Machine Learning and Intelligent Systems

Supervised Learning

Maria A. Zuluaga

October 6 2023

EURECOM - Data Science Department

Table of contents

Setup

Machine Learning: Definition

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Mitchell).

Machine Learning: Definition

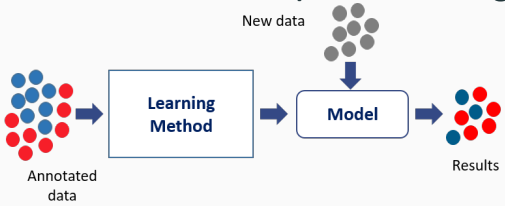
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Mitchell).

Goal

Starting from Tom Mitchell's definition, we will formalize the supervised learning setup

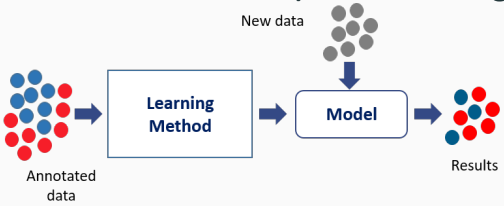
Supervised Learning: Procedure

Condensed View of Supervised Learning:

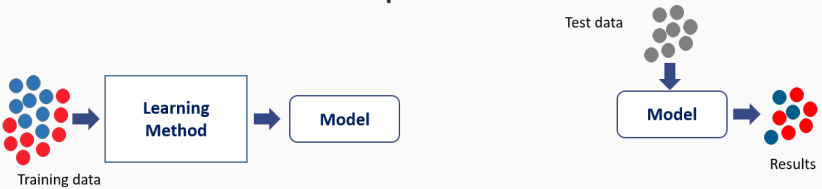


Supervised Learning: Procedure

Condensed View of Supervised Learning:



Decompressed View:



Training Phase **Testing Phase**

Training Data

The training data comes in input pairs $(\mathbf{x}, y)$, with $\mathbf{x} \in \mathbb{R}^D$ and $y \in \mathcal{C}$.

The entire training set is denoted as:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \subseteq \mathbb{R}^D \times \mathcal{C}$$

with

- $\mathbb{R}^D$ - D-dimensional feature space
- $\mathcal{C}$ - label space
- $\mathbf{x}_i$ - input vector of the i^{th} training sample
- y_i - label of the i^{th} training sample
- N - number of training samples

Question: In the previous slide, what is $\mathbf{x}$? and y?

Training Data

The **training set** points $(\mathbf{x}_i, y_i)$ are drawn from an unknown probability distribution $\mathcal{P}(X, Y)$.

Training Data

The **training set** points $(\mathbf{x}_i, y_i)$ are drawn from an unknown probability distribution $\mathcal{P}(X, Y)$.

Goal of Supervised Learning:

Use $\mathcal{D}$ to learn a function h, such that for an **unseen point** $(\mathbf{x}, y) \sim \mathcal{P}$:

$$h(\mathbf{x}) \approx y$$

with high probability

Training Data

The **training set** points $(\mathbf{x}_i, y_i)$ are drawn from an unknown probability distribution $\mathcal{P}(X, Y)$.

Goal of Supervised Learning:

Use $\mathcal{D}$ to learn a function h, such that for an **unseen point** $(\mathbf{x}, y) \sim \mathcal{P}$:

$$h(\mathbf{x}) \approx y$$

with high probability

Goal of this course (75%):

To present different methods to obtain h

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification

$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification
$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red $(1/+1)$, blue $(0/-1)$

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification
$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red $(1/+1)$, blue $(0/-1)$
2) Spam filtering (how?)

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification

$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red $(1/+1)$, blue $(0/-1)$
2) Spam filtering (how?)

Multi-class Classification

$\mathcal{C} = \{1, 2, \ldots, K\}$ with
$K \geq 2$

Example: Fruit classification from photos
(how?)

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification
$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red $(1/+1)$, blue $(0/-1)$
2) Spam filtering (how?)

Multi-class Classification
$\mathcal{C} = \{1, 2, \ldots, K\}$ with $K \geq 2$

Example: Fruit classification from photos (how?)

Regression
$\mathcal{C} = \mathbb{R}^O$
In this course, $O = 1$

Example: Predict MALIS grades $(O = 1)$

The Output Space $\mathcal{C}$

- $y \in \mathcal{C}$: Output, Target, Label, Dependent Variable.
- The output or label space $\mathcal{C}$ can take different forms.
- Depending on this, we use a specific term to refer to the supervised learning task

Binary Classification
$\mathcal{C} = \{0, 1\}$ or
$\mathcal{C} = \{-1, +1\}$

Example: 1) Red/blue ball labeling - red $(1/+1)$, blue $(0/-1)$
2) Spam filtering (how?)

Multi-class Classification
$\mathcal{C} = \{1, 2, \ldots, K\}$ with $K \geq 2$

Example: Fruit classification from photos (how?)

Regression
$\mathcal{C} = \mathbb{R}^O$
In this course, $O = 1$

Example: Predict MALIS grades $(O = 1)$
Predict weight and height of a person $(O = 2)$

The Feature Space

The feature vector $\mathbf{x}_i$ is a D-**dimensional vector** containing D attributes (or features) describing the i^{th} sample.

Often $\mathbf{x}$ is referred to as:

- Input
- Feature vector
- Attributes
- Independent variable

The Feature Space: Examples

MALIS students data. $\mathbf{x}_i = (x_i^1, x_i^2, x_i^3)$, with x_i^1 encoding the gender (x_i^1=0 or 1), x_i^2 the height (cm) and x_i^3 the age (years). Here $D = 3$.

It is a **dense** vector, since the number of non-zero coordinates in $\mathbf{x} \gg D$.

The Feature Space: Examples

MALIS students data. $\mathbf{x}_i = (x_i^1, x_i^2, x_i^3)$, with x_i^1 encoding the gender (x_i^1=0 or 1), x_i^2 the height (cm) and x_i^3 the age (years). Here $D = 3$.
It is a **dense** vector, since the number of non-zero coordinates in $\mathbf{x} \gg D$.

Text document. The bag-of-words encoding counts the occurrences of a dictionary word in a text. Hence, in $\mathbf{x}_i = (x_i^1, x_i^2, \ldots x_i^D)$, x_i^α is the count for the α^{th} word.
Most entries will have zeros. It is a **sparse** feature vector.

The Feature Space: Examples

MALIS students data. $\mathbf{x}_i = (x_i^1, x_i^2, x_i^3)$, with x_i^1 encoding the gender (x_i^1=0 or 1), x_i^2 the height (cm) and x_i^3 the age (years). Here $D = 3$.
It is a **dense** vector, since the number of non-zero coordinates in $\mathbf{x} \gg D$.

Text document. The bag-of-words encoding counts the occurrences of a dictionary word in a text. Hence, in $\mathbf{x}_i = (x_i^1, x_i^2, \ldots x_i^D)$, x_i^α is the count for the α^{th} word.
Most entries will have zeros. It is a **sparse** feature vector.

Medical image. The features represent the gray scale pixel values $\mathbf{x}_i = (x_i^1, x_i^2, \ldots x_i^D)$. Here, D represents the total number of pixels.
Question: Is this a sparse or a dense feature vector?

Summary: Notation

Symbol	Reads as
X	Input variable ($\mathbb{R}^D$)
$\mathbf{x}_i$	i^{th} feature vector. Observed value of X.
$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T$	Matrix of N input $D-$dimensional vectors $\mathbf{x}_i$
x_j	j^{th} element of the i^{th} input vector $\mathbf{x}_i$, i.e. x_i^j
Y	Output variable ($\mathcal{C}$)
y_i	i^{th} output label
$\mathbf{y} = (y_1, \ldots, y_N)^T$	Observed vector of outputs y_i

Table 1: Different notation for the input and output variables

Note

For regression, we will deal with $y \in \mathcal{C} = \mathbb{R}^{O=1}$

Setup: Where are we?

Training data

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Mitchell).

Classification/Regression

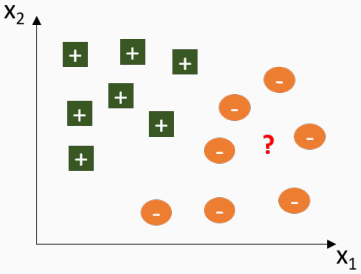
The Hypothesis Class

Recall: The goal of supervised learning is to use $\mathcal{D}$ to learn a function $h : \mathbb{R}^D \to \mathcal{C}$ that can predict y from $\mathbf{x}$.

The Hypothesis Class

Recall: The goal of supervised learning is to use $\mathcal{D}$ to learn a function $h : \mathbb{R}^D \to \mathcal{C}$ that can predict y from $\mathbf{x}$.

Example:

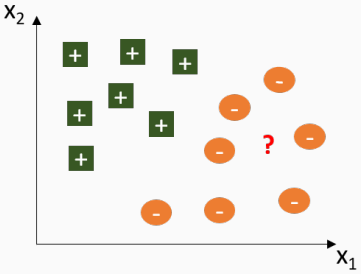


- $D =$

The Hypothesis Class

Recall: The goal of supervised learning is to use $\mathcal{D}$ to learn a function $h : \mathbb{R}^D \to \mathcal{C}$ that can predict y from $\mathbf{x}$.

Example:

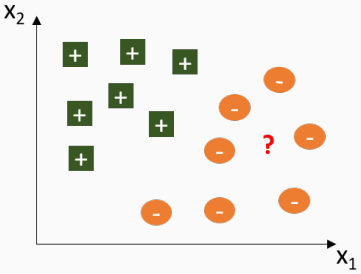


- $D=$
- $h(\mathbf{x}) = +1$
- Is this a hypothesis?

The Hypothesis Class

Recall: The goal of supervised learning is to use $\mathcal{D}$ to learn a function $h : \mathbb{R}^D \to \mathcal{C}$ that can predict y from $\mathbf{x}$.

Example:



- $D =$
- $h(\mathbf{x}) = +1$
- Is this a hypothesis?
- Is this a good hypothesis?

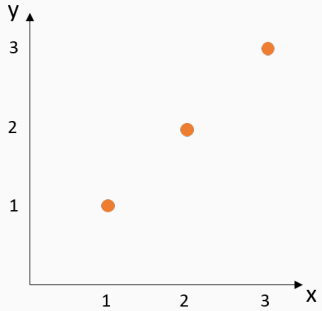
The Hypothesis Class

- We have $h \in \mathcal{H}$, where $\mathcal{H}$ denotes the hypothesis class
- **Examples:**
  - Linear Classifiers
  - Decision Trees
  - Neural Networks
  - Support Vector Machines

- First task: Pick a hypothesis class
- Warning: No Free Lunch Theorem

No Free Lunch

- Which hypothesis class $\mathcal{H}$ to choose?
- Every ML algorithm has to make assumptions
- The choice will depend on the data
- $\mathcal{H}$ encodes assumptions about the data and its distribution



$h(2.5)=?$

No Free Lunch: There is no single perfect choice for all problems

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

How?

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

How?

- **Idea:** Pick $h \in \mathcal{H}$ making the least mistakes in $\mathcal{D}$ and, preferably, the simplest.

The Loss Function

- **First task:** Pick a hypothesis class $\mathcal{H}$, i.e. pick a type of machine learning algorithm.
- **Second task:** Find the best function within the hypothesis class, $h \in \mathcal{H}$.
- Finding the best $h \in \mathcal{H}$ using $\mathcal{D}$ is denoted the **learning process**.

How?

- **Idea:** Pick $h \in \mathcal{H}$ making the least mistakes in $\mathcal{D}$ and, preferably, the simplest.
- **Measure:** Loss function

The Loss Function

- A loss or risk function $l : \mathbb{R} \to \mathbb{R}$ quantifies how well $h(\mathbf{x})$ approximates y.

$$l(a, b)$$

The Loss Function

- A loss or risk function $l : \mathbb{R} \to \mathbb{R}$ quantifies how well $h(\mathbf{x})$ approximates y.

$$l(a, b)$$

- The lower the value of $l(y, h(\mathbf{x}))$ the better the approximation
- $l(y, y) = 0$
- Typically (but not always) $l(y, h(\mathbf{x})) \geq 0$ for all $y, h(\mathbf{x})$

The Loss Function

- A loss or risk function $l : \mathbb{R} \to \mathbb{R}$ quantifies how well $h(\mathbf{x})$ approximates y.

$$l(a, b)$$

- The lower the value of $l(y, h(\mathbf{x}))$ the better the approximation
- $l(y, y) = 0$
- Typically (but not always) $l(y, h(\mathbf{x})) \geq 0$ for all $y, h(\mathbf{x})$

Loss	Expression	Task
0/1 Loss	$l(y, h(\mathbf{x})) = \begin{cases} 1 & \text{if } h(\mathbf{x}) \neq y \\ 0 & \text{otherwise} \end{cases}$	Classification
Quadratic loss	$l(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2$	Regression
Absolute loss	$l(y, h(\mathbf{x})) = \lvert y - h(\mathbf{x}) \rvert$	Regression

Table 2: Common loss functions

Loss Minimization

- Using the training data $\mathcal{D}$, we can compute the average loss over all the data points

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} l(y_i, h(\mathbf{x}_i))$$

Loss Minimization

- Using the training data $\mathcal{D}$, we can compute the average loss over all the data points

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} l(y_i, h(\mathbf{x}_i))$$

- Finding the best hypothesis means finding the h that minimizes the loss.
- This can be formalized as

$$h^* = \arg\min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} l(y_i, h(\mathbf{x}_i))$$

Generalization

Suppose the following hypothesis:

$$h(\mathbf{x}) = \begin{cases} y_i & \text{if } \exists\, (\mathbf{x}_i, y_i) \in \mathcal{D} \text{ s.t.} \mathbf{x} = \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases}$$

Generalization

Suppose the following hypothesis:

$$h(\mathbf{x}) = \begin{cases} y_i & \text{if } \exists\, (\mathbf{x}_i, y_i) \in \mathcal{D} \text{ s.t.} \mathbf{x} = \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases}$$

Questions:

- What is the value of the loss $\mathcal{L}$? Pick the loss you prefer.

Generalization

Suppose the following hypothesis:

$$h(\mathbf{x}) = \begin{cases} y_i & \text{if } \exists\,(\mathbf{x}_i, y_i) \in \mathcal{D} \text{ s.t.} \mathbf{x} = \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases}$$

Questions:

- What is the value of the loss $\mathcal{L}$? Pick the loss you prefer.
- When new samples arrive $\mathbf{x} \notin \mathcal{D}$, how will $h(\cdot)$ perform?

When $h(\cdot)$ has a very low loss, but it does not perform well in unseen data, we say there is **overfitting** causing that our model does not **generalize** well.

Generalization

Reminder: The goal is to find h such that, for an unseen point $(\mathbf{x}, y) \sim \mathcal{P}$, $h(\mathbf{x}) \approx y$.

In other words, we want h to **generalize**.

Generalization

Reminder: The goal is to find h such that, for an unseen point $(\mathbf{x}, y) \sim \mathcal{P}$, $h(\mathbf{x}) \approx y$.

In other words, we want h to **generalize**.

However, the loss over the training set does not give us information about the generalization capabilities of the trained model.

Generalization

Reminder: The goal is to find h such that, for an unseen point $(\mathbf{x}, y) \sim \mathcal{P}$, $h(\mathbf{x}) \approx y$.

In other words, we want h to **generalize**.

However, the loss over the training set does not give us information about the generalization capabilities of the trained model.

Generalization loss:

$$\epsilon = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}}[l(y, h(\mathbf{x}))]$$

We can resort to data splitting to obtain an estimate of the generalization loss.

Train/Test Splits

- We split $\mathcal{D}$ into three sets:
  - Training set $\mathcal{D}_{TR}$ - Used to learn h
  - Validation set $\mathcal{D}_{VAL}$ - To check for overfitting
  - Test set $\mathcal{D}_{TEST}$ - Used to evaluate the chosen h and have an estimate of the **generalization error** or loss

- Typical splits are 70/10/20, 80/10/10, 60/20/20.
- If the samples are drawn i.i.d. from the same distribution P, then the testing loss is an unbiased estimator of the true generalization loss.

Train/Test Splits

- It is important to split the data properly to simulate a real life scenario and to avoid **data leakage**.

- How to split?
  - **By time:** if the data is collected temporally, the split needs to be done in time. Example: First 70% point will be for training, next 10% for validation, last 20% for test.
  - **Uniformly at random** if the data is independent and identically distributed

Summary: Supervised Learning Formalization

Back to the Definition

hypothesis h

Training data

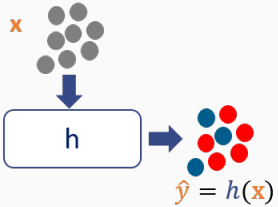
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E (Tom M. Mitchell).

loss

Classification/Regression

Back to the Supervised Learning Process

Training Phase



Testing Phase

The Learning Algorithm

Given a hypothesis class $\mathcal{H}$:

1. Train a model by minimizing the training loss:

$$h^* = \arg\min_{h \in \mathcal{H}} \frac{1}{|\mathcal{D}_{TR}|} \sum_{(\mathbf{x},y) \in \mathcal{D}_{TR}} l(y, h(\mathbf{x}))$$

2. Evaluate the testing loss of the model:

$$\epsilon_{TEST} = \frac{1}{|\mathcal{D}_{TEST}|} \sum_{(\mathbf{x},y) \in \mathcal{D}_{TEST}} l(y, h(\mathbf{x}))$$

Question: As $|\mathcal{D}_{TEST}| \to \infty$, $\epsilon_{TEST} \to \epsilon$, why?

Wrap-Up

Recap

- We introduced the basic terminology used in supervised learning
- We formalized the supervised learning setup

Key Concepts

- Feature vector, attributes, input
- Label, target output
- Classification & regression
- Hypothesis class
- Loss function
- Generalization
- Overfitting
- Data splits
- Training, validation and testing data
- No Free Lunch [link]

Any questions?