis in the X-space: $g(\boldsymbol{x}) = \tilde{g}(\Phi(\boldsymbol{x})) = \text{sign}(\tilde{\boldsymbol{w}}^{\mathrm{T}} \Phi(\boldsymbol{x}))$

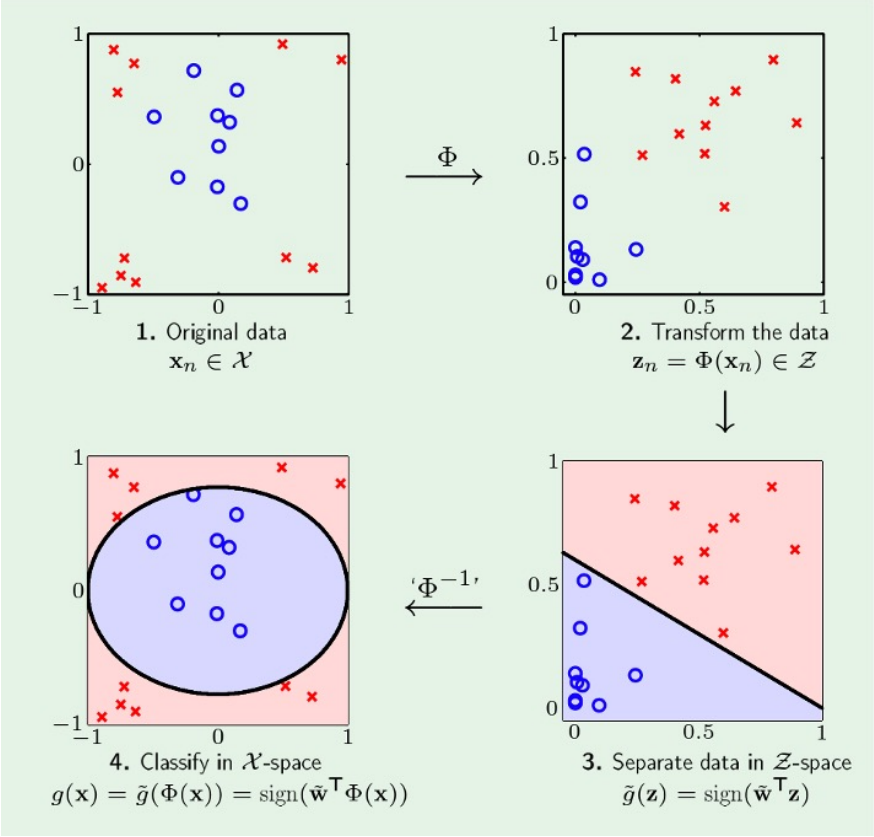

Figure 8: Transform function $\phi = \boldsymbol{x}^2$

6 Types of Learning

- Supervised learning: in the training dataset, every input $\boldsymbol{x}_n$ comes with an output label y_n.
- Unsupervised learning: learning without y_n.
- Semi-supervised learning: Learning with some y_n.
- Reinforced learning: Explicit y_n is not available, but implicit $\widehat{y}_n$ with its goodness.

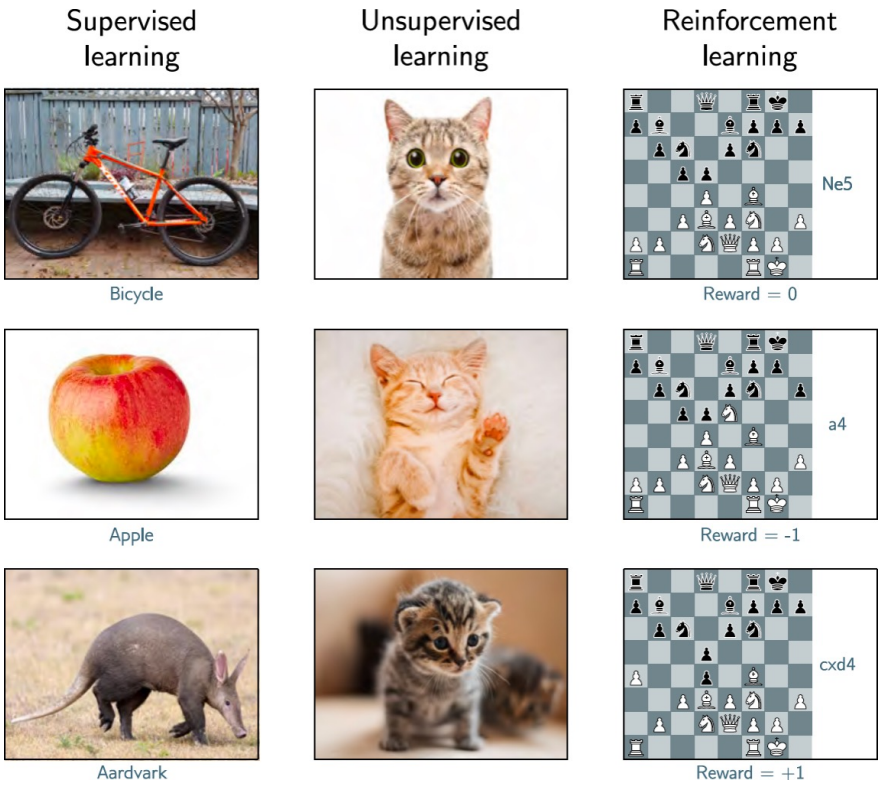

Figure 9: Different type of learning (image source: [2])

7 Some Machine Learning Problems

- *Classification*: this is the problem of assigning a category to each item.
- *Regression*: this is the problem of predicting a real value for each item.
- *Dimensionality reduction*: this problem consists of transforming an initial representation of items into a lower-dimensional representation while preserving some properties of the initial representation.
- *Clustering*: this is the problem of partitioning a set of items into homogeneous subsets.

References

[1] Yaser S Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*, volume 4. AMLBook New York, 2012.

[2] Simon JD Prince. *Understanding Deep Learning*. MIT press, 2023.