AI6102: Machine Learning Methodologies & Applications

L1: Introduction

ZHANG Hanwang

hanwangzhang@ntu.edu.sg

Nanyang Technological University, Singapore

Homepage: https://mreallab.github.io/

General Information

- Instructor:
  - Hanwang Zhang
- Time/venue
  - Wednesdays 2:30 – 5:30pm
  - Venue: LT2A

General Information (cont.)

- Q&A
  - After class
  - Make an appointment via email hanwangzhang@ntu.edu.sg
  - Send me questions via email
- Course Webpage
  - AI6102 @ NTULearn (official course webpage)

Prerequisites (recommended)

- The prerequisites for taking this module is "AI6104: MATHEMATICS FOR ARTIFICIAL INTELLIGENCE"
- Alternatively, showing evidence on taking the probability and linear algebra courses in your previous studies
  - Transcript containing the related courses need to be submitted for verification
- Otherwise, taking this module next semester

Evaluation

- Individual assignment (25%)
  - Details will be released by Week 5
  - Submission deadline: 28th Sep (end of Week 7)
- Open book quiz (35%)
  - ~1 hour (in Week 13 during lecture time)
  - Scope: Lectures 2 – 11
  - Details will be released in Week 6
- Team project (40%)
  - Details will be released by Week 3
  - Submission deadline: 21 Nov (end of Week 14).

What is Machine Learning?

- Motivation: human beings can always learn from examples/experience



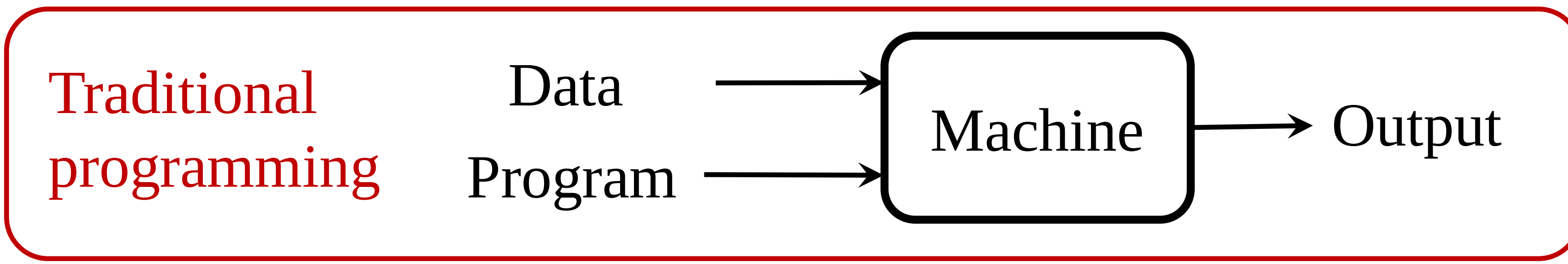


- Can machines teach themselves to grow from data and change when exposed to new data?

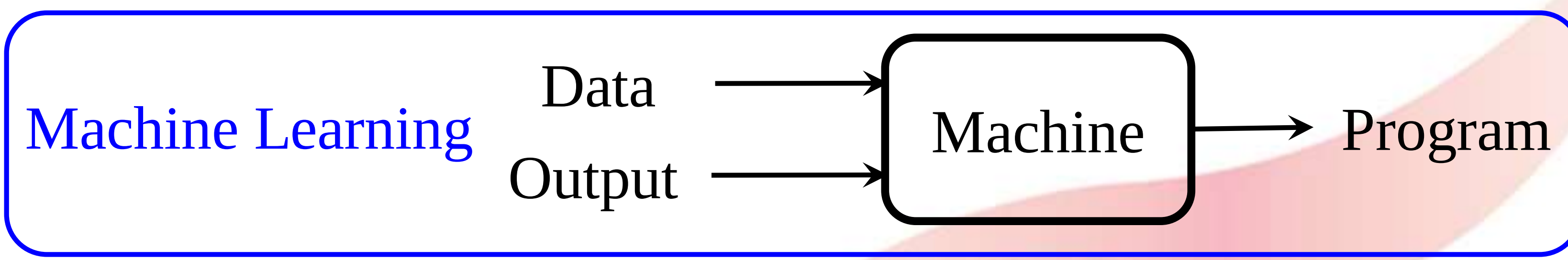



What is Machine Learning? (cont.)

- A type of artificial intelligence that provides computers with the ability to learn from examples/experience <u>without being explicitly programmed</u>



V.S.



A Motivating Example: Credit risk estimation

- Goal: to automatically evaluate the risk that a potential borrower will repay a loan or not
- Accessible information:
  - Profile of the loan applicant: name, age, gender, income, saving, marital status, profession, past financial history, etc
  - Records of past loans containing such profile data of previous borrowers and whether they finally repaid the loans or not

A Motivating Example:
Credit risk estimation

Records of past loans

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

Information of a new applicant

ID	Gender	Profession	Income	Saving	Repay
11	F	Lawyer	70k	100k	?

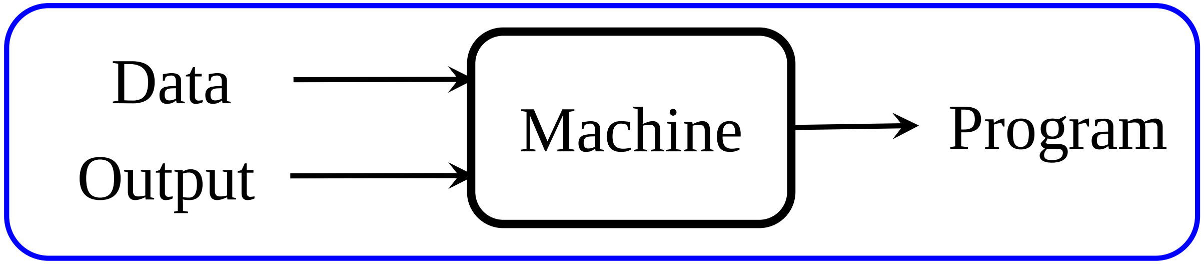
Credit Risk Estimation
A traditional approach

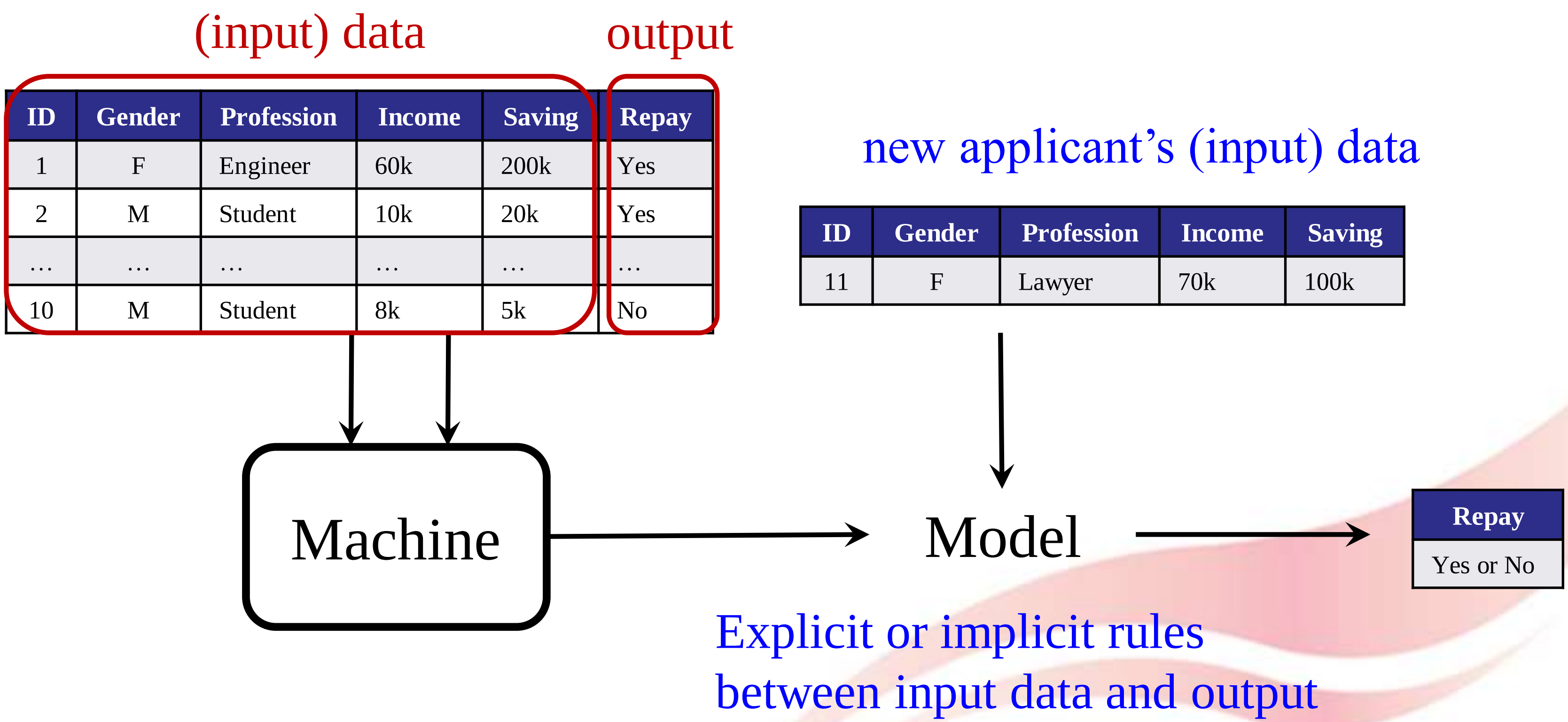
Data → Machine → Output
Program → Machine

New applicant's data → Machine → Repay= Yes OR No
Hand-crafted rules → Machine

E.g., IF income > 100k AND saving > 150k, THEN Repay = Yes

Credit Risk Estimation
A machine learning approach





ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

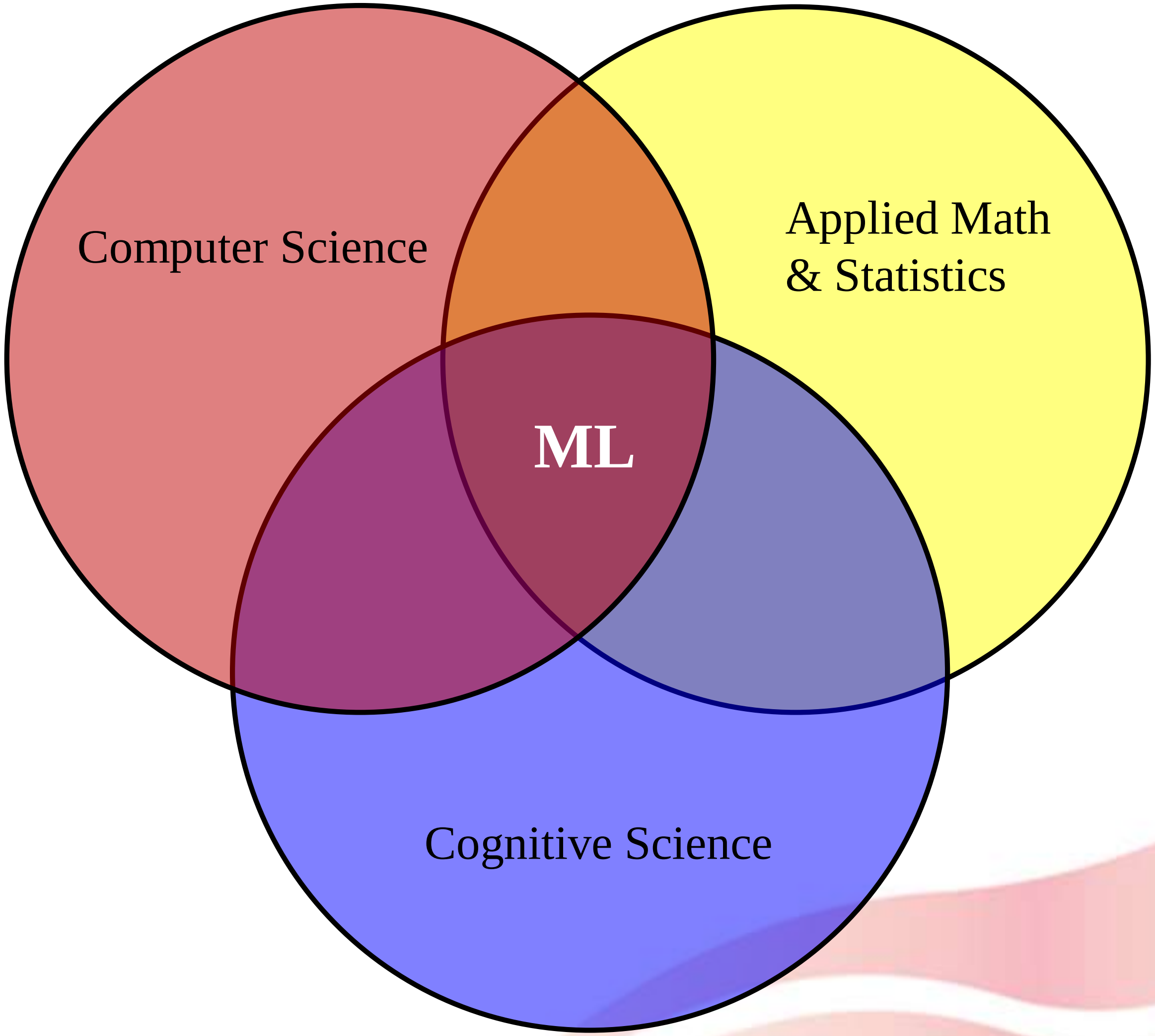
ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

Repay
Yes or No

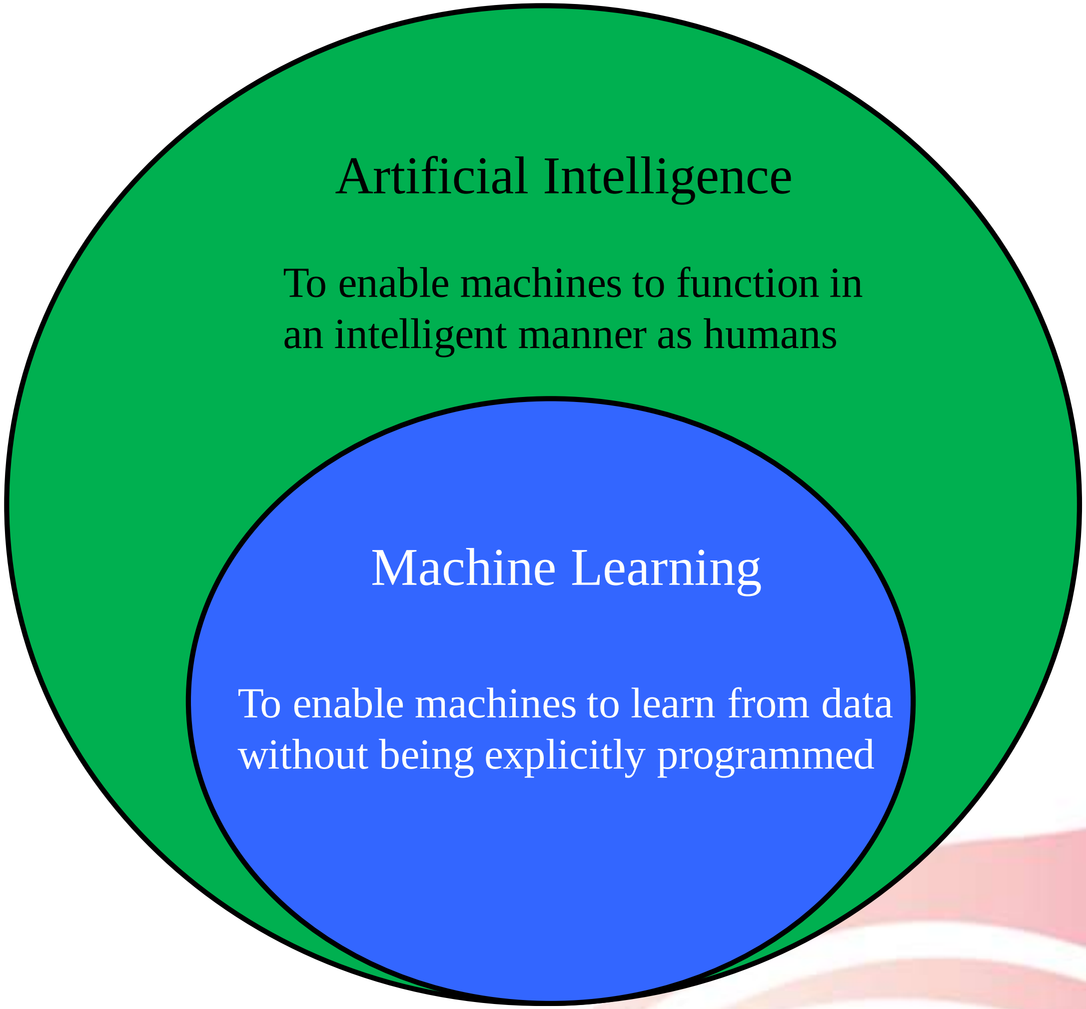
Machine Learning Characteristics

- Data driven
  - Aims to automatically learn a model from data
- Focus on the development of general algorithms to learn a model from data
  - Not focus on specific application problems

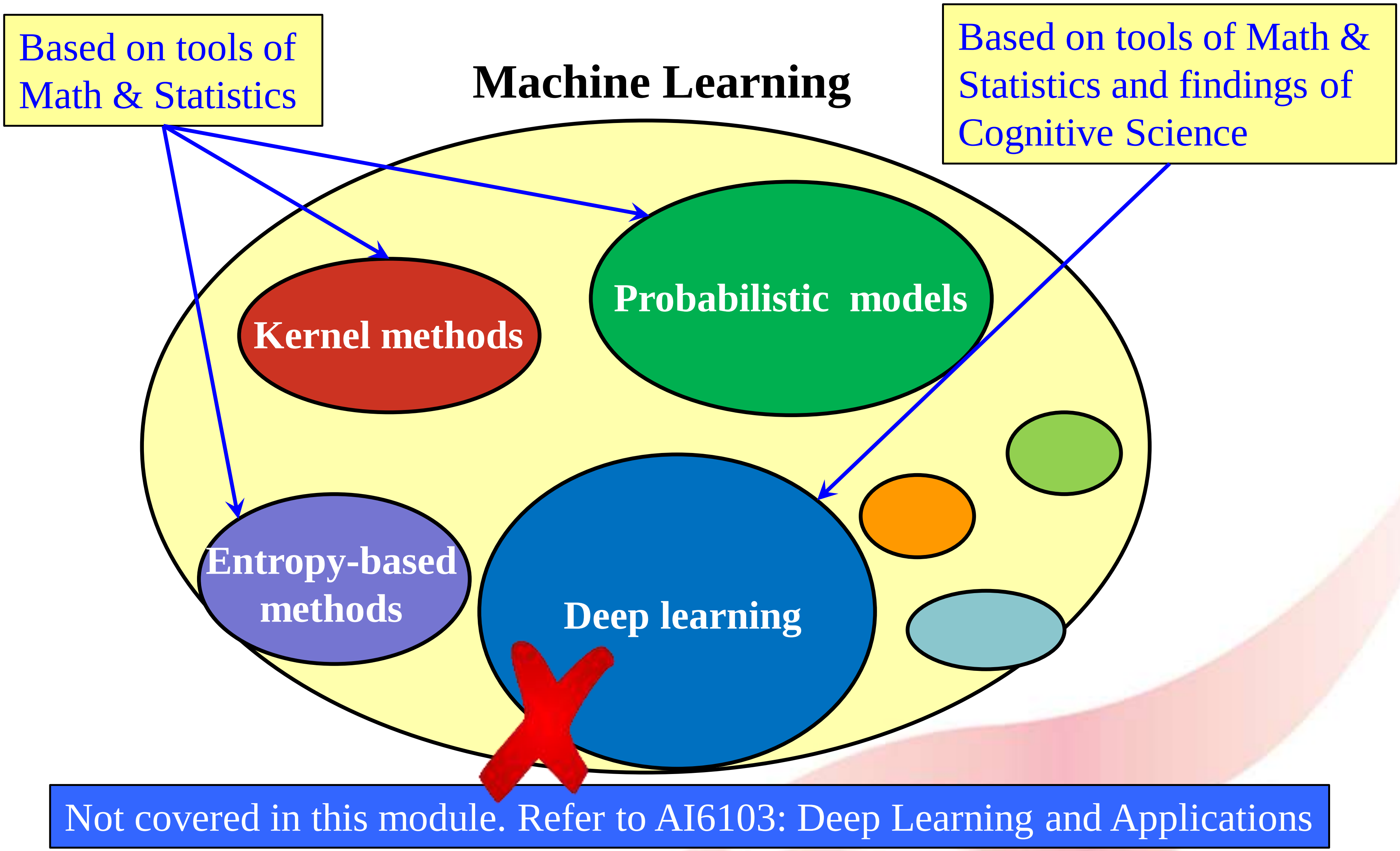
ML is Interdisciplinary



Machine Learning & AI



Machine Learning Methodologies



Different Learning Paradigms

- Three important paradigms:
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Some advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning
  - etc.

Supervised Learning

- Learning with "supervision"
- A set of examples are presented to a machine, and each example is a pair of an input data instance and a desired output

labeled data

- The goal is to "learn" a function or prediction model from inputs to outputs

supervision or label

Supervised Learning (cont.)

In mathematics

- Given: a set of $\{\boldsymbol{x}_i, y_i\}$ for $i = 1, \dots, N$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ is m-dimensional vector of numerical values, and y_i is a scalar
- Goal: to learn a mapping $f: \boldsymbol{x} \to y$ by requiring $f(\boldsymbol{x}_i) = y_i$
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$

Supervised Learning Tasks

Use discrete numerical values to represent **categories**

- Classification v.s. Regression
- If the value of each output y_i is $\boxed{\text{discrete}}$, then it is a classification task
  - Only two distinct values: binary classification
  - More than two distinct values: multi-class classification
- If the value of each output y_i is continuous, then it is a regression task

Credit Risk Estimation: Binary

(input) data — output

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

Yes: 1
No: -1

	X_1	X_2	…	X_{m-1}	X_m	Y	
$\boldsymbol{x}_1$	1	0	…	60	200	1	y_1
$\boldsymbol{x}_2$	0	1	…	10	20	1	y_2
	…	…	…	…	…	…	
$\boldsymbol{x}_{10}$	0	1	…	8	5	-1	y_{10}

new applicant's (input) data

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

	X_1	X_2	…	X_{m-1}	X_m
$\boldsymbol{x}^*$	1	0	…	70	100

$f: \boldsymbol{x} \rightarrow y$ s.t. $f(\boldsymbol{x}_i) = y_i, i = 1 \ldots, 10$

Learning

$f: \boldsymbol{x}^* \rightarrow y^*$ via $f(\boldsymbol{x}^*) = y^*$

Using

Credit Risk Estimation: Multiclass

(input) data | output

ID	Gender	Profession	Income	Saving	Repay Time
1	F	Engineer	60k	200k	On Time
2	M	Student	10k	20k	Late
…	…	…	…	…	…
10	M	Student	8k	5k	Never

new applicant's (input) data

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

	X_1	X_2	…	X_{m-1}	X_m	Y	
$\boldsymbol{x}_1$	1	0	…	60	200	1	y_1
$\boldsymbol{x}_2$	0	1	…	10	20	2	y_2
	…	…	…	…	…	…	
$\boldsymbol{x}_{10}$	0	1	…	8	5	3	y_{10}

On Time: 1
Late: 2
Never: 3

	X_1	X_2	…	X_{m-1}	X_m
$\boldsymbol{x}^*$	1	0	…	70	100

$$f: \boldsymbol{x} \rightarrow y \text{ s.t. } f(\boldsymbol{x}_i) = y_i, i = 1 \ldots, 10$$

Learning

$$f: \boldsymbol{x}^* \rightarrow y^* \text{via } f(\boldsymbol{x}^*) = y^*$$

Using

Credit Risk Estimation: Regression

(input) data output

ID	Gender	Profession	Income	Saving	Repay %
1	F	Engineer	60k	200k	100
2	M	Student	10k	20k	95
...	...	...	...	...	...
10	M	Student	8k	5k	0

new applicant's (input) data

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

	X_1	X_2	...	X_{m-1}	X_m	Y	
$\boldsymbol{x}_1$	1	0	...	60	200	100	y_1
$\boldsymbol{x}_2$	0	1	...	10	20	95	y_2
	...	...	...	...	...	...	
$\boldsymbol{x}_{10}$	0	1	...	8	5	0	y_{10}

	X_1	X_2	...	X_{m-1}	X_m
$\boldsymbol{x}^*$	1	0	...	70	100

$f: \boldsymbol{x} \to y$ s.t. $f(\boldsymbol{x}_i) = y_i, i = 1 \dots, 10$

Learning

$f: \boldsymbol{x}^* \to y^*$ via $f(\boldsymbol{x}^*) = y^*$

Using

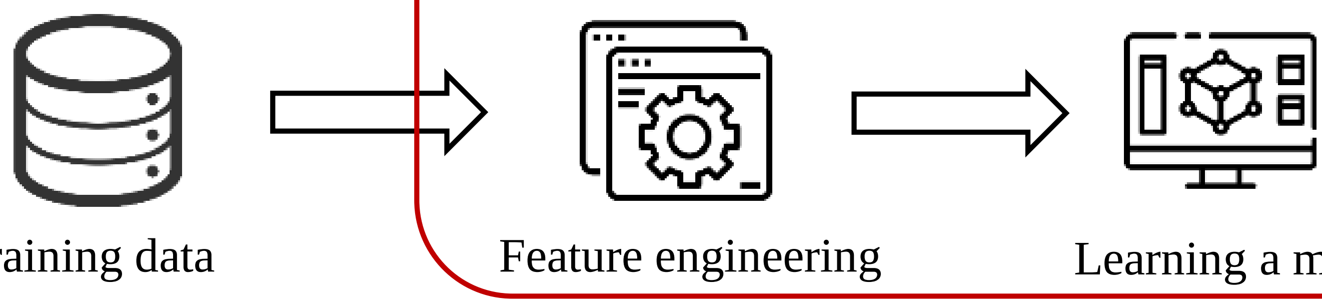
Typical Procedure of Supervised Learning

Training phase

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
...	...	...	...	...	...
10	M	Student	8k	5k	No

X_1	X_2	...	X_{m-1}	X_m	Y
1	0	...	60	200	1
0	1	...	10	20	1
...	...	...	...	...	...
0	1	...	8	5	-1

$$f: x \rightarrow y$$

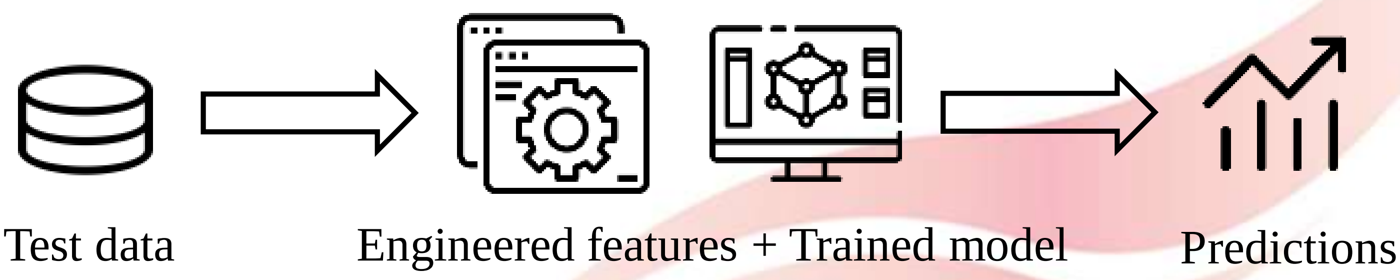


Test phase

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

X_1	X_2	...	X_{m-1}	X_m
1	0	...	70	100

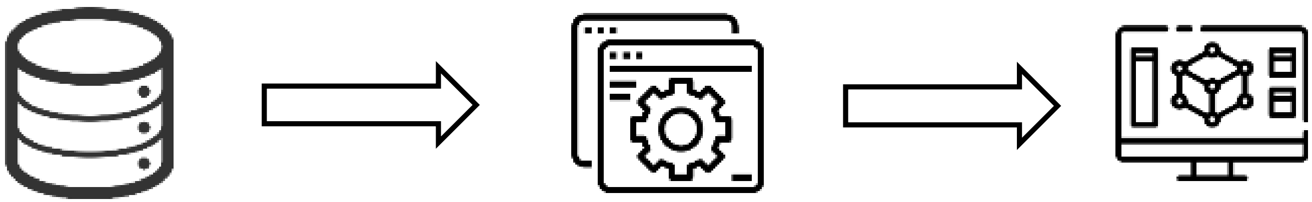
$$f(x^*)$$



Content of Supervised Learning

$\{\boldsymbol{x}_i, y_i\}$ for $i = 1, \dots, N$

$f: \boldsymbol{x} \rightarrow y$



Training data

Feature engineering

Learning a model

In this module

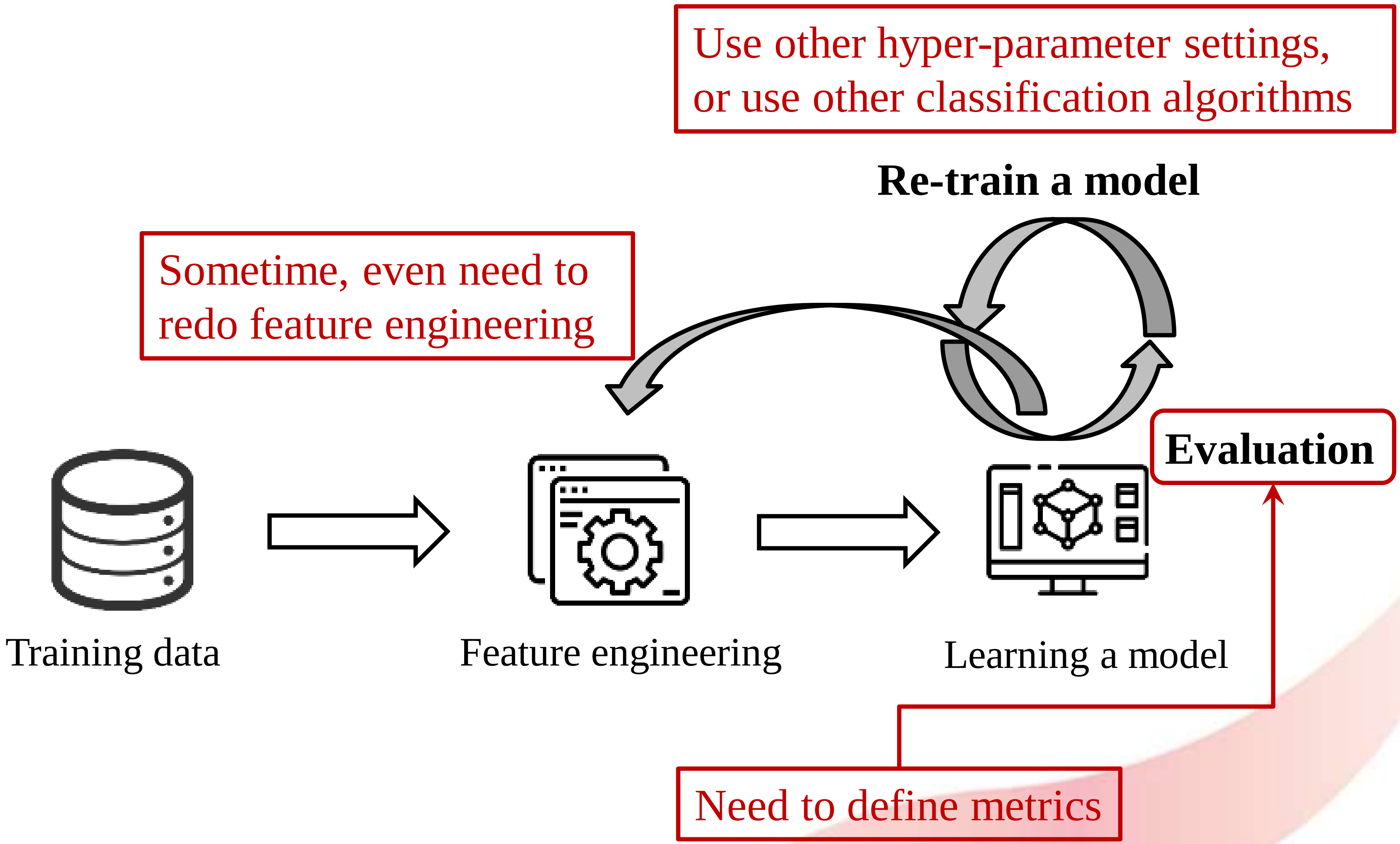
- Basic transformations to numerical vectors

Note: domain knowledge or experience is required to manually design good features

- Linear models
- Kernel methods
- Tree-based methods
- Bayesian classifiers
- KNN classifiers (a lazy classifier)

An end-to-end solution: deep learning, effective on unstructured data, like images, text, speech data, etc. (out of scope)

Loop of Training a Model



Common Performance Metrics

- Classification: accuracy or error rate

Ground-truth y_i	Prediction $f(\boldsymbol{x}_i)$	
$+1$	-1	✗
-1	-1	✓
$+1$	$+1$	✓
-1	$+1$	✗
-1	-1	✓

The rate of correct predictions

$$\text{Accuracy} = 3/5$$

The rate of incorrect predictions

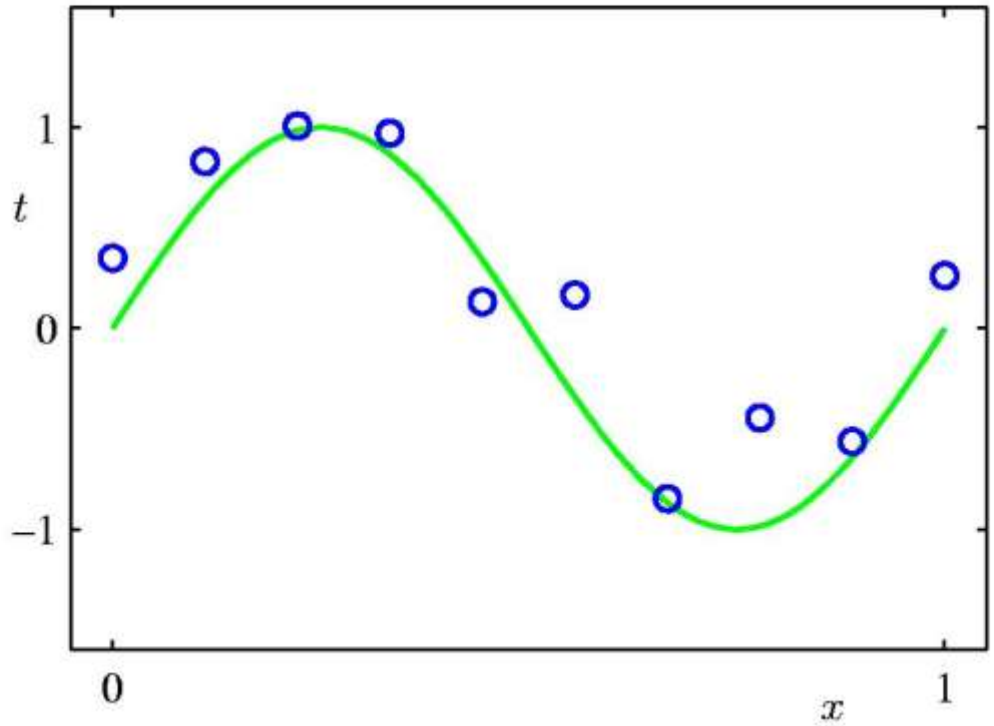
$$\text{Error rate} = 2/5$$

- Regression:
  - Mean Absolute Error (MAE): $\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|f(\boldsymbol{x}_i) - y_i|$
  - Root Mean Squared Error (RMSE): $\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2}$

Model Evaluation

- With a performance metric, how to evaluate the performance of a trained model?
- A straight-forward solution:
  - Step 1: Use the training data to train a model
  - Step 2: Apply the trained model to make predictions on the training data, and calculate the performance metric, e.g., error rate
- The model highly risky to “over-fit” the training data

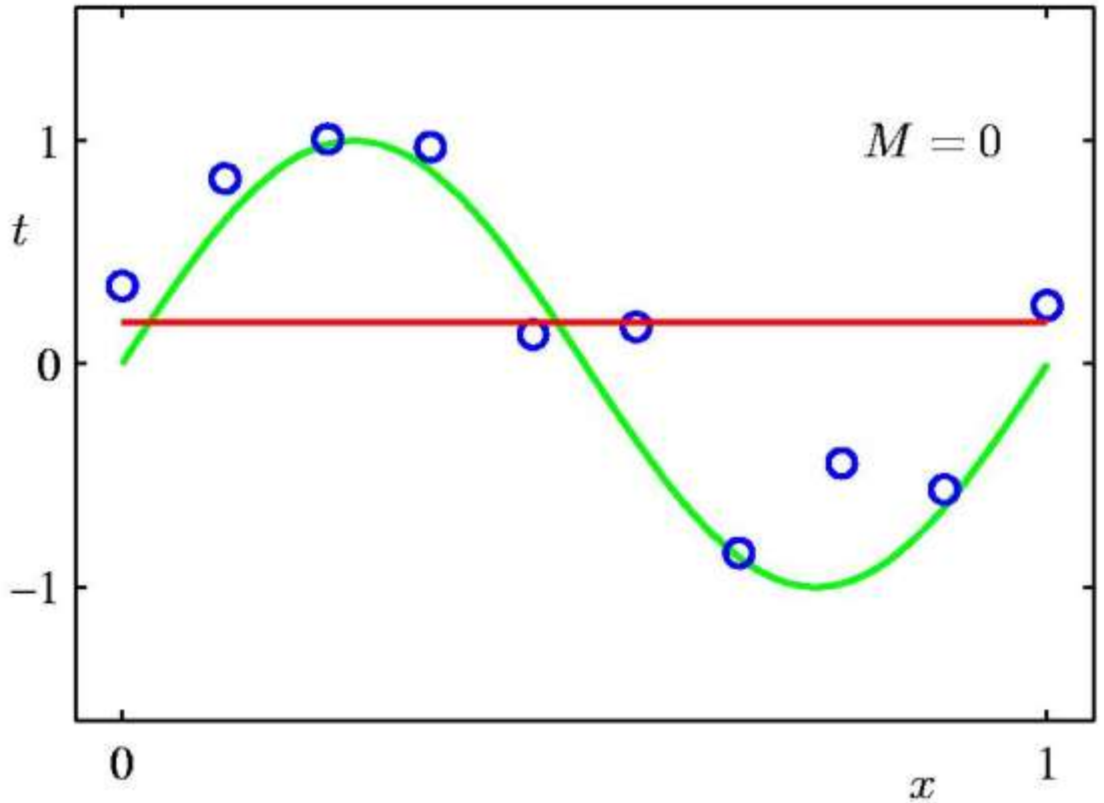
Example: Polynomial Curve Fitting



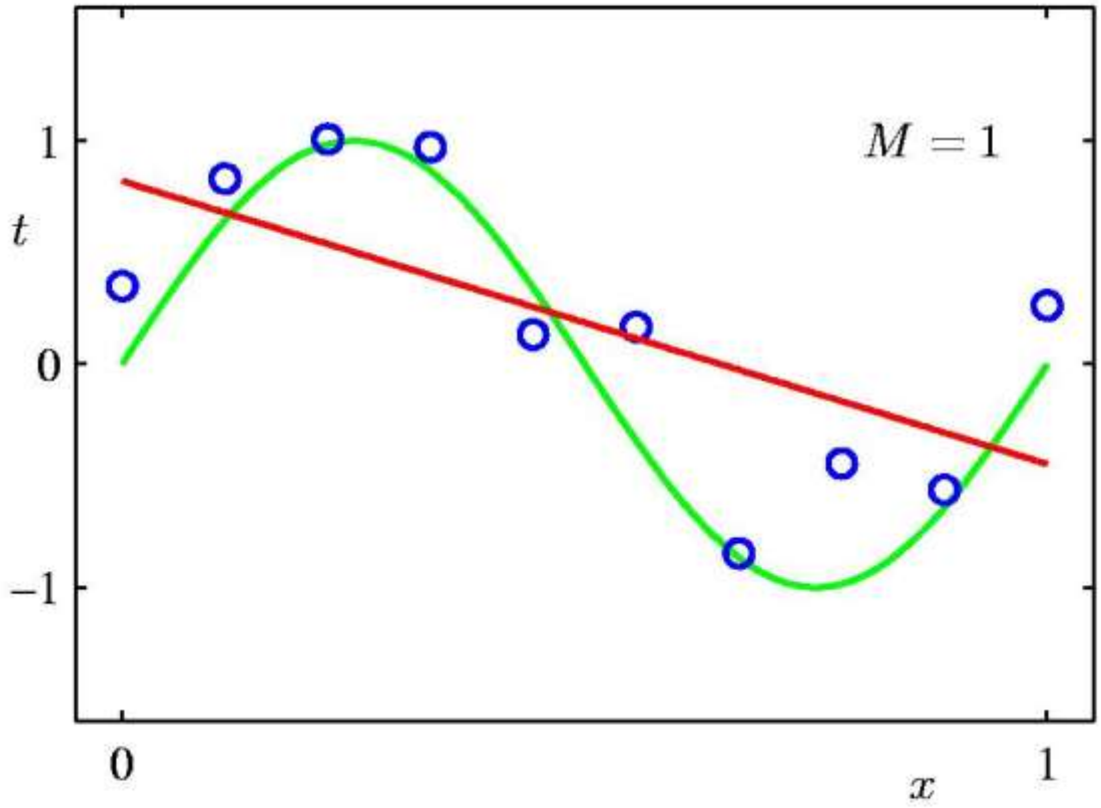
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Evaluation metric: Root Mean Squared Error (RMSE)

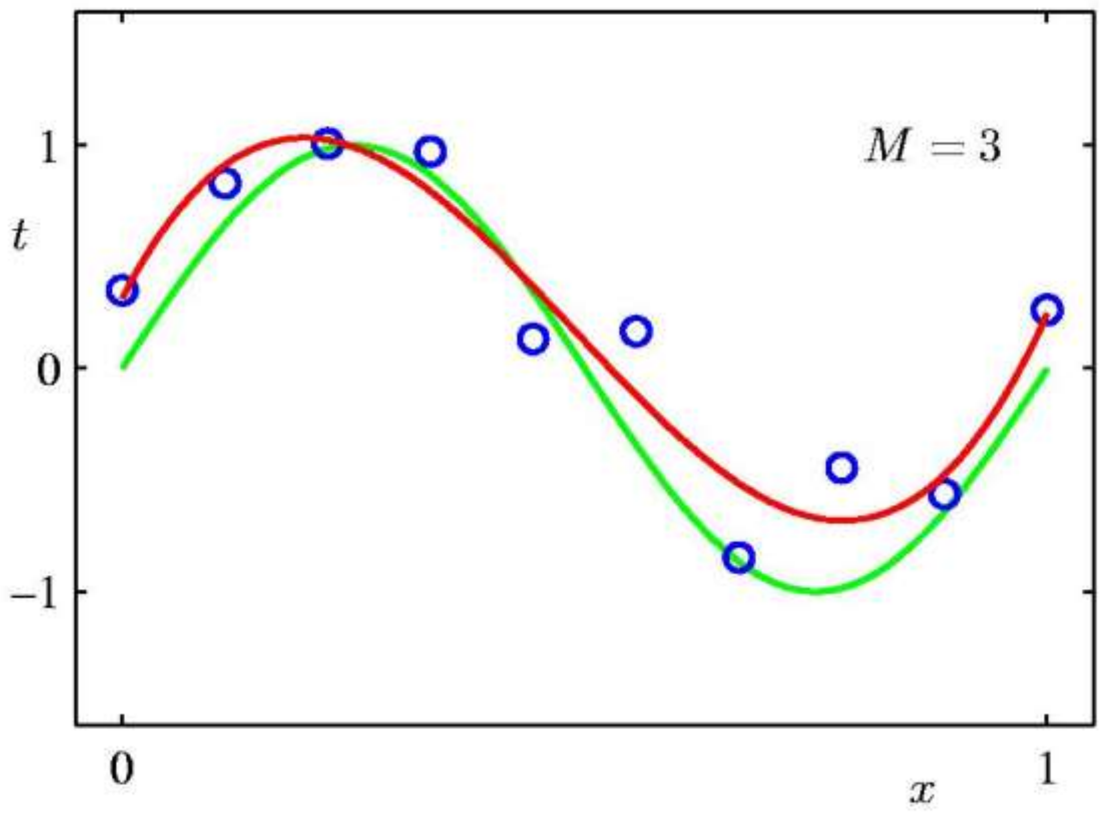
0^{th} Order Polynomial



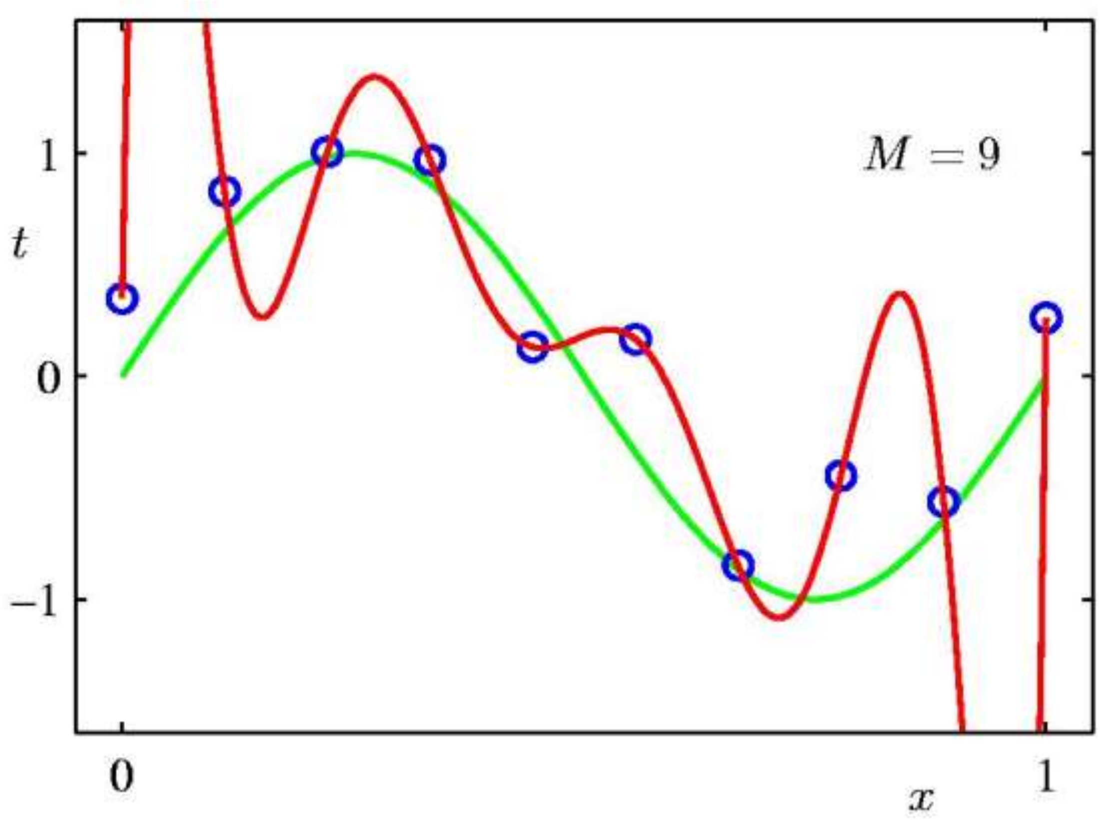
1^{st} Order Polynomial



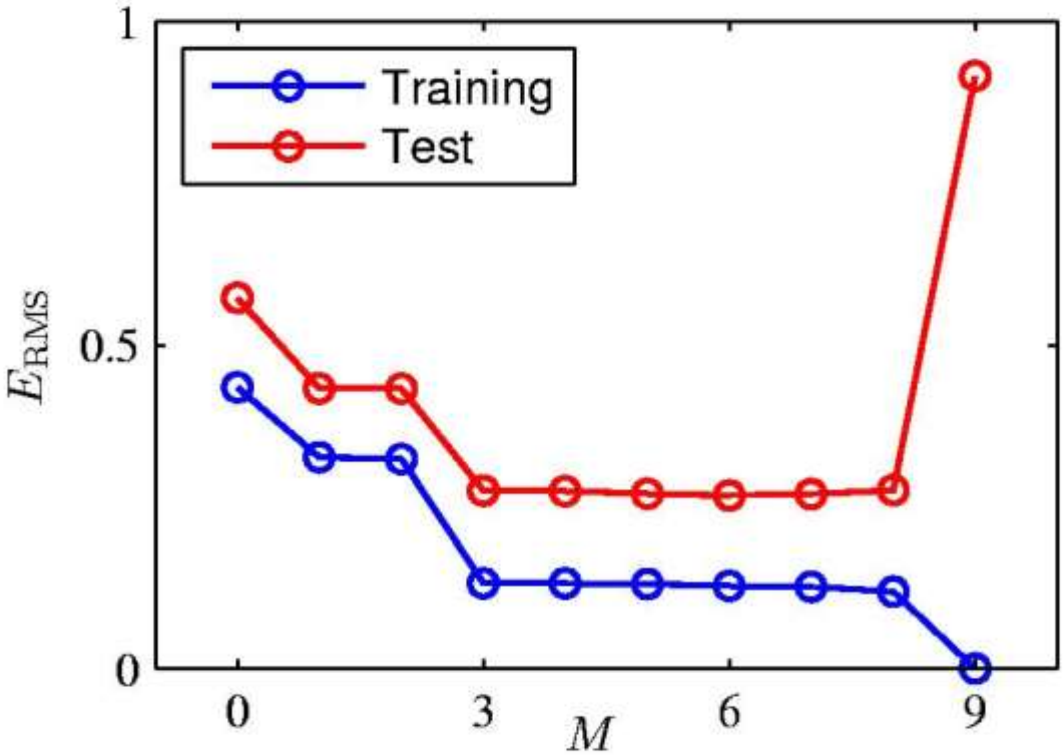
3rd Order Polynomial



9^{th} Order Polynomial



Over-fitting



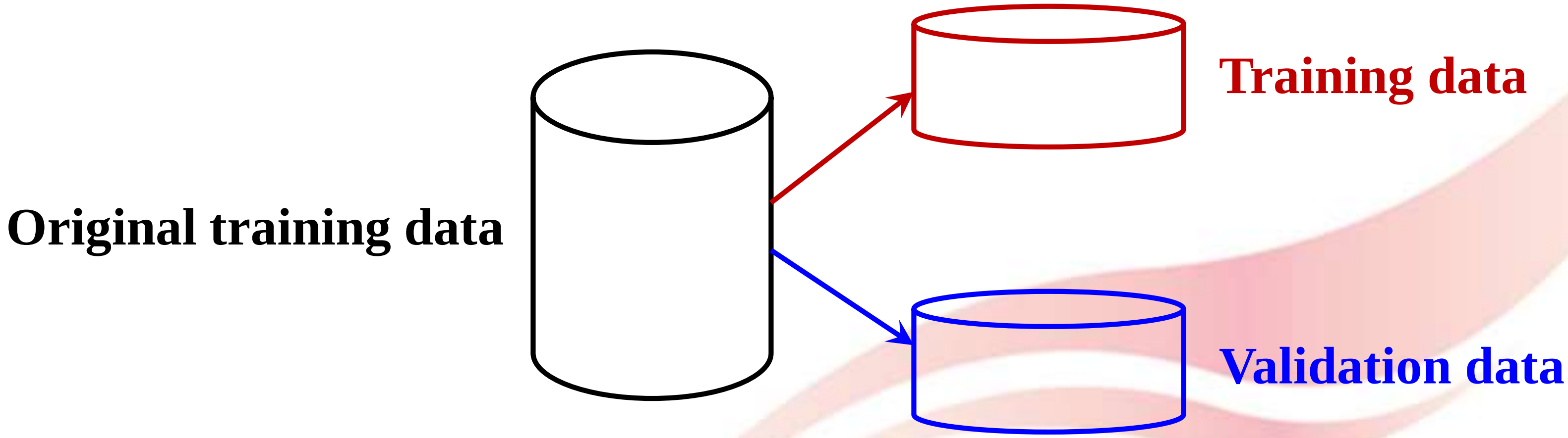
Root Mean Square Error (RMSE)

Note on Over-fitting

- Recall: the goal of supervised learning is learn a model such that it makes precise predictions on any unseen data (i.e., test data) NOT training data
- Performance calculated on training data fails to provide a good estimate of how well the model will perform on previously unseen data
- Solution: validation set

Validation Set

- Split the whole training dataset into two disjoint sets: “training” set and “validation” set
- The split “training” set is used to train a prediction model and the “validation” set is used to evaluate the performance of the trained model
- How to split the dataset?

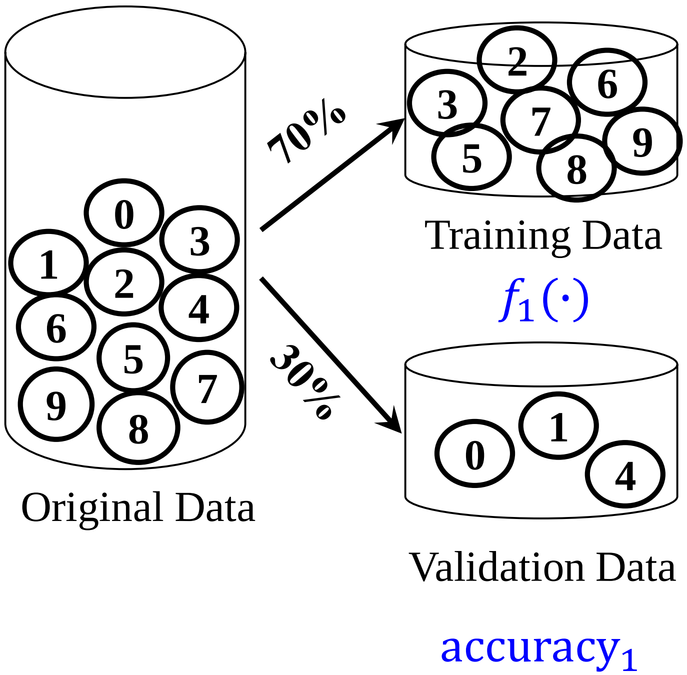


Random Subsampling

For a classification algorithm $\mathcal{A}$, set hyper-parameter $\Theta = \theta_1$

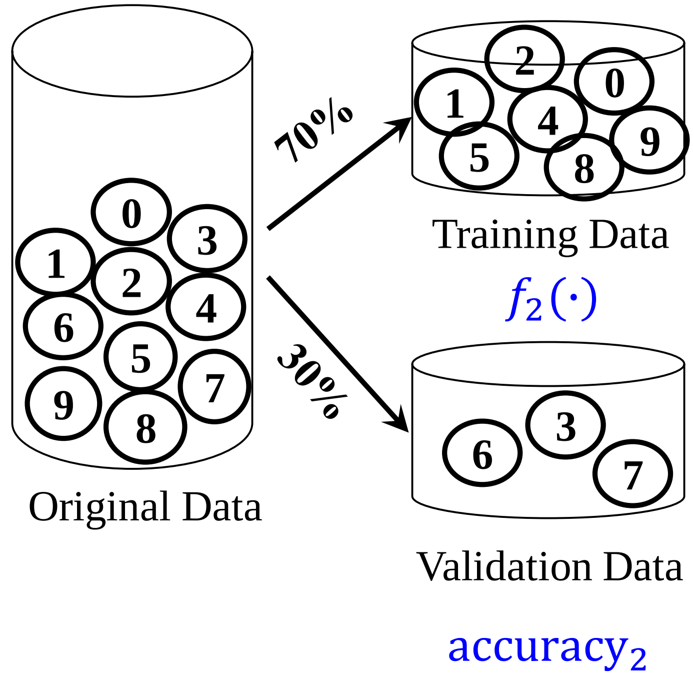
1st random sampling

(Sampled without replacement)



2nd random sampling

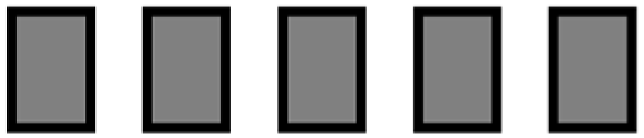
(Sampled without replacement)



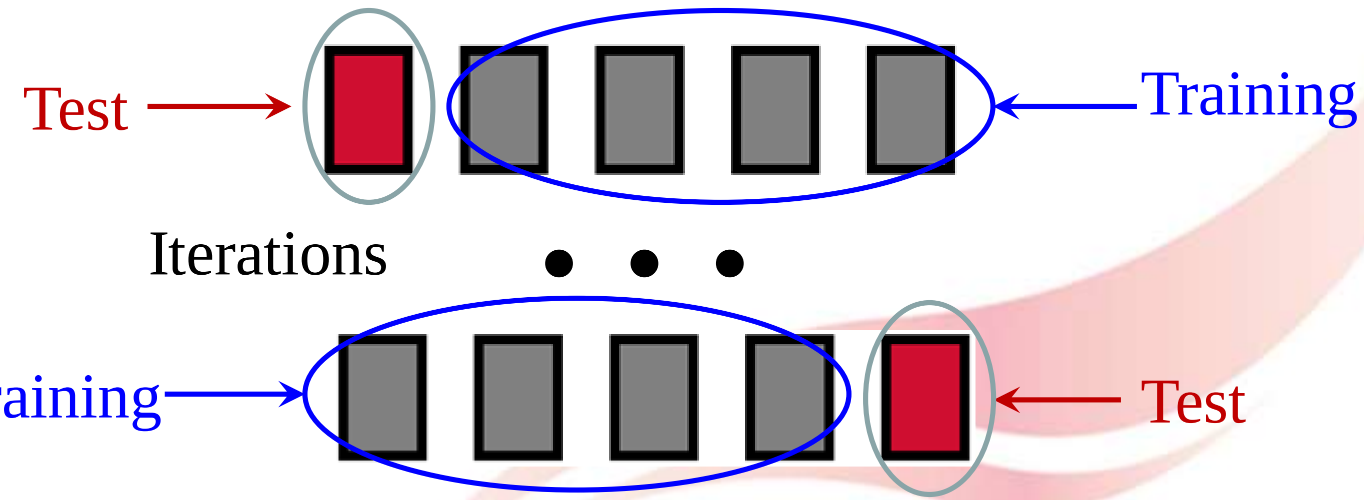
$$\text{acc} = \frac{1}{k}\sum_{i=1}^{k} \text{accuracy}_i \longrightarrow \text{Performance evaluation for } \mathcal{A} \text{ with hyper-parameter } \Theta = \theta_1$$

K-fold Cross Validation

- *k*-fold cross-validation: partition data into *k* subsets of the same size



- Hold aside one group for testing and use the rest to build model



An Example: 5-fold

For a classification algorithm $\mathcal{A}$, set hyper-parameter $\Theta = \theta_1$

Partition into 5 subsets

1	2	3	4	5
$\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	$\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	$\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	$\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	$\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}\}$

Hold aside 1 group for testing, use the rest 4 for training

1: Test $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}\}$
1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Test** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}\}$
1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Test** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}\}$
1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Test** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}\}$
1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Test** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}\}$

Evaluation, e.g., accuracy for $\mathcal{A}$, with $\Theta = \theta_1$

1: Prediction $\hat{y}_1, \hat{y}_2$	2: Prediction $\hat{y}_3, \hat{y}_4$	3: Prediction $\hat{y}_5, \hat{y}_6$	4: Prediction $\hat{y}_7, \hat{y}_8$	5: Prediction $\hat{y}_9, \hat{y}_{10}$
1: Ground-truth y_1, y_2	2: Ground-truth y_3, y_4	3: Ground-truth y_5, y_6	4: Ground-truth y_7, y_8	5: Ground-truth y_9, y_{10}

Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Unsupervised Learning

- Learning without "supervision"
- A set of examples presented to a machine only contains input data instances without desired outputs, the goal is to "learn" intrinsic structures or patterns underlying the input data instances

Unlabeled data

Unsupervised Learning (cont.)

In mathematics

- Given: a set of $\{\boldsymbol{x}_i\}$ for $i = 1, \dots, N$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ is m-dimensional vector of numerical values
- Goal: to learn a model $g: \boldsymbol{x} \rightarrow \boldsymbol{z}$, where $\boldsymbol{z}$ captures patterns or hidden structure of the $\boldsymbol{x}$

Unsupervised Learning Tasks

- Clustering, (unsupervised) dimensionality reduction, density estimation
  - Clustering: to automatically assign each input data instance $\boldsymbol{x}_i$ to a group
    - $g: \boldsymbol{x}_i \rightarrow z_i$, here z_i is the index of a group
  - Dimensionality reduction: to map each input data instance $\boldsymbol{x}_i$ to a lower-dimensional vector
    - $g: \boldsymbol{x}_i \rightarrow \boldsymbol{z}_i$, here $\boldsymbol{z}_i$ is a vector of d dimensions, $d < m$
  - Density estimation: to learn a density function
    - $g: \boldsymbol{x}_i \rightarrow z_i$, here z_i is the value of the density function at the data point $\boldsymbol{x}_i$

Clustering: User Segmentation

Common hyper-parameter of most clustering algorithms

Suppose we want to cluster potential customers into $\boxed{3}$ groups, and advertise a different loaning plan to different groups

ID	Gender	Profession	Income	Saving
1	F	Engineer	60k	200k
2	M	Student	10k	20k
…	…	…	…	…
10	M	Student	8k	5k

	X_1	X_2	…	X_{m-1}	X_m
x_1	1	0	…	60	200
x_2	0	1	…	10	20
	…	…	…	…	…
x_{10}	0	1	…	8	5

$$g: \boldsymbol{x} \rightarrow z$$

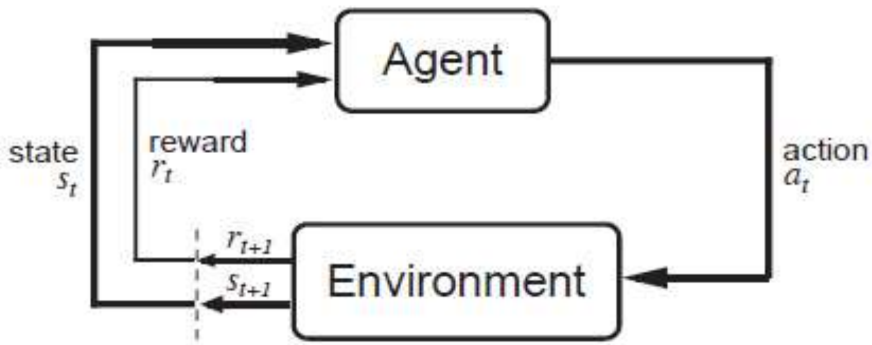
	X_1	X_2	…	X_{m-1}	X_m	Z
x_1	1	0	…	60	200	1
x_2	0	1	…	10	20	3
	…	…	…	…	…	
x_{10}	0	1	…	8	5	1

Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Reinforcement Learning

- Learning by interacting with an environment to achieve a goal $p: \boldsymbol{s}_i \rightarrow a_i$
- Objective: to learn an optimal $\boxed{\text{policy}}$ mapping states to actions

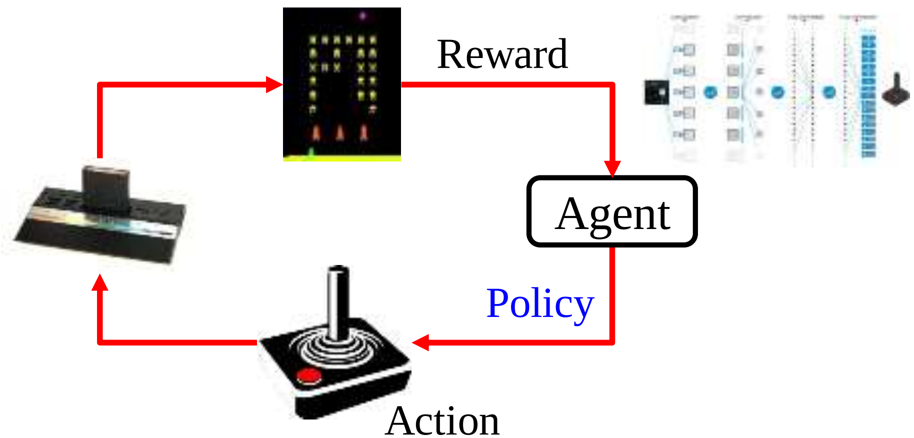


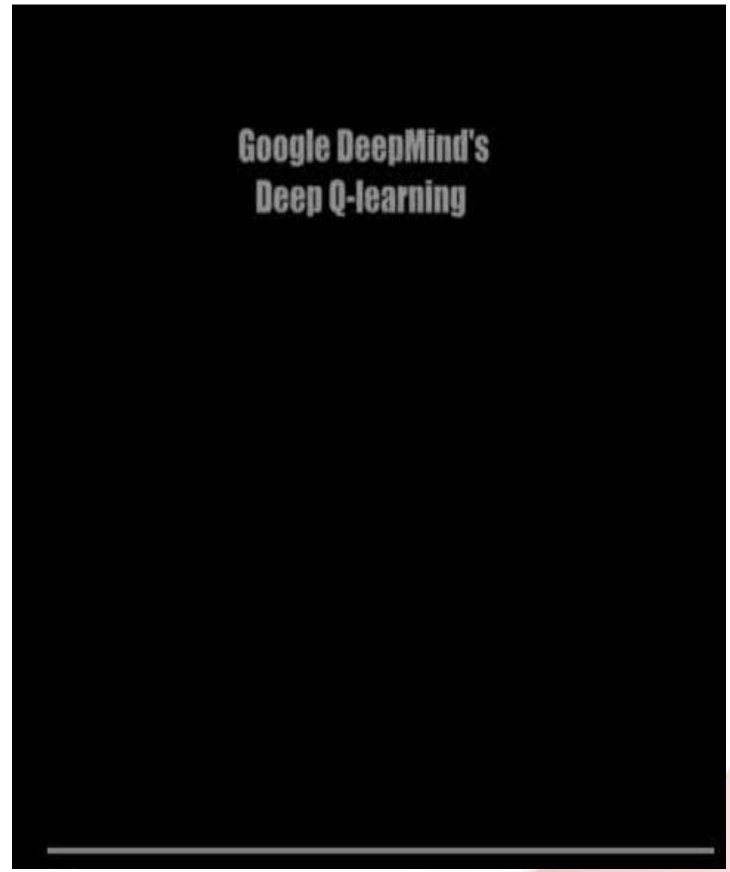
$$S_0 \xrightarrow[r_0]{a_0} S_1 \xrightarrow[r_1]{a_1} S_2 \xrightarrow[r_2]{a_2} S_3$$

- Each state S_i is represented by a feature vector $\boldsymbol{s}_i$
- Each action is represented by a discrete value (categorical)
- Each reward is a scalar (indirect supervision)

Reinforcement Learning (cont.)

- Deep Q-Network (DQN) [1]
  - Play Atari 2600 Games





[1] Mnih et al, Human-level control through deep reinforcement learning. *Nature*, 2015

Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Semi-supervised Learning

- Recall: in supervised learning, a set of labeled data is given for training. In general, sufficient labeled data set is required to train a precise prediction model
- However, in some real-world application scenarios, labeled data is in short supply while unlabeled data is easy to collect
- Semi-supervised learning aims to make use of unsupervised learning techniques with unlabeled data to help a supervised learning task with insufficient labeled data

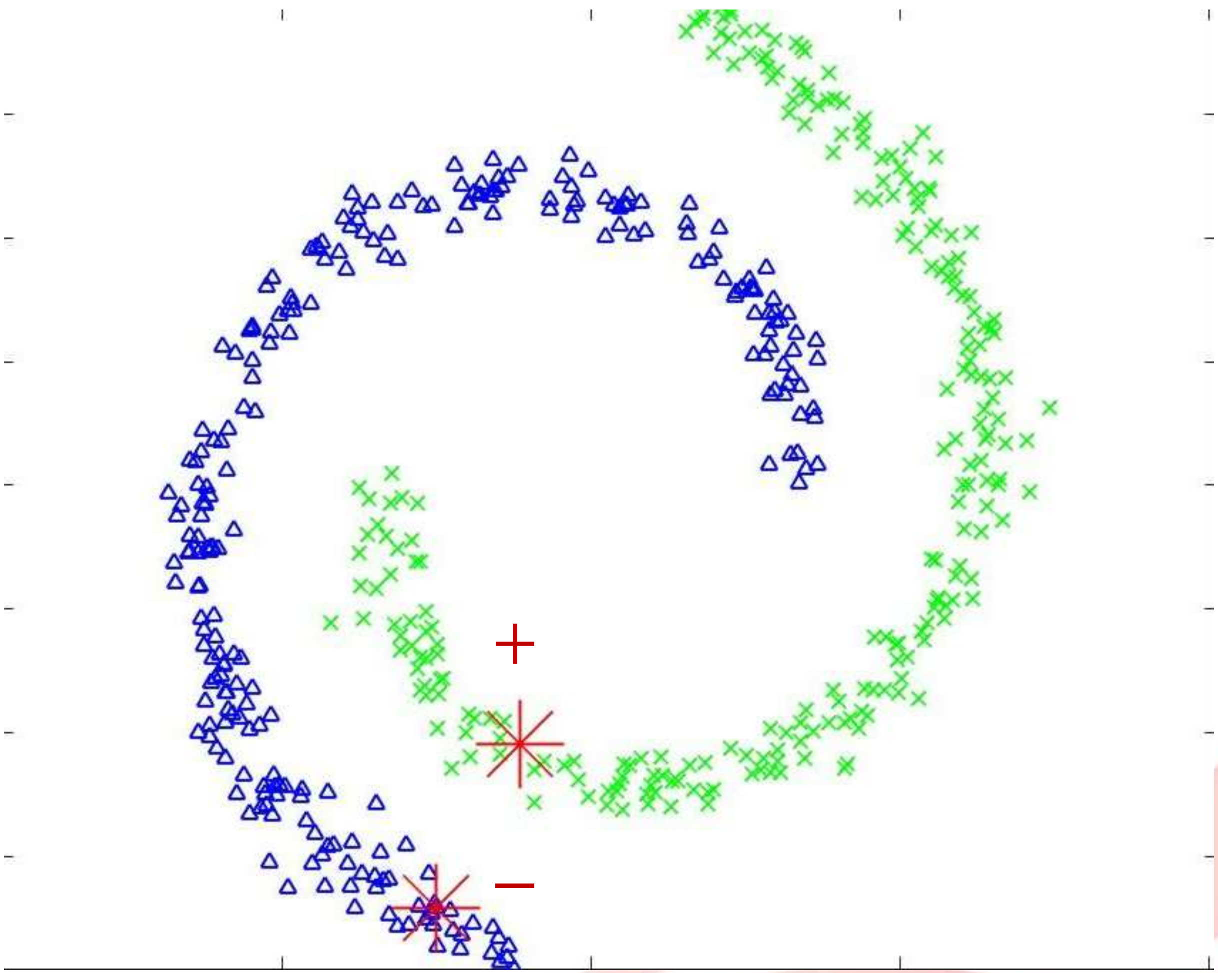
Semi-supervised Learning (cont.)

In mathematics

- Given: a small set of labeled data $\{\boldsymbol{x}_i, y_i\}$ for $i = 1, \ldots, L$, and a set of unlabeled data $\{\boldsymbol{x}_j\}$ for $j = 1, \ldots, U$. In general, $L \ll U$
- Goal: to learn a precise mapping $f: \boldsymbol{x} \to y$ by requiring $f(\boldsymbol{x}_i) = y_i$ and making using of the patterns underlying unlabeled data $\{\boldsymbol{x}_j\}$'s
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$

A Motivating Example

Two moons dataset



A Real-world Example

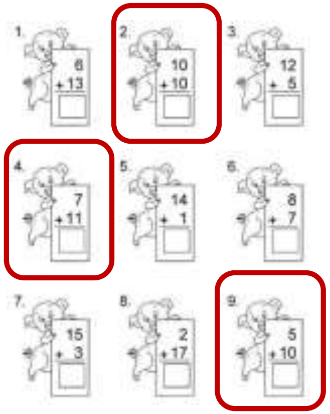
- Suppose our task is to predict whether a patient has a rare gene disease based on some symptoms and some basic tests
- Input: symptoms and basic tests results
- Output: whether the patient has the gene disease or not
- However, to get the “true” label, a gene test is needed, which is very expensive and time consuming
  - The labeled dataset available for training is small
  - Meanwhile, we may have a lot of unlabeled data, i.e., patients with symptoms and basic tests results
  - Semi-supervised learning can help!

Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Motivation

To understand a mathematical concept, a teacher may assign a lot of exercises to practice



If a student is smart and active, he can first analyze the exercises to figure out which are the key ones to understand the concept, then he can selectively do exercises rather than doing all of them

Active Learning

In mathematics

- Given: a pool of unlabeled data $\{\boldsymbol{x}_i\}$ for $i = 1, \ldots, N$, and a budget to query labels from an oracle
- Goal:
  - 1) to an selectively choose unlabeled data to query the oracle based on the budget to retrieve the corresponding labels : $\{\boldsymbol{x}_j, y_j\}$, $j = 1, \ldots, L$, and $L \ll N$
  - 2) learn a mapping $f: \boldsymbol{x} \rightarrow y$ with $\{\boldsymbol{x}_j, y_j\}$ using a supervised learning algorithm
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$

Gene Disease Example

- Suppose we have budget to conduct the gene test on 20 more patients to verify whether they have the gene disease or not
  - i.e., we are able to get 20 more labeled data instances
- Which 20 patients should we conduct the gene test on?
- Active learning is focused on selecting 20 more patients to test their "labels" such that with these 20 more labeled data instances, a <u>precise</u> classifier can be trained

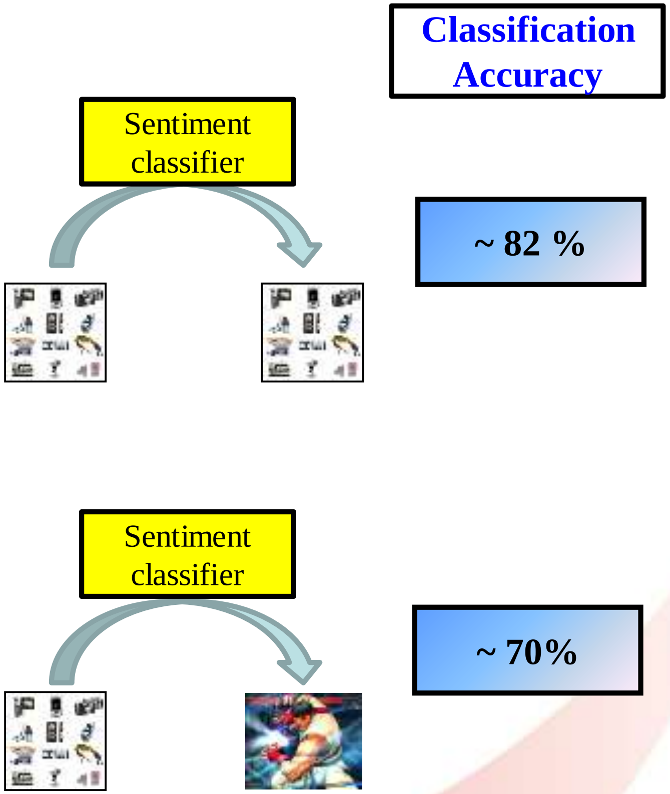
Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Motivating Example I

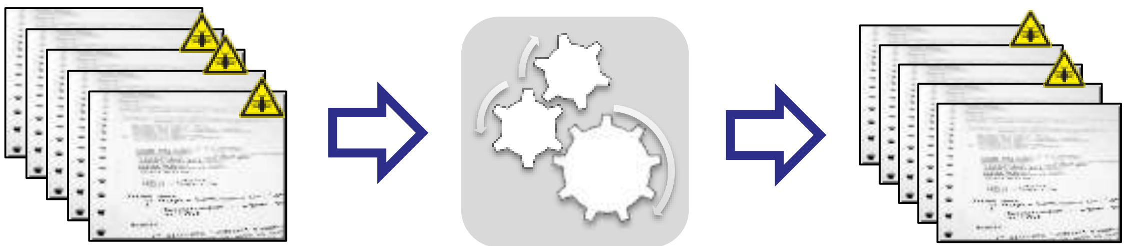
Electronics	Video Games
(1) **Compact**; easy to operate; very good picture quality; looks **sharp**!	(2) A very good game! It is action packed and full of excitement. I am very much **hooked** on this game.
(3) I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and **sharp**.	(4) Very **realistic** shooting action and good plots. We played this and were **hooked**.
(5) It is also quite **blurry** in very dark settings. I will never buy HP again.	(6) The game is so **boring**. I am extremely unhappy and will probably never buy UbiSoft again.

Product reviews on different domains



Motivating Example II

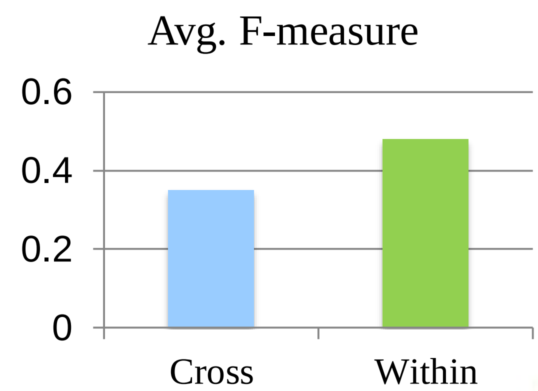
For a particular project:



Program with defect information

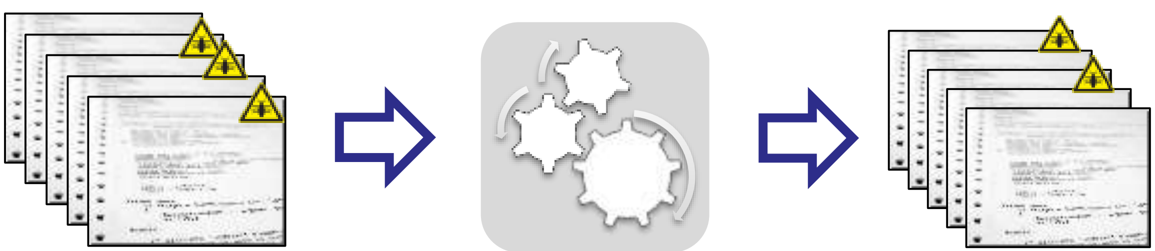
Predictive Model

Future defects



Cross-project:

OR



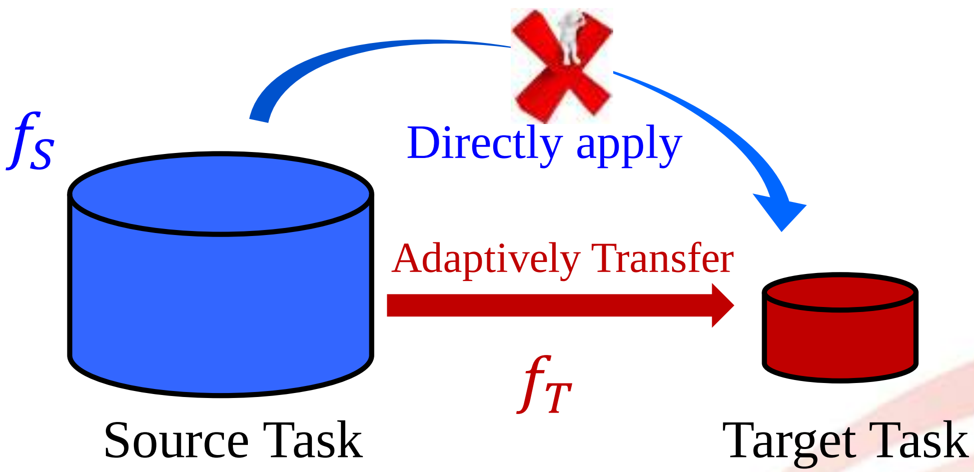
Program with defect information

Predictive Model

Program in another Project

Motivation

- **Assumption:** training and test data are assumed to be
  - Represented in the same feature space, AND
  - Follow the same data distribution
- **In practice:** training and test data come from different domains
  - Represented in different feature spaces, OR
  - Follow different data distributions



What if machines have transfer learning ability?

Transfer of Learning

- The study of dependency of human conduct, learning or performance on prior experience
  - [Thorndike and Woodworth, 1901] explored how individuals would transfer in one context to another context that share similar characteristics.

Transfer Learning

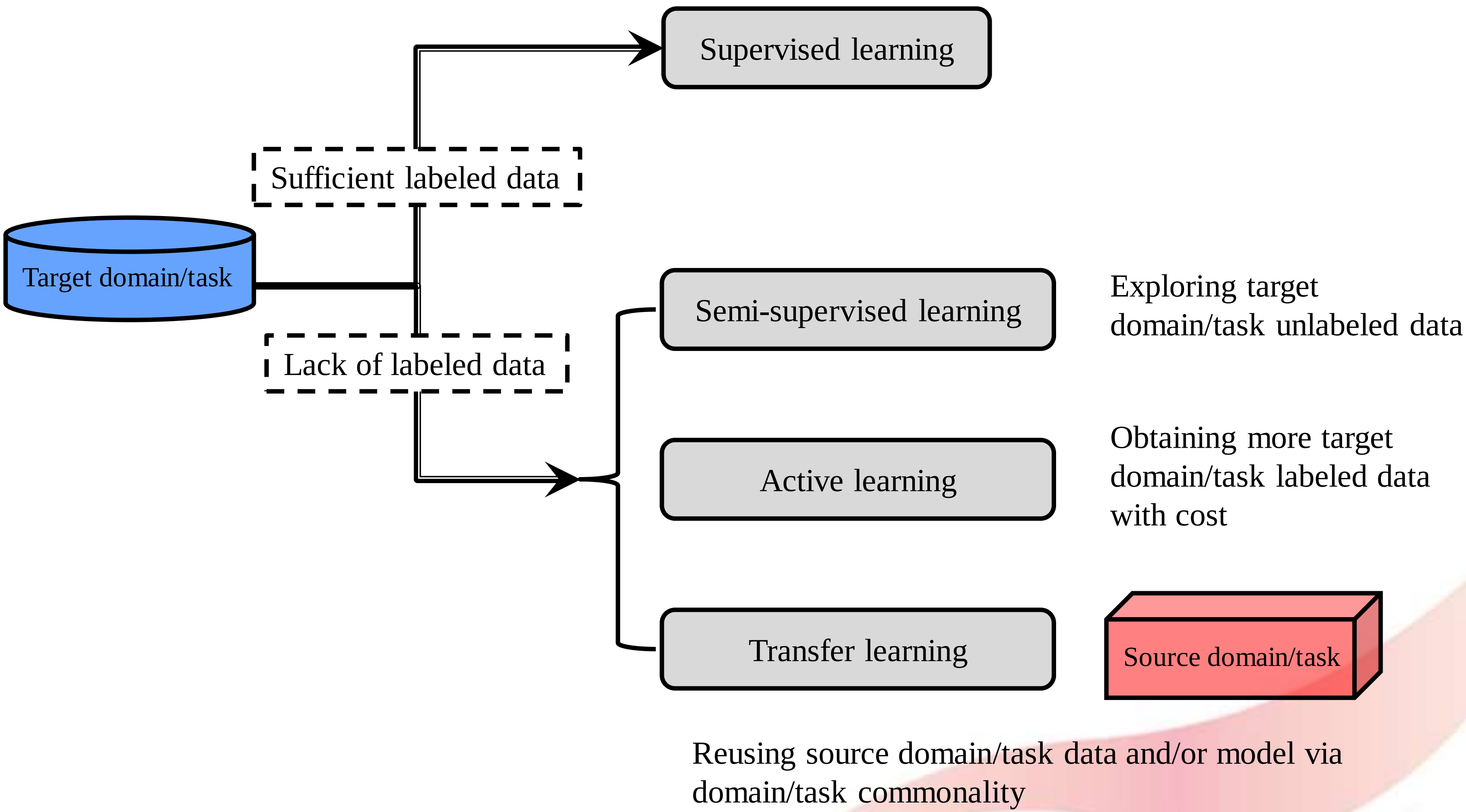
In mathematics

- Given: a small set of labeled data $T = \{\boldsymbol{x}_i, y_i\}$ for $i = 1, \ldots, L$, for a target domain/task, and a set of plenty labeled data $S = \{\boldsymbol{x}_j, y_j\}$ for $j = 1, \ldots, N$, from a source domain/task. In general, $L \ll N$
- Goal: to identify the commonality between the source data S and target data T, and utilize S to help learning a precise model $f: \boldsymbol{x} \rightarrow y$ from T
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$ for the target domain/task

Gene Disease Example

- We may have labeled data of other gene diseases
- Different gene diseases may have some correlations or share some common characteristics
- If the correlations/common characteristics can be automatically extracted and utilized, then labeled data from other gene disease may be used to help learning a classifier for the target gene disease
- Transfer learning can help！

Supervised Learning v.s. Advanced Paradigms



Different Learning Paradigms

- Supervised Learning ✓
- Unsupervised Learning ✓
- Reinforcement Learning ✗

AI6101 Introduction to AI and AI Ethics OR AI6125 Multi-agent System

- Advanced paradigms:

Semi-supervised learning literature survey

  - Semi-supervised learning ✗
  - Active learning ✗

Active learning literature survey

  - Transfer learning ✗

Transfer learning literature survey

Course Schedule (Tentative)

Date		Topics	Note
Week 1	13 Aug.	Introduction	
Week 2	20 Aug.	Data and Operations	
Week 3	27 Aug.	Linear models: regression	Team project released (<=3 members/team)
Week 4	3 Sep.	Linear models: classification	
Week 5	10 Sep.	Kernel methods	Assignment released (2 weeks to finish)
Week 6	17 Sep.	Tree-based methods	
Week 7	24 Sep.	Bayesian classifiers	
Recess Week (Assignment Deadline) 28 Sep.			
Week 8	9 Oct.	Dimensionality reduction	
Week 9	16 Oct.	Clustering	
Week 10	23 Oct.	Density estimation	
Week 11	30 Oct.	Flow-based Generative AI	Partially examined
Week 12	6 Nov.	Flow-based Generative AI	Partially examined
Week 13	13 Nov.	Quiz	In-person quiz! Scope: Lectures 2 – 12, ~1 hour open book
21 Nov			Project submission deadline

Reference

- Reference:
  - **Introduction to Machine Learning (2nd Ed.)**, by Ethem Alpaydin, The MIT Press, 2010.
  - **Introduction to Data Mining**, by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison Wesley, 2005.
  - **Pattern Recognition and Machine Learning**, by Christopher M. Bishop, Springer, 2006.
  - **Transfer Learning**, by Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan, Cambridge University Press, 2020.
- Regarding Mathematics:
  - Part I of the MIT Press book "*Deep Learning*" http://www.deeplearningbook.org/

Machine Learning Practice

- Important note: different from many other modules in the Master of Science in Artificial Intelligence (MSAI) programme, this machine learning module can be considered as an "applied mathematics" module, focusing on introducing the principles of different machine learning methods
- For practice:
  - Kaggle (highly recommended): http://www.kaggle.com/
  - UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html

Libraries and Platforms

- scikit-learn (Python):
  - http://scikit-learn.org/stable/
- Weka (Java)
  - http://www.cs.waikato.ac.nz/ml/weka/
- MALLET (Java)
  - http://mallet.cs.umass.edu/
- Tensorflow:
  - https://www.tensorflow.org/
- Pytorch:
  - https://pytorch.org/
- Many other libraries on deep learning
  - http://deeplearning.net/software_links/

Deep learning related, not necessary in this module

Top-tier Academic Conferences

- International Conference on Machine Learning (ICML)
- Neural Information Processing Systems (NIPS)
- Conference on Learning Theory (COLT)

General machine learning

- Uncertainty in Artificial Intelligence (UAI)
- International Conference on AI & Statistics (AISTATS)

Statistical AI methods

- International Joint Conference on Artificial Intelligence (IJCAI)
- AAAI Conference on Artificial Intelligence (AAAI)

General AI

- International Conference on Learning Representations (ICLR)

Deep learning

- International Conference on Knowledge Discovery and Data Mining (KDD)

Data mining

Top-tier Academic Journals

- Journal of Machine Learning Research (JMLR)
- Machine Learning (MLJ)
- IEEE Transactions on Neural Networks and Learning Systems (TNNLS)

General machine learning

- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)

Machine learning application in CV

- Artificial Intelligence (AIJ)
- Journal of Artificial Intelligence Research (JAIR)

General AI

Thank you!