CSE 4553
Machine Learning
Lecture 1: Introduction to Machine Learning

Winter 2024

Dr. Hasan Mahmud | hasan@iut-dhaka.edu

Contents

- Introduction
- What is Machine Learning?
- Types of machine learning
- Machine Learning applications

Introduction

“Data is abundant and cheap but knowledge is scarce and expensive.”

- Machine = computer, computer program
- Learning = improving performance on a given task, based on experience / examples

What is machine learning?

- Arthur Samuel (1959). Machine Learning:
  - *Field of study that gives computers the ability to learn without being explicitly programmed.*
- Tom Mitchell (1998). Well-posed Learning Problem:
  - *A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*

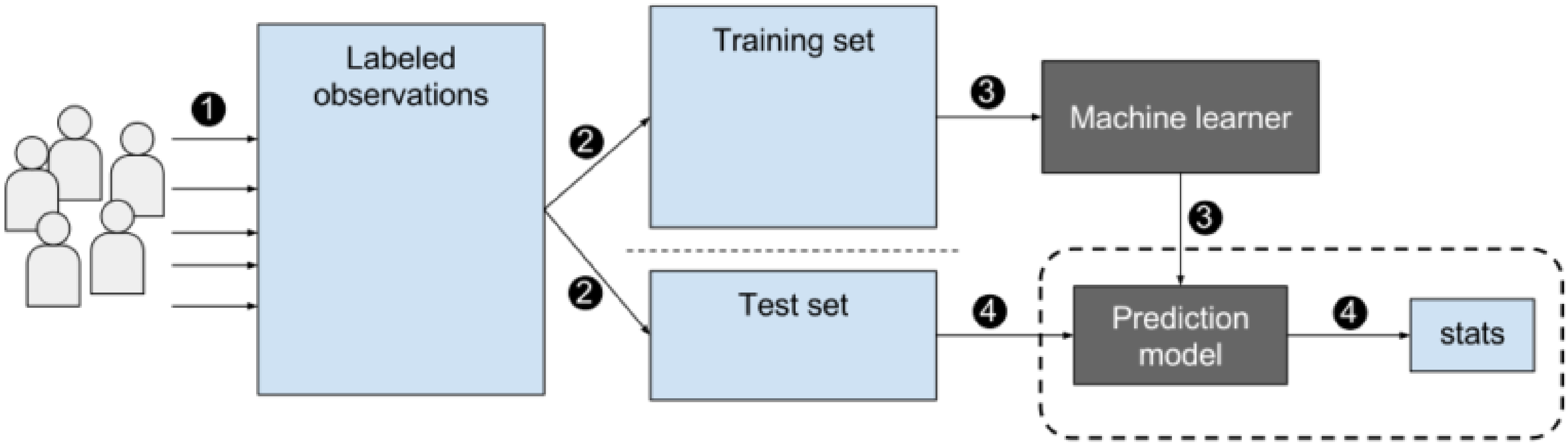
What is machine learning?...

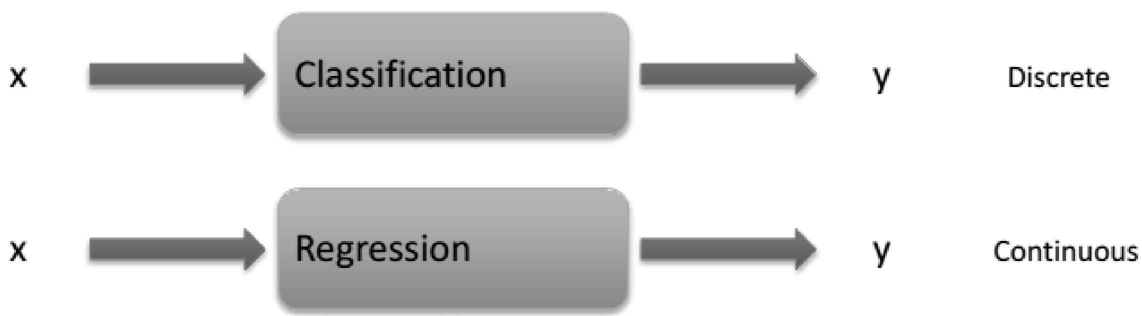
- Improving some measure of performance P when executing some task T through some type of training experience E
- E.g. 1
  - So if you want your program to predict, for example, traffic patterns at a busy intersection (task T), you can run it through a machine learning algorithm with data about past traffic patterns (experience E) and, if it has successfully “learned”, it will then do better at predicting future traffic patterns (performance measure P).
- E.g. 2
  - Task T: Assign label of fraud or not fraud to credit card transaction
  - Performance measure P: Accuracy of fraud classifier with higher penalty when fraud is labeled as not fraud
  - Training experience E: Historical credit card transactions labeled as fraud or not

Types of learning

- Supervised (inductive) learning
  - Training data includes desired outputs
- Unsupervised learning
  - Training data does not include desired outputs
- Semi-supervised learning
  - Training data includes a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

Supervised learning





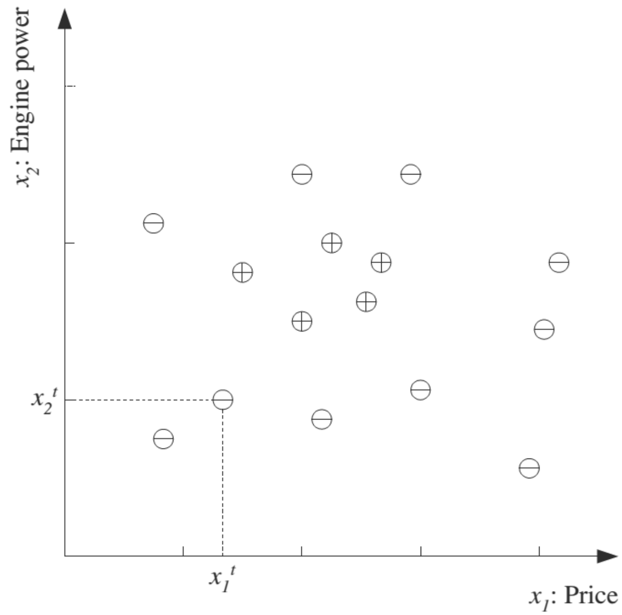
Some examples for supervised algorithms include Linear Regression, Decision Trees, Random Forest, k-nearest neighbours, SVM, Neural Network etc.

Supervised learning…

- Given a set of data points $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ associated to a set of outcomes, $\{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\}$ we want to build a classifier that learns how to predict y from x.

Input   Target Label

$$D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$$



- Training
- Testing
- Hypothesis
- Loss function
- Cost estimation

- We want estimate the prediction function h, by minimizing prediction error, $\sum(y^{(i)} - h(x^{(i)}))^2$

Supervised learning...

Input Target label

True data distribution

$$\min_f \mathbb{E}_{(x,y)\sim D}\left[l(f(x),y)\right]$$

Loss function Model

Training: find the model f that minimizes the **expected** loss

Zero/one loss:

$$\ell(y,\hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise} \end{cases}$$

Squared loss:

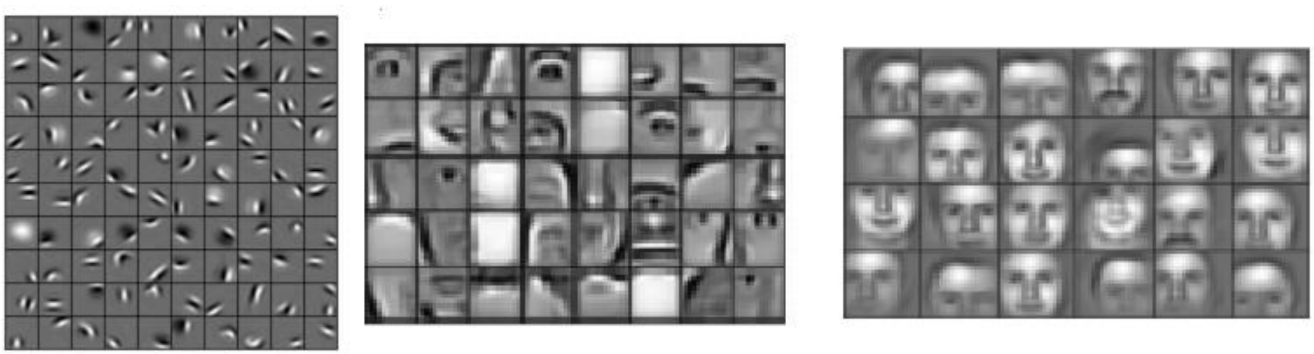
$$\ell(y,\hat{y}) = (y-\hat{y})^2$$

- Training
- Testing
- Hypothesis
- Loss function
- Cost estimation

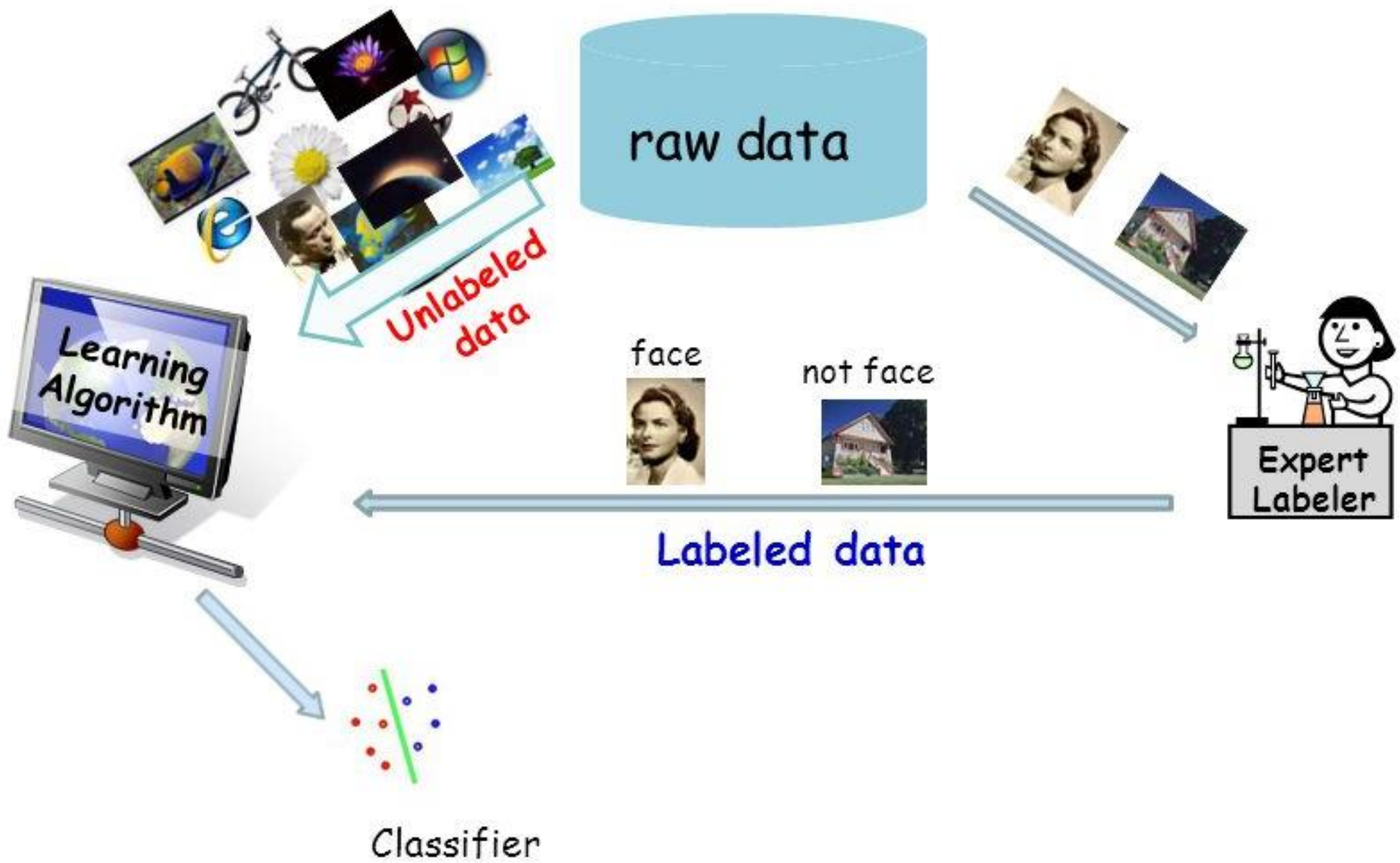
Unsupervised Learning

- The goal of unsupervised learning is to find hidden patterns in unlabeled data, $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

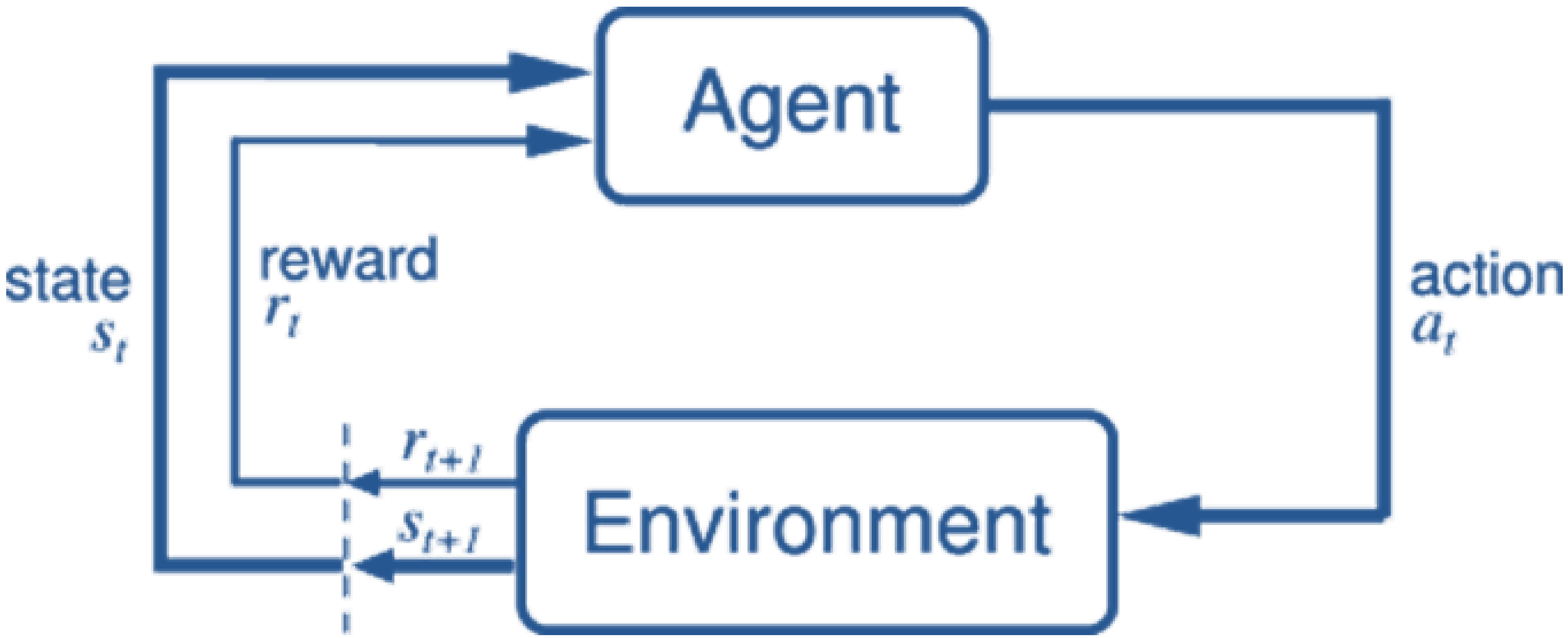
$$D = \{x^{(i)}\}_{i=1}^{n}$$



Semi supervised learning

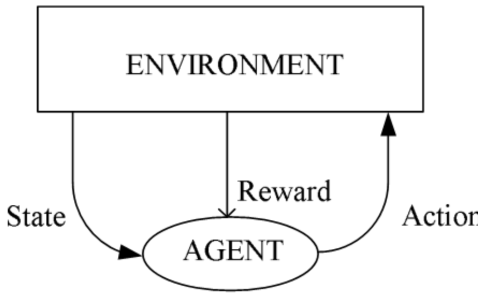


Reinforcement Learning



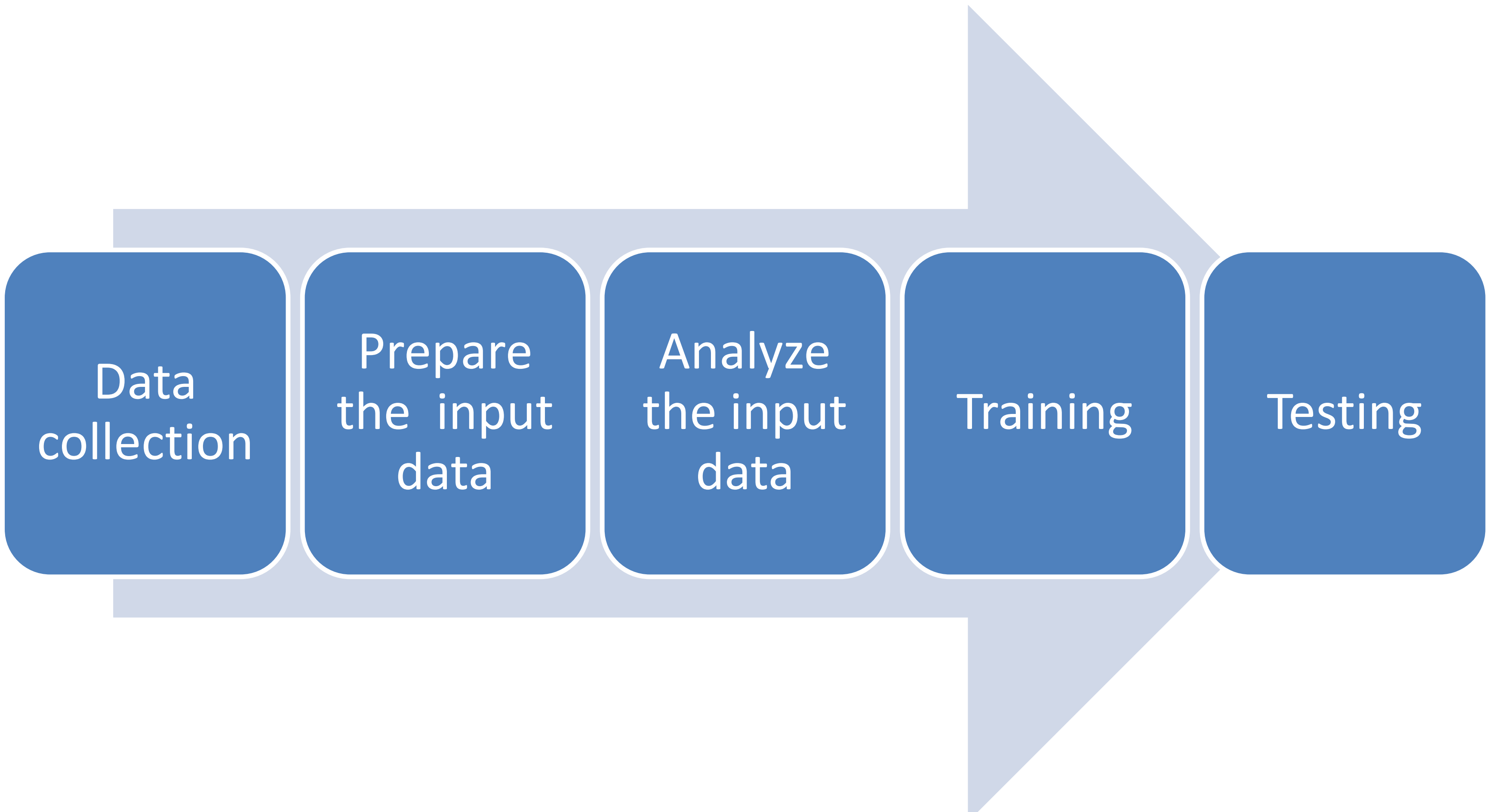
Reinforcement learning…

- Basic elements of a reinforcement learning problem:
  - *Agent*: the decision maker (e.g. game player, robot). It has *sensors* to observe the environment (e.g. robot camera).
  - *Environment* (e.g. board, maze). At any time t, the environment is in a certain *state* s_t that is one of a set of possible states $\mathcal{S}$ (e.g. board state, robot position). Often, there is an initial state and a goal state.
  - A set $\mathcal{A}$ of possible *actions* a_t (e.g. legal chess movements, possible robot steps). The state changes after an action: $s_{t+1} = \delta(s_t, a_t)$. The solution requires a sequence of actions.
  - *Reward* $r_t = r(s_t, a_t) \in \mathbb{R}$: the feedback we receive, usually at the end of the game. It helps to learn the policy.
  - *Policy* π: $\mathcal{S} \to \mathcal{A}$: a control strategy for choosing actions that achieve a goal.



$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \dots$$

Steps developing a machine learning application



References

- Chapter 2: Introduction to Machine learning by Ethem Alpydin.