Learning

YZM 3226 – Makine Öğrenmesi
Outline

- What is Learning?
- Types of Learning
  - Supervised (Gözetimli) Learning
  - Unsupervised (Gözetimsiz) Learning
  - Reinforcement (Destekli) Learning
Learning process:
Learner (a computer program) processes data $D$ representing past experiences and tries to either
- develop an appropriate response to future data, or
- describe in some meaningful way the data seen.

Example:
Learner sees a set of patient cases (patient records) with corresponding diagnoses. It can either try:
- to predict the presence of a disease for future patients
- describe the dependencies between diseases, symptoms
Defining the Learning Task

**Improve on task, \( T \), with respect to performance metric, \( P \), based on experience, \( E \).**

**Example 1:** Learning to play checkers
- \( T \): Playing checkers
- \( P \): Percentage of games won against an arbitrary opponent
- \( E \): Playing practice games against itself

**Example 2:** Learning to recognize
- \( T \): Recognizing hand-written words
- \( P \): Percentage of words correctly classified
- \( E \): Database of human-labeled images of handwritten words

**Example 3:** Learning to categorize
- \( T \): Categorize email messages as spam or legitimate.
- \( P \): Percentage of email messages correctly classified.
- \( E \): Database of emails, some with human-given labels

**Example 4:** Learning to drive
- \( T \): Driving on four-lane highways using vision sensors
- \( P \): Average distance traveled before a human-judged error
- \( E \): A sequence of images and steering commands recorded while observing a human driver.
Learning Types

- **Supervised Learning**
  - Learning mapping between *input x and desired output y*
  - Teacher gives me y’s for the learning purposes

- **Unsupervised Learning**
  - *No specific outputs* given by a teacher
  - Learning relations between data components

- **Reinforcement Learning**
  - Learning mapping between *input x and desired output y*
  - Critic does not give me y’s but instead a signal of how good my answer was (for example true or false)

- **Other types of learning**
  - Learning associations, concept learning, explanation-based learning, etc.
Learning Associations

- Basket analysis:
  - $P( Y | X )$ probability that somebody who buys $X$ also buys $Y$, where $X$ and $Y$ are products/services.
  - Example: $P( \text{chips} | \text{beer} ) = 0.7$

- If we know more about customers:
  - $P( X | Y, D )$
    - where $D$ is the customer profile (age, gender, marital status, …)
  - In case of a Web portal, items correspond to links to be shown in advance
Learning Types

Supervised Classification = Classification
⇒ We know the class labels and the number of classes

Unsupervised Classification = Clustering
⇒ We do not know the class labels and may not know the number of classes
Supervised vs. Unsupervised Learning

- **Supervised Learning**: Class labels are known
- **Unsupervised Learning**: Class labels are unknown

![Diagram](chart.png)

*Fig. 1 Difference between supervised and unsupervised learning*
Learning Types

- Supervised Learning with a Teacher
- Reinforcement Learning
- Unsupervised Learning
Supervised vs. Unsupervised Learning

Supervised Learning Workflow:
- Raw Data
- Scaled Data
- Training Set
- Validation Set
- Build Model
- Analyze and Tune
- Validate
- New Data
- Profit!

Unsupervised Learning Workflow:
- Raw Data
- Scaled Data
- Build Model
- Analyze and Tune
- Validate
- Apply Model to either input data, or new data
- Profit!
Supervised Learning

**Data:** \( D = \{d_1, d_2, \ldots, d_n\} \) \hspace{1cm} a set of \( n \) examples
\[ d_i = \langle x_i, y_i \rangle \]
\( x_i \) is input vector, and \( y \) is desired output (given by a teacher)

**Objective:** learn the mapping \( f : X \rightarrow Y \)
\[ s.t. \quad y_i \approx f(x_i) \quad \text{for all} \quad i = 1, \ldots, n \]

**Two types of problems:**
- **Regression:** \( X \) discrete or continuous \( \rightarrow \)
  \( Y \) is **continuous**
- **Classification:** \( X \) discrete or continuous \( \rightarrow \)
  \( Y \) is **discrete**
Supervised Learning Examples

● **Classification**
  - Example: Credit scoring
    - The bank may calculate the risk given the amount of credit and the information about the customer.

● **Regression**
  - Example: A system for predicting the price of a used car
    - Input: car attributes such as brand, year, ..., etc.
    - Output: the price of the car.

\[ f : X \rightarrow Y \]

**Regression:** X discrete or continuous \( \rightarrow \) Y is **continuous**

**Classification:** X discrete or continuous \( \rightarrow \) Y is **discrete**
Supervised Learning: Two Steps

- **Learning (training)**
  - from a **training set** of examples to newly learnt knowledge

- **Validation and Application**
  - **test** knowledge is checked,
    - if necessary, additional training is given
    - if not necessary, new situation is applied
Supervised Learning: Two Steps

- **Learning (training):** Learn a model using the training data
- **Testing:** Test the model using test data to assess the model accuracy

\[ \text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}} \]
Output of Training

- Tree

- Classification rules
  
  - if example satisfies condition
  - then assign it to class X

- Weight values in Neural Network

- Probabilities in Naive Bayesian

- Hyperplane
Example Problem

- Making prediction about “neighbour will go to work by walk or drive”.
Example Problem

- Making prediction about “neighbour will go to work by walk or drive”.
- Inputs:
  - Temperature (Sıcaklık)
  - Precipitation (Yağış)
  - The day of the week (Haftanın hangi günü)
  - She/he made shopping when returning back (Alışveriş yaptığı güne denk geliyor mu?)
  - Type of clothes (casual or formal) (Nasıl giyinmiş?)
<table>
<thead>
<tr>
<th>temp</th>
<th>precip</th>
<th>day</th>
<th>shop</th>
<th>clothes</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>19</td>
<td>snow</td>
<td>mon</td>
<td>yes</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>65</td>
<td>none</td>
<td>tues</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>19</td>
<td>snow</td>
<td>mon</td>
<td>yes</td>
<td>casual</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>temp</th>
<th>precip</th>
<th>day</th>
<th>shop</th>
<th>clothes</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>19</td>
<td>snow</td>
<td>mon</td>
<td>yes</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>65</td>
<td>none</td>
<td>tues</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>19</td>
<td>snow</td>
<td>mon</td>
<td>yes</td>
<td>casual</td>
<td>drive</td>
</tr>
</tbody>
</table>
## Prediction 2

<table>
<thead>
<tr>
<th>temp</th>
<th>precip</th>
<th>day</th>
<th>shop</th>
<th>clothes</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
</tbody>
</table>

```
<table>
<thead>
<tr>
<th>temp</th>
<th>precip</th>
<th>day</th>
<th>shop</th>
<th>clothes</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
</tbody>
</table>
```
Prediction 3

<table>
<thead>
<tr>
<th>temp</th>
<th>precip</th>
<th>day</th>
<th>shop</th>
<th>clothes</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>82</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>78</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>21</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>18</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>19</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>formal</td>
<td>drive</td>
</tr>
<tr>
<td>17</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>20</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>temp</th>
<th>precip</th>
<th>day</th>
<th>shop</th>
<th>clothes</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>82</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>78</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>walk</td>
</tr>
<tr>
<td>21</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>18</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>19</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>formal</td>
<td>drive</td>
</tr>
<tr>
<td>17</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
<tr>
<td>20</td>
<td>none</td>
<td>sat</td>
<td>no</td>
<td>casual</td>
<td>drive</td>
</tr>
</tbody>
</table>
Alternatives:
1. Drive because today is Monday.
2. Walk because it is raining.
3. Walk when we consider temperature and clothes.
4. …
Decision Tree

- precip
  - none
  - rain
    - clothes
      - formal
        - drive
      - casual
    - walk
  - snow
    - drive

- shop?
  - yes
    - weekend?
      - yes
        - drive
      - no
        - walk
  - no
    - temp > 90?
      - yes
        - drive
      - no
        - walk

Unsupervised Learning

- Learning “what normally happens”
- No output (we do not know the right answer)
Unsupervised Learning

- **Data:** \( D = \{d_1, d_2, ..., d_n\} \)
  \[ d_i = x_i \quad \text{vector of values} \]
  No target value (output) \( y \)

- **Objective:**
  - learn relations between samples, components of samples

**Types of problems:**
- **Clustering**
  Group together “similar” examples, e.g. patient cases
- **Density estimation**
  - Model probabilistically the population of samples
Unsupervised Learning

- *Clustering*: A kind of unsupervised Learning
- Grouping similar instances
Example Clustering Applications

- Customer segmentation in CRM
- Image clustering
- Document Clustering
## Distance Functions

### Continuous Data

**Euclidean distance**

\[ d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \]

**Hamming distance**

\[ d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \]

**Tchebyschev distance**

\[ d(x, y) = \max_{i=1,2,\ldots,n} |x_i - y_i| \]

**Minkowski distance**

\[ d(x, y) = \left( \sum_{i=1}^{n} (x_i - y_i)^p \right)^{1/p}, \quad p > 0 \]

**Canberra distance**

\[ d(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{x_i + y_i}, \quad \text{x_i and y_i are positive} \]

**Angular separation**

\[ d(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\left( \sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2 \right)^{1/2}} \]
Reinforcement Learning (RL)

- We want to learn: $f : X \rightarrow Y$
- We see samples of $x$ but not $y$
- Instead of $y$ we get a feedback (reinforcement) from a critic about how good our output was
Reinforcement Learning (RL)

- **Learning a policy**: A sequence of actions/outputs
- The goal is to select outputs that lead to the best reinforcement
- Learn through experience from **trial and error**
- Just provides an indication of **success** or **failure**.
Learning to ride a bicycle:

- The goal given to the Reinforcement Learning (RL) system is simply to ride the bicycle without falling over.
- Begins riding the bicycle and performs a series of actions that result in the bicycle being tilted **45 degrees** to the right.
- RL system turns the handle bars to the **LEFT**
  - Result: CRASH!!!
  - Receives negative reinforcement
- RL system turns the handle bars to the **RIGHT**
  - Result: CRASH!!!
  - Receives negative reinforcement
- RL system has learned that the “state” of being titled 45 degrees to the right is bad.
- Repeat trial using **40 degree** to the right.
- By performing enough of these **trial-and-error interactions** with the environment, the RL system will ultimately learn how to prevent the bicycle from ever falling over.
Applications of RL

- Checker’s Game
- Learning to play backgammon
- Inventory management
- Job-shop scheduling
- Robotic manipulation
- Robot arm control
- Robot in a maze (Path planning)
- Parking
- Football (Robo-soccer)
- Tetris
- ...

Reinforcement Learning - Example

Internal state

Environment

Reward

Observation

Learning rate $\alpha$
Inverse temperature $\beta$
Discount rate $\gamma$