Special Topics/Directions on Machine Leaning (SDML)

Prof. SD Lin 林守德 (SDLIN@csie.ntu.edu.tw)

Mission

- This course discusses a variety of goals and sub-tasks for machine learning.
- This course aims at developing students the capability of utilizing machine learning techniques to solve real-world tasks.
- This course aims at training students how to deal with practical challenges while utilizing machine learning models.

Course Information

- Meeting time: Thu 14:20-17:20
- Office hours: After class, or email me to make an appointment
- Reading materials: There will be many papers
- Pre-requisite
  - Machine Learning (or similar courses)
    - Basic Programming Skill
    - Probability and Linear Algebra
- Course FB link?
- TA: Skyly Yang d05922017@ntu.edu.tw, Tue 13:00~14:00 (R302)
  Nancy Cheng nancy.cheng.tl@gmail.com, Fri 14:00~15:00
  Liang-Shin Shen r06922011@ntu.edu.tw, Mon 11:00~12:00
  Yen-Ting Lee r06922008@ntu.edu.tw, Mon 11:00~12:00

FAQ for the SML Course

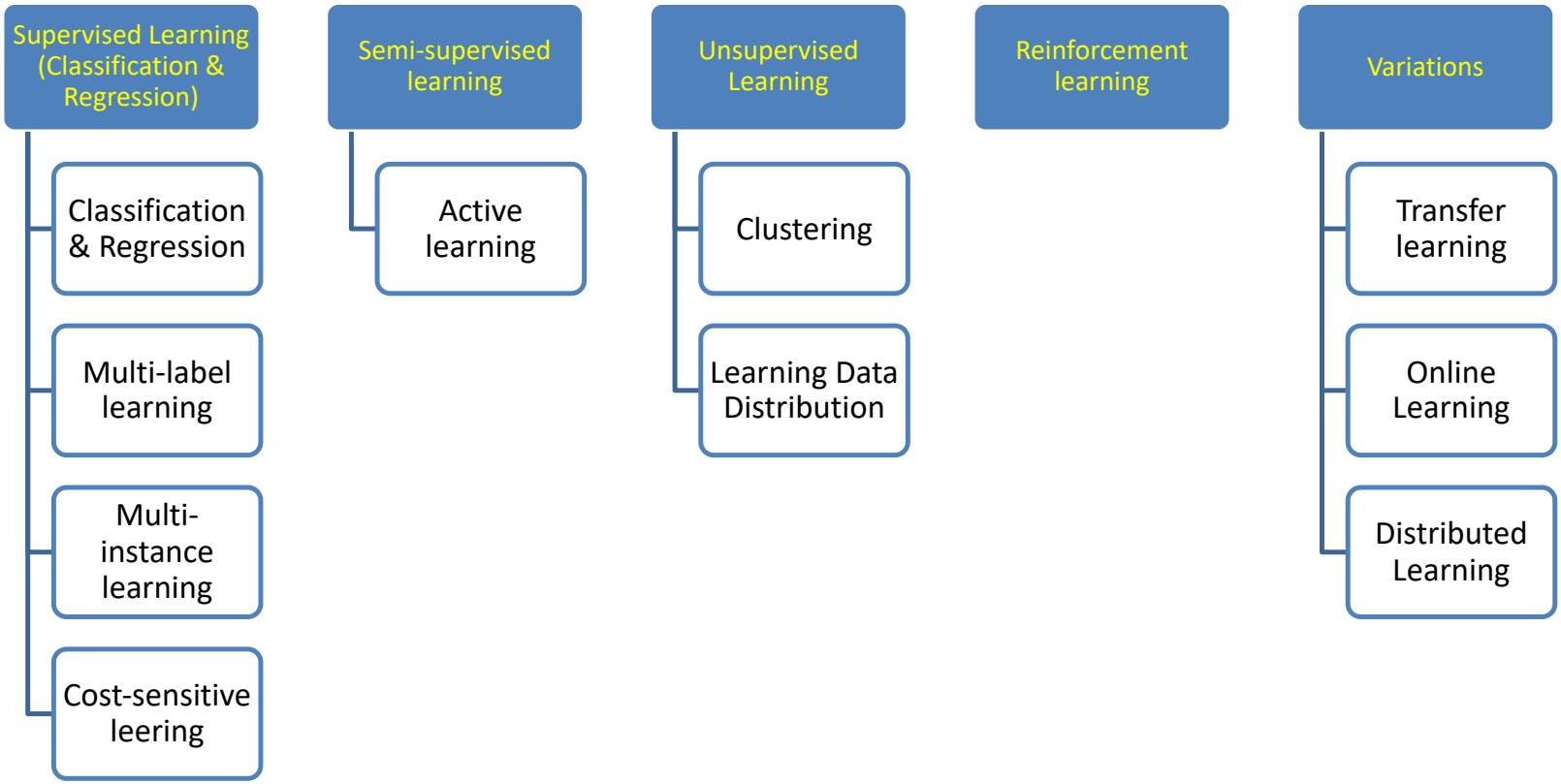
- Why should I take this course
  - I have interests in this area/topic ☺
  - It's cool, useful, and unique ☺
  - It's relevant to my research ☺
  - I can get high grade without working hard ☹
  - I have no other choices ☹
  - The instructor and TA is charming …..THANK YOU
- How to do well in this class
  - Come to the class and spend time thinking/working on the assignments/projects
  - Base on past experience, about 10% of the students will drop or fail

Basics about Machine Learning

What is Machine Learning

- ML tries to optimize a performance criterion using example data or past experience.
- Mathematically speaking: given some data X, we want to learn a function mapping f(X) for certain purpose
  - f(x)= a label y → classification
  - f(x)= a set Y in X → clustering
  - f(x)=p(x) → probabilistic graphical model
  - f(x)= a set of y → multi-label classification
  - f(x)= an action → reinforcement learning
  - …
- ML techniques tell us how to produce high quality f(x), given certain objective and evaluation metrics

A variety of ML Scenarios



Supervised Learning

- Given: a set of <input, output> pairs
- Goal: given an unseen input, predict the corresponding output
- For example:
  1. Input: X-ray photo of chests, output: whether it is cancerous
  2. Input: a sentence, output: whether a sentence is grammatical
  3. Input: some indicators of a company, output: whether it will make profit next year
- There are two kinds of outputs an ML system generates
  - Categorical***: classification problem (E1 and E2)***
    - *Ordinal outputs: small, medium, large*
    - *Non-ordinal outputs: blue, green, orange*
  - Real values***: regression problem (E3)***

Classification (1/2)

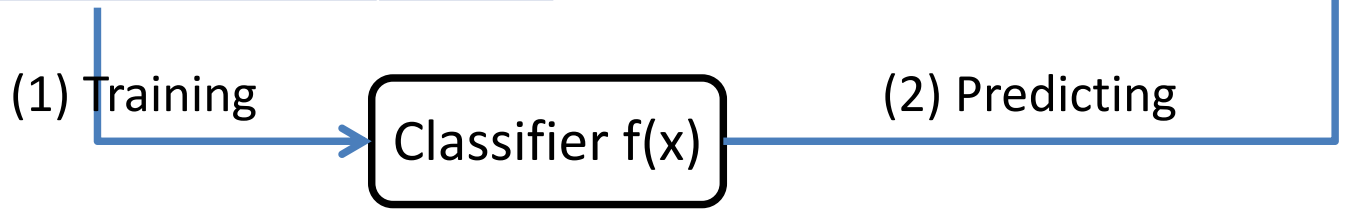
- It's a supervised learning task that, given a real-valued feature vector x, predicts which class in C may be associated with x.
- $|C|=2$ ➔ Binary Classification
  $|C|>2$ ➔ Multi-class Classification
- Training and predicting of a binary classification problem:

Training set (Binary Classification)

Feature Vector ($x_i \in R^d$)	Class
x_1	+1
x_2	-1
…	…
x_{n-1}	-1
x_n	+1

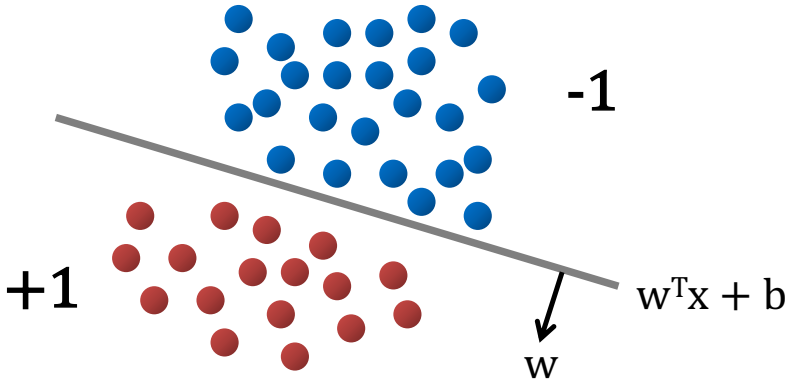
A new instance

Feature Vector ($x_{new} \in R^d$)	Class
x_{new}	?



Classification (2/2)

- A classifier can be either linear or non-linear
- The geometric view of a linear classifier

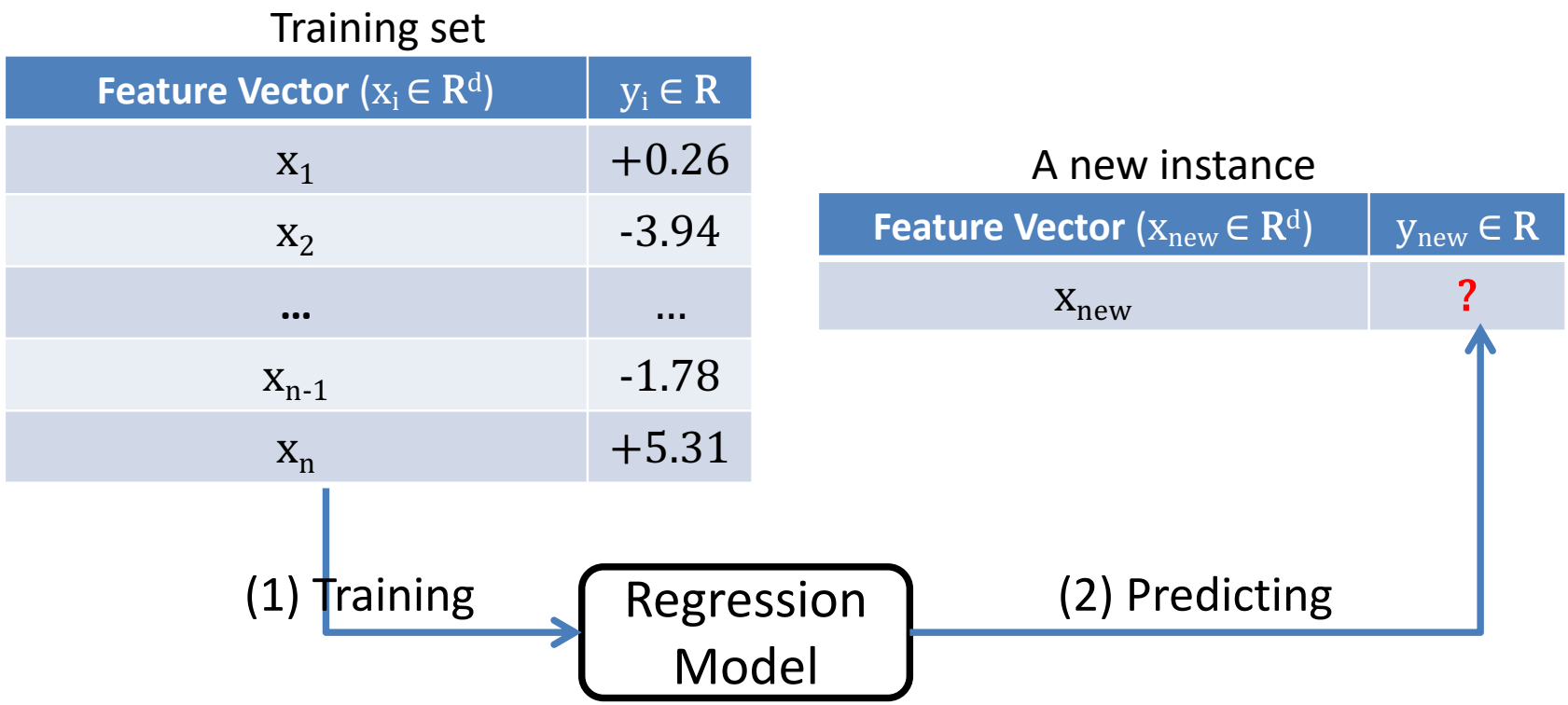


We will talk more about classification later !!

- Famous classification models:
  - k-nearest neighbor (kNN)
  - Decision Tree (DT)
  - Support Vector Machine (SVM)
  - Neural Network (NN)
  - …

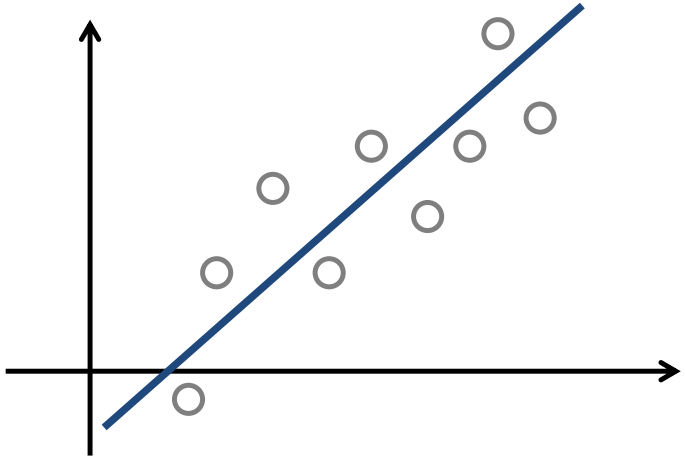
Regression (1/2)

- A supervised learning task that, given a real-valued feature vector x, predicts the target value $y \in R$.
- Training and predicting of a regression problem:



Regression (2/2)

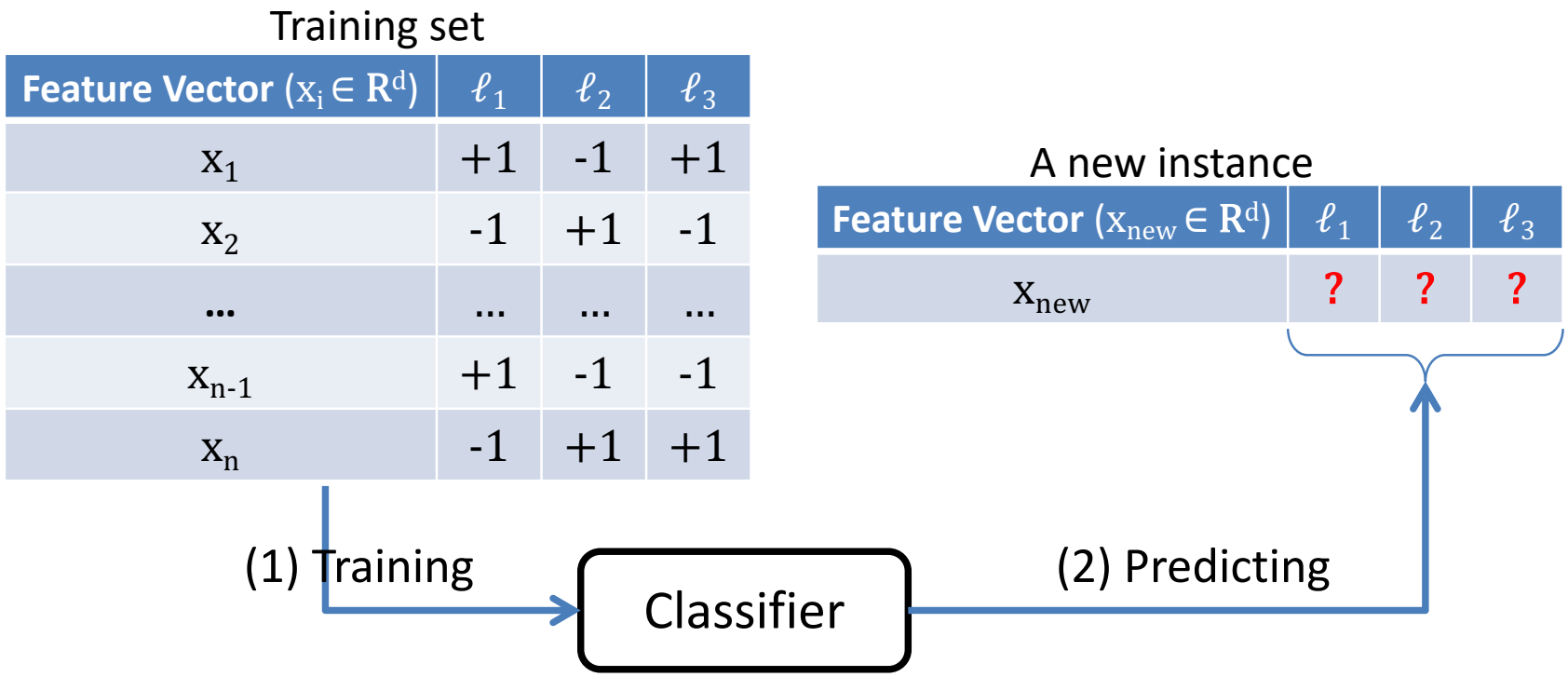
- The geometric view of a linear regression function



- Some types of regression: linear regression, support vector regression, …

Multi-label Learning

- A classification task in that an instance is associated with a set of labels, instead of a single label.



- Existing models: Binary Relevance, Label Powerset, ML-KNN, IBLR, …

Multimedia tagging

- Many websites allow the user community to add tags, thus enabling easier retrieval.

Example of a tag cloud: the beach boys, from Last.FM (Ma et al., 2010)

Cost-sensitive Learning

- A classification task with non-uniform cost for different types of classification error.
- Goal: To predict the class C* that minimizes the expected cost rather than the misclassification rate

$$C^* = \arg\min_j \sum_k P(Y = C_k \mid x) L_{jk}$$

- An example cost matrix L : medical diagnosis

L_{jk}	Actual Cancer	Actual Normal
Predict Cancer	0	1
Predict Normal	10000	0

- Methods: cost-sensitive SVM, cost-sensitive sampling

Examples for Cost-sensitive Learning

- Highly non-uniform misclassification costs are very common in a variety of challenging real-world machine learning problems
  - Fraud detection
  - Medical diagnosis
  - Various problems in business decision-making.

Credit cards are one of the most famous targets of fraud. The cost of missing a target (fraud) is much higher than that of a false-positive.

Hung-Yi Lo, Shou-De Lin, and Hsin-Min Wang, "Generalized k-Labelsets Ensemble for Multi-Label and Cost-Sensitive Classification," IEEE Transactions on Knowledge and Data Engineering.

Multi-instance Learning (1/2)

- A supervised learning task in that the training set consists of *bags of instances*, and instead of associating labels on instances, labels are *only* assigned to bags.
- In the binary case,

  - Positive bag → at least one instance in the bag is positive
  - Negative bag → all instances in the bag are negative

- The goal is to learn a model and predict the label of a new instance or a new bag of instances.

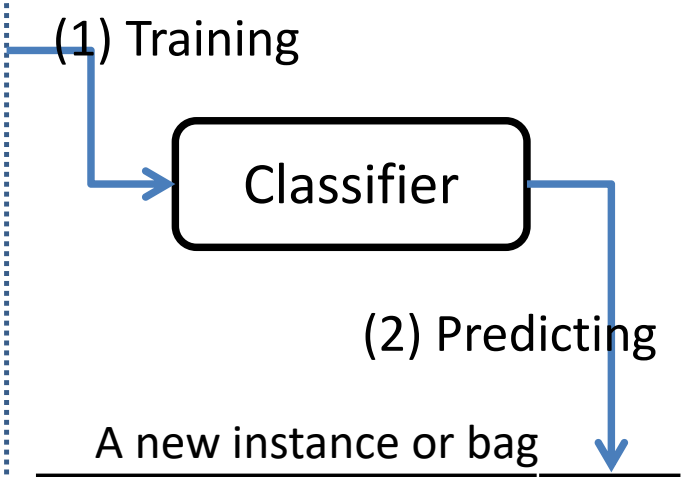
Multi-instance Learning (2/2)

- Training and Prediction

Training Set

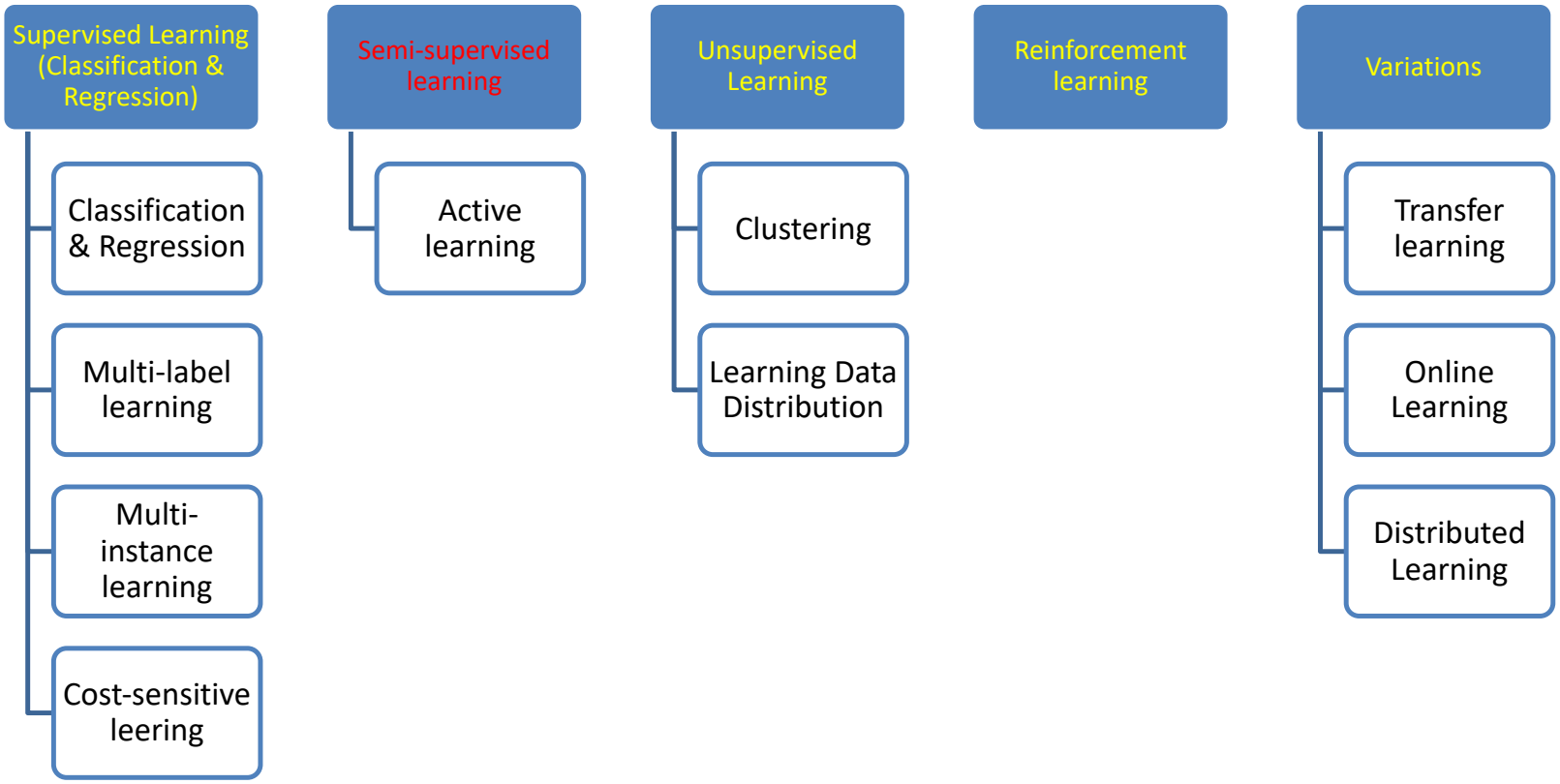
Feature Vector ($x_i \in R^d$)	Bag
x_1	1
x_2	1
x_3	2
x_4	2
x_5	2
…	…
x_{n-1}	m
x_n	m

Bag	Class
1	+1
2	-1
…	…
m	-1



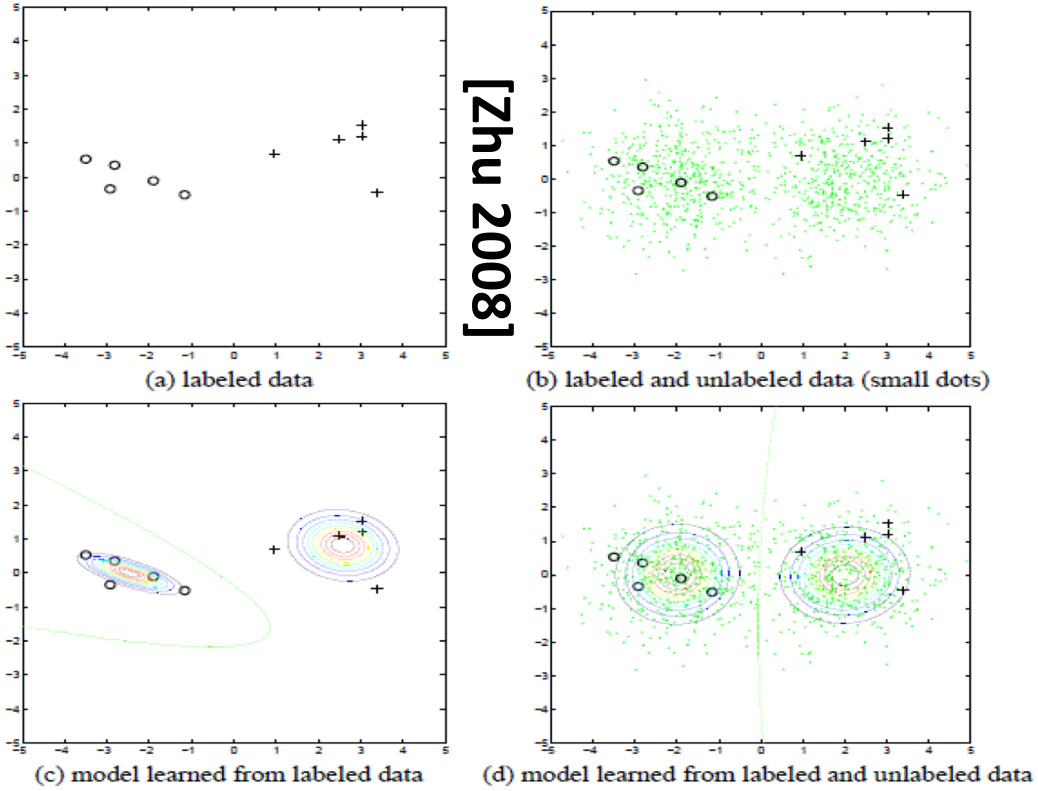
- Some methods: Learning Axis-Parallel Concepts, Citation kNN, mi-SVM, MI-SVM, Multiple-decision tree, …

A variety of ML Scenarios



Semi-supervised Learning

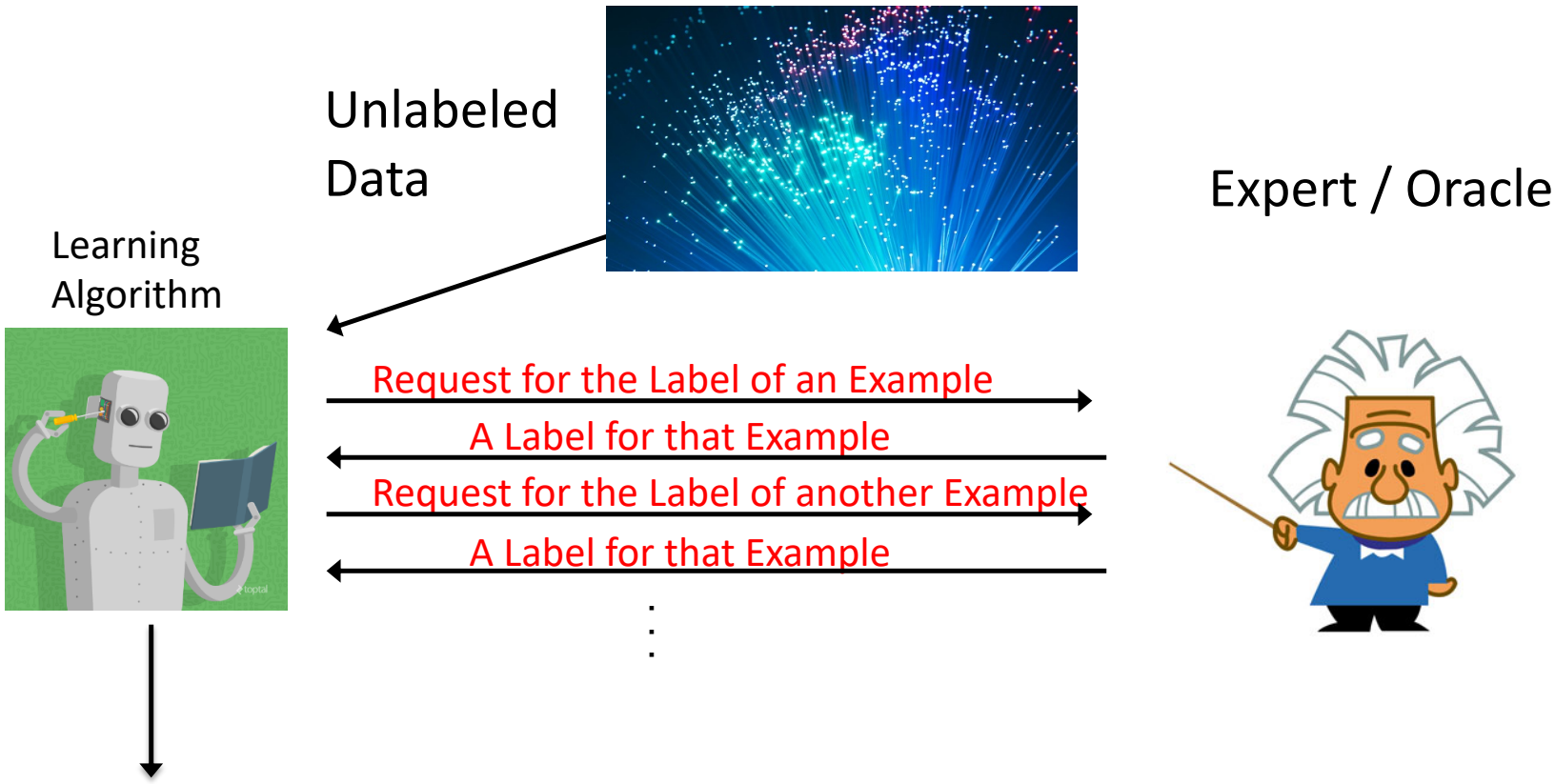
- Only a small portion of data are annotated (usually due to high annotation cost)
- Leverage the unlabeled data for better performance



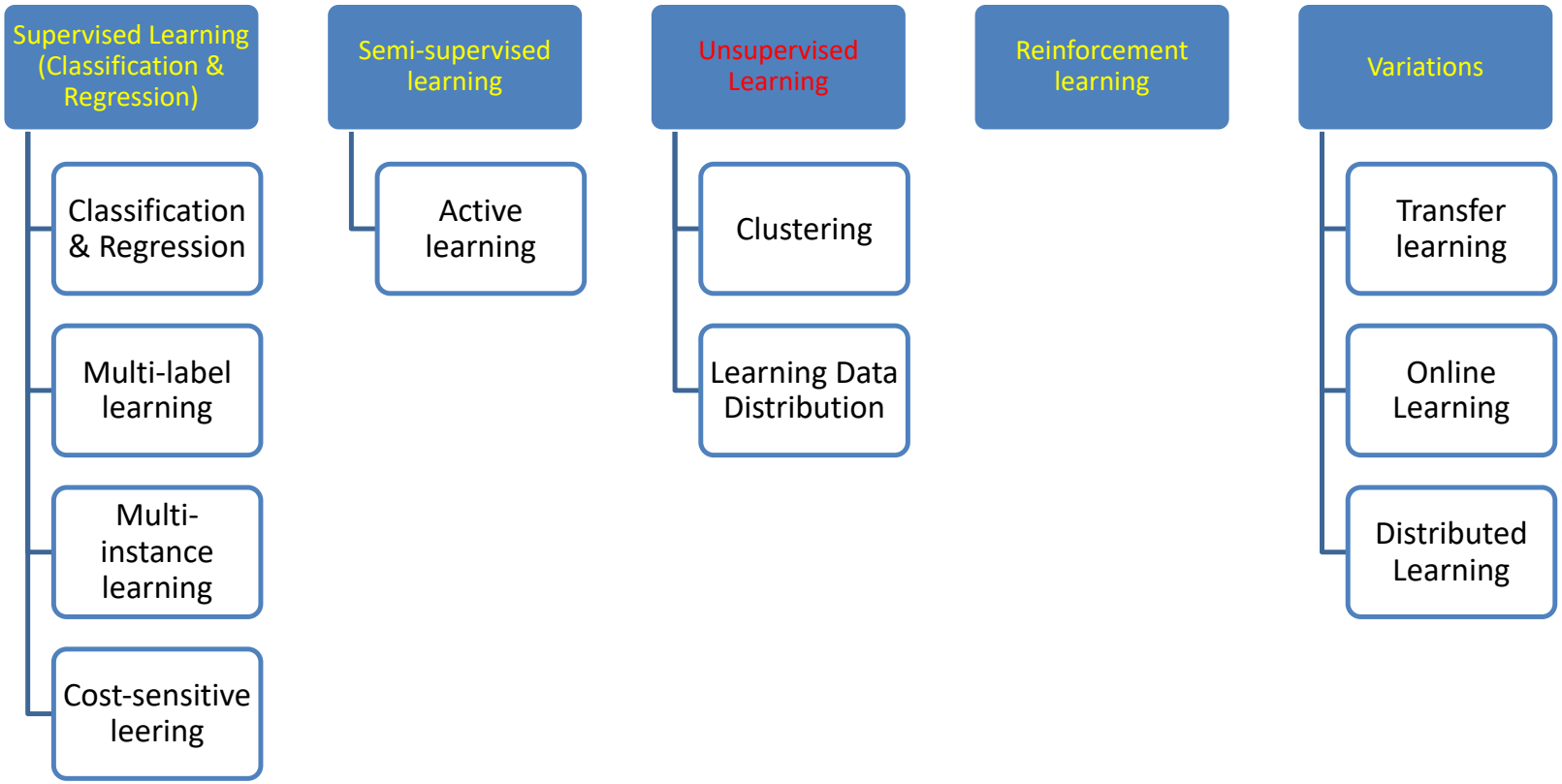
[Zhu 2008]

Active Learning

- Achieves better learning with fewer labeled training data via actively selecting a subset of unlabeled data to be annotated



A variety of ML Scenarios



Unsupervised Learning

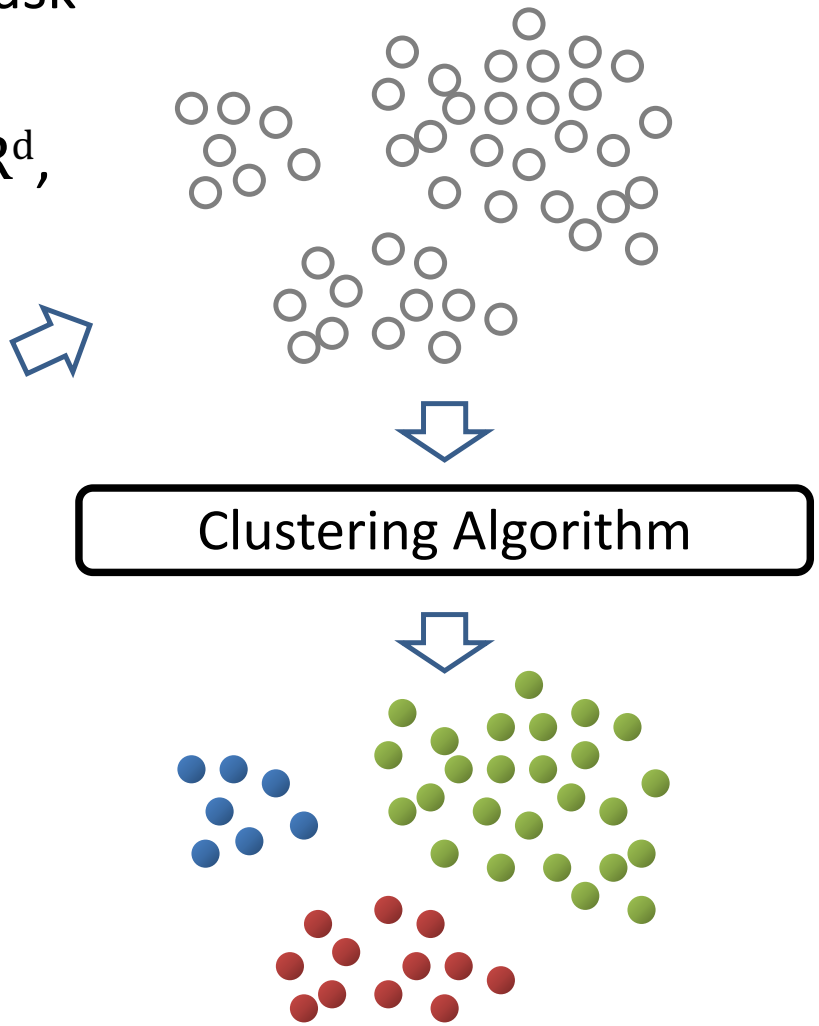
- Learning without teachers (presumably harder than supervised learning)
  - Learning “what normally happens”
  - Think of how babies learn their first language (unsupervised) comparing with how people learn their 2nd language (supervised).
- Given: a bunch of input X (there is no output Y)
- Goal: depending on the tasks, for example
  - Estimate P(X) ➔ then we can find augmax P(X) ➔PGM
  - Finding $P(X_2|X_1)$➔ we can know the dependency between inputs ➔ PGM
  - Finding $Sim(X_1,X_2)$ ➔ then we can group similar X’s ➔ clustering

Clustering

- An unsupervised learning task
- Given a finite set of real-valued feature vector $S \subset R^d$, discover clusters in S

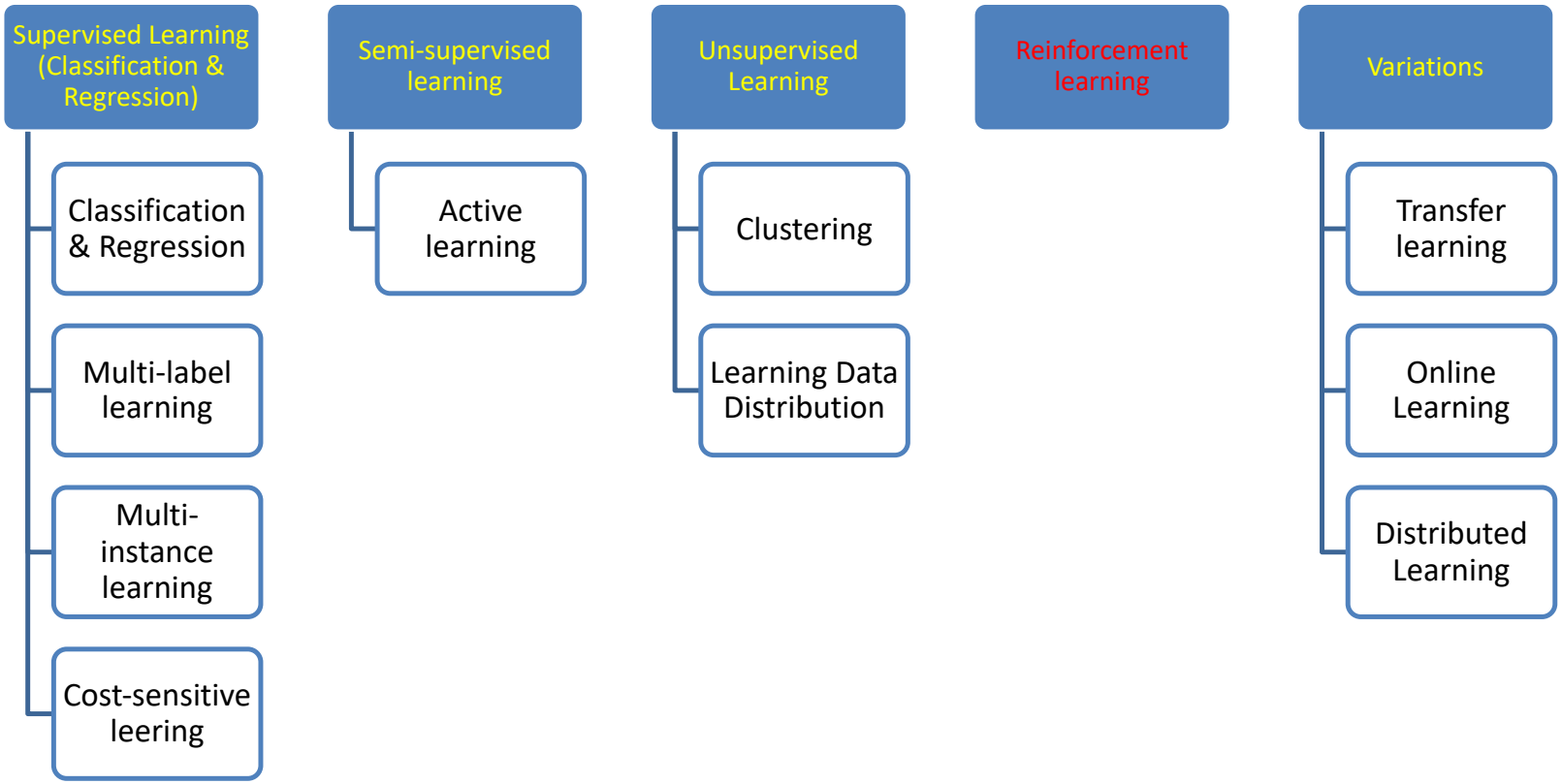
S

Feature Vector ($x_i \in R^d$)
x_1
x_2
…
x_{n-1}
x_n



- K-Means, EM, Hierarchical classification, etc

A variety of ML Scenarios



Reinforcement Learning (RL)

- RL is a “decision making” process.
  - How an agent should make decision to maximize the long-term rewards
- RL is associated with a sequence of states X and actions Y (i.e. think about Markov Decision Process) with certain “rewards”.
- It’s goal is to find an optimal policy to guide the decision.

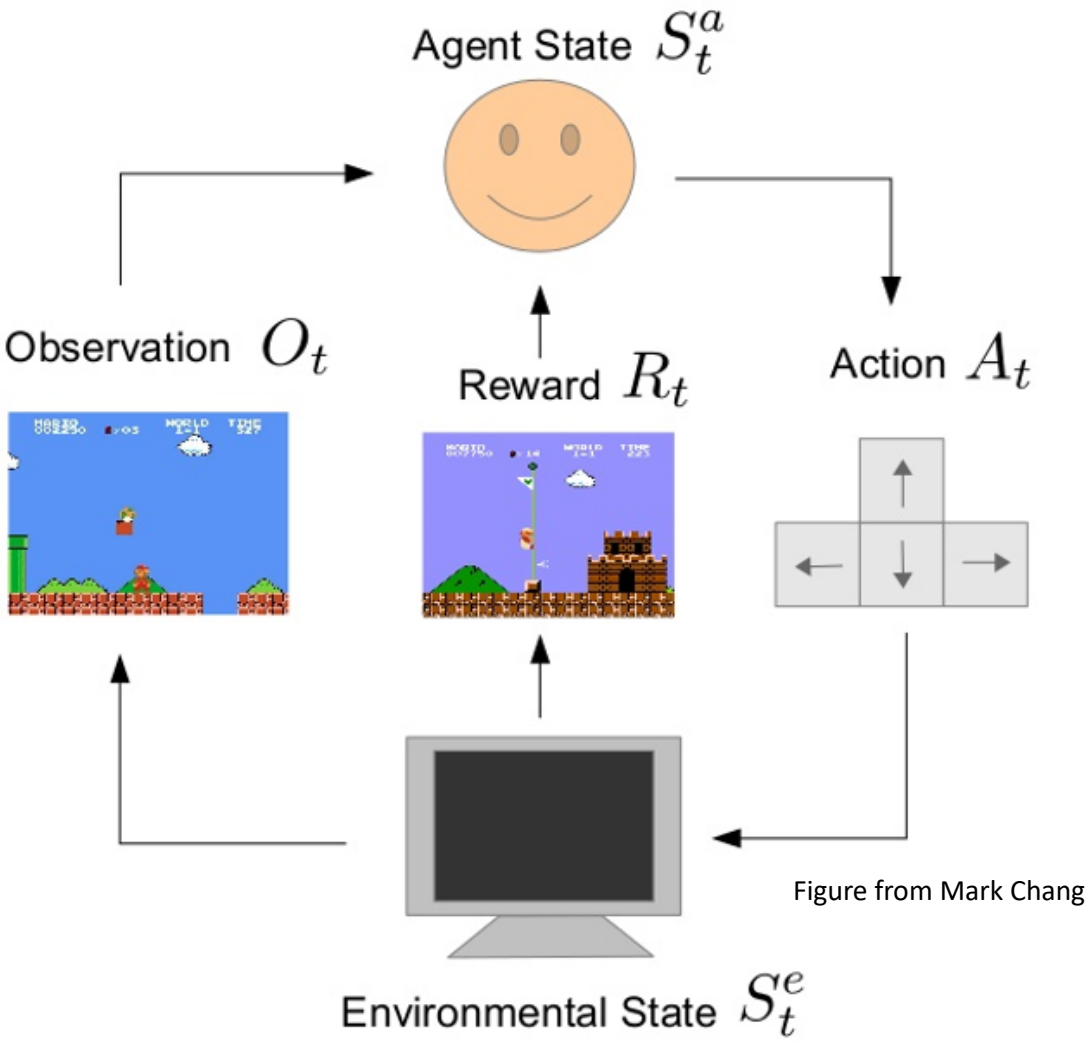
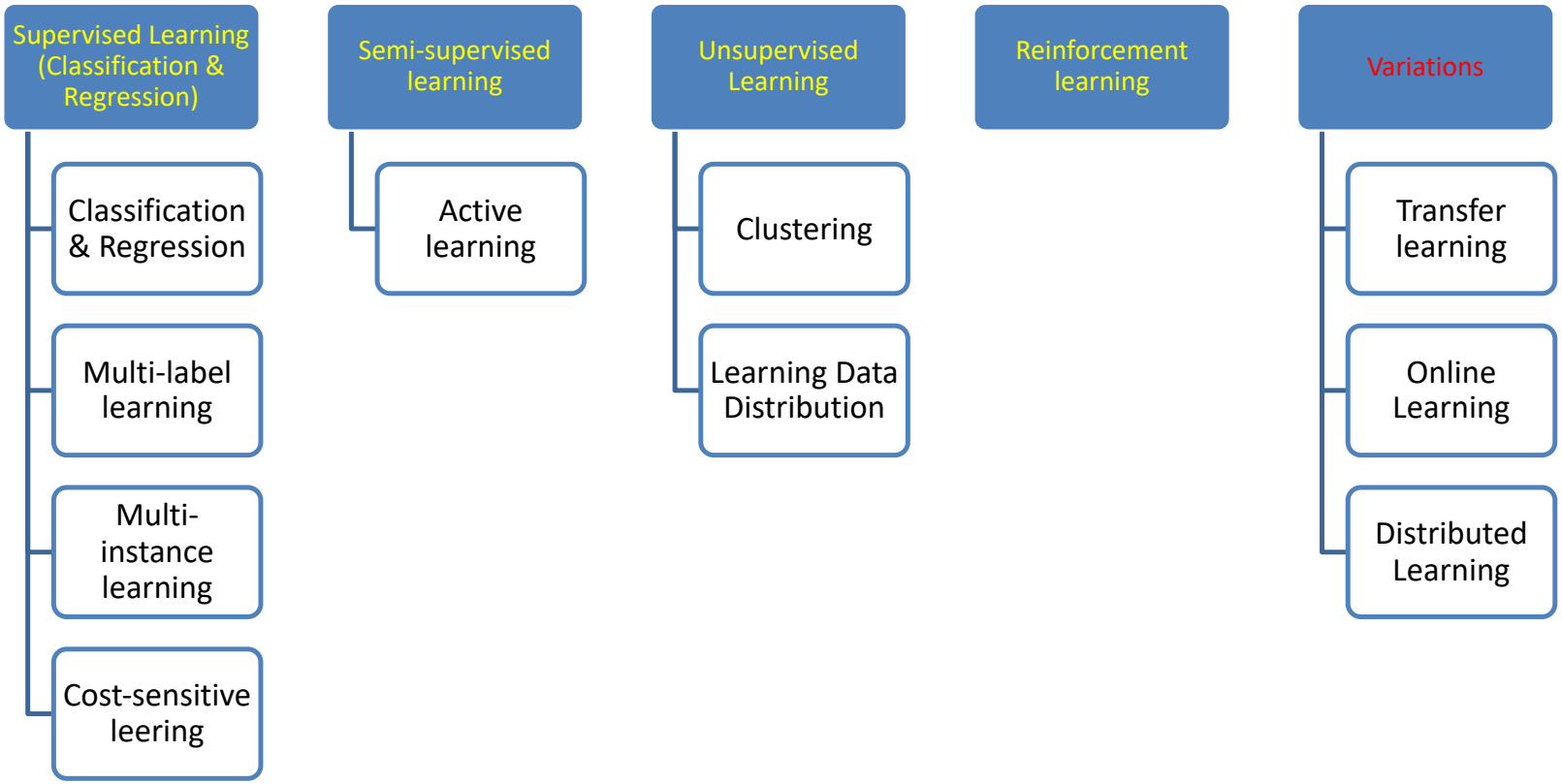


Figure from Mark Chang

AlphaGo: SL+RL

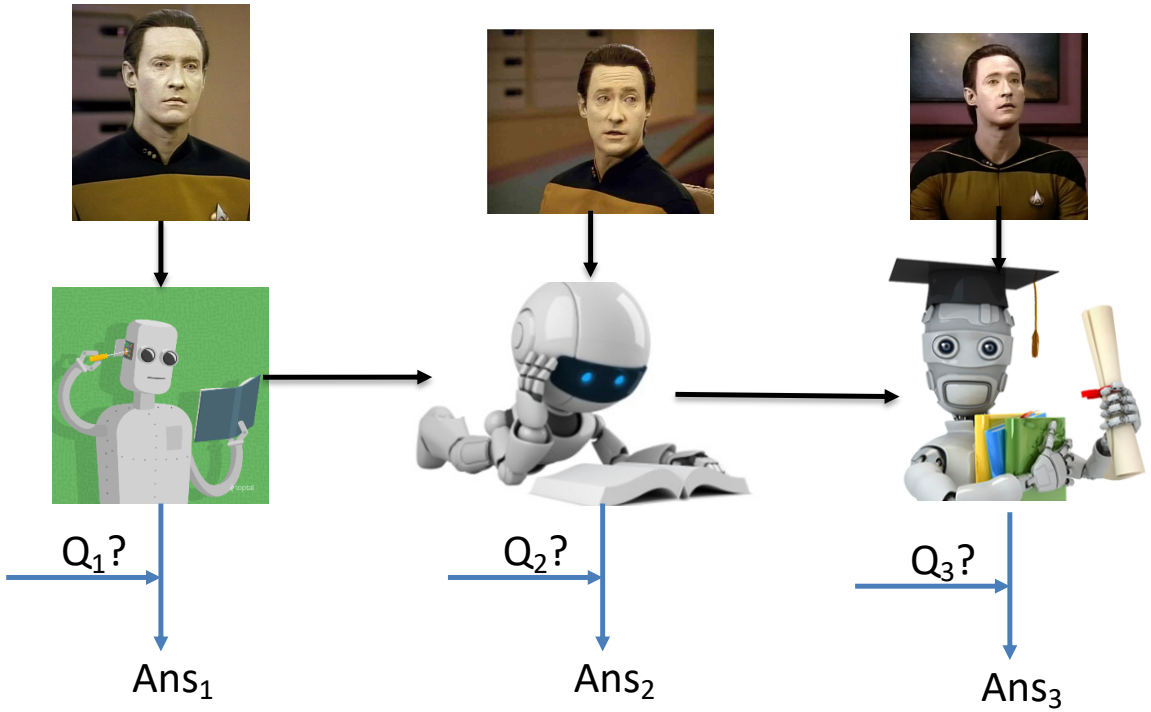
- 1st Stage:天下棋手為我師 (Supervised Learning)
  - Data: 過去棋譜
  - Learning: f(X)=Y, X: 盤面, Y:next move
  - Results: AI can play Go now, but not an expert
- 2nd Stage:超越過去的自己(Reinforcement Learning)
  - Data: generating from playing with 1st Stage AI
  - Learning: Observation→盤面, reward→if win, action→ next move

A variety of ML Scenarios

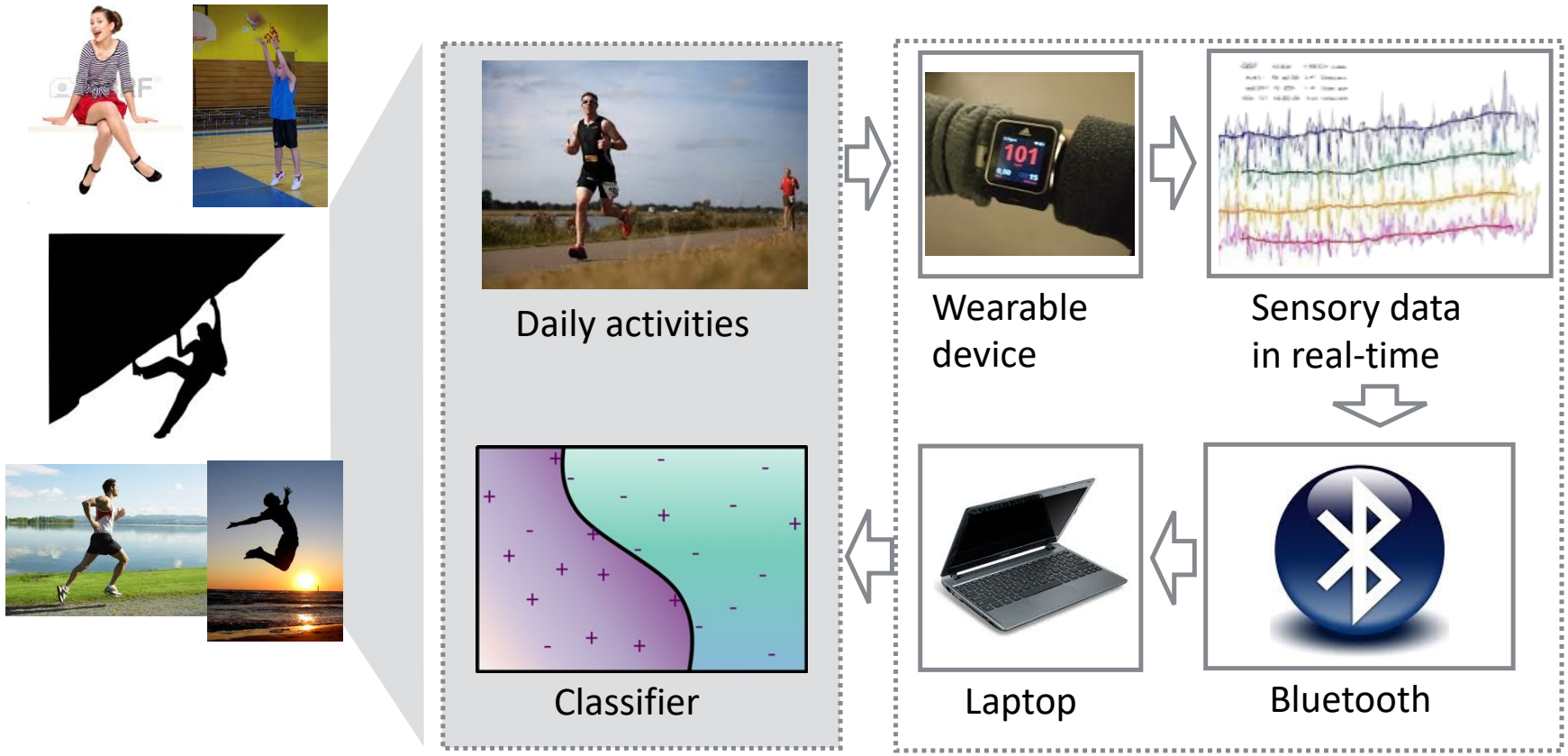


Online Learning

- Data arrives incrementally (one-at-a-time)
  - Once a data point has been observed, it might never be seen again.
  - Learner makes a prediction on each observation.
- Time and memory usage cannot scale with data.
  - Algorithms may not store previously seen data and perform batch learning.
  - Models resource-constrained learning, e.g. on small devices.



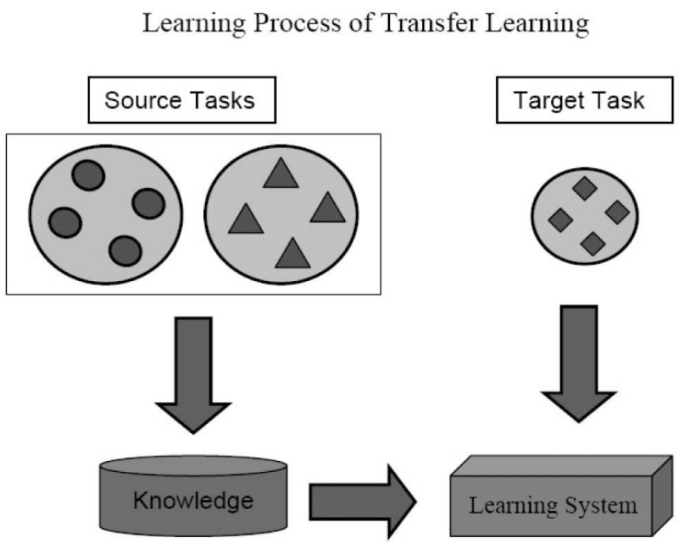
Scenario: Human activity recognition using data from wearable devices (one of the dataset we have experimented)



- There are a variety of models to be learned (individual * activity)
- Data are coming incrementally while we don't want to transmit everything to server
- The incoming data are unlabeled

Transfer Learning

- Improving a learning task via incorporating knowledge from learning tasks in other domains with different feature space and data distribution.
- Reduces expensive data-labeling efforts
- Example: the knowledge for recognizing an airplane may be helpful for recognizing a bird.
- Approach categories:
  (1) Instance-transfer
  (2) Feature-representation-transfer
  (3) Parameter-transfer
  (4) Relational-knowledge-transfer

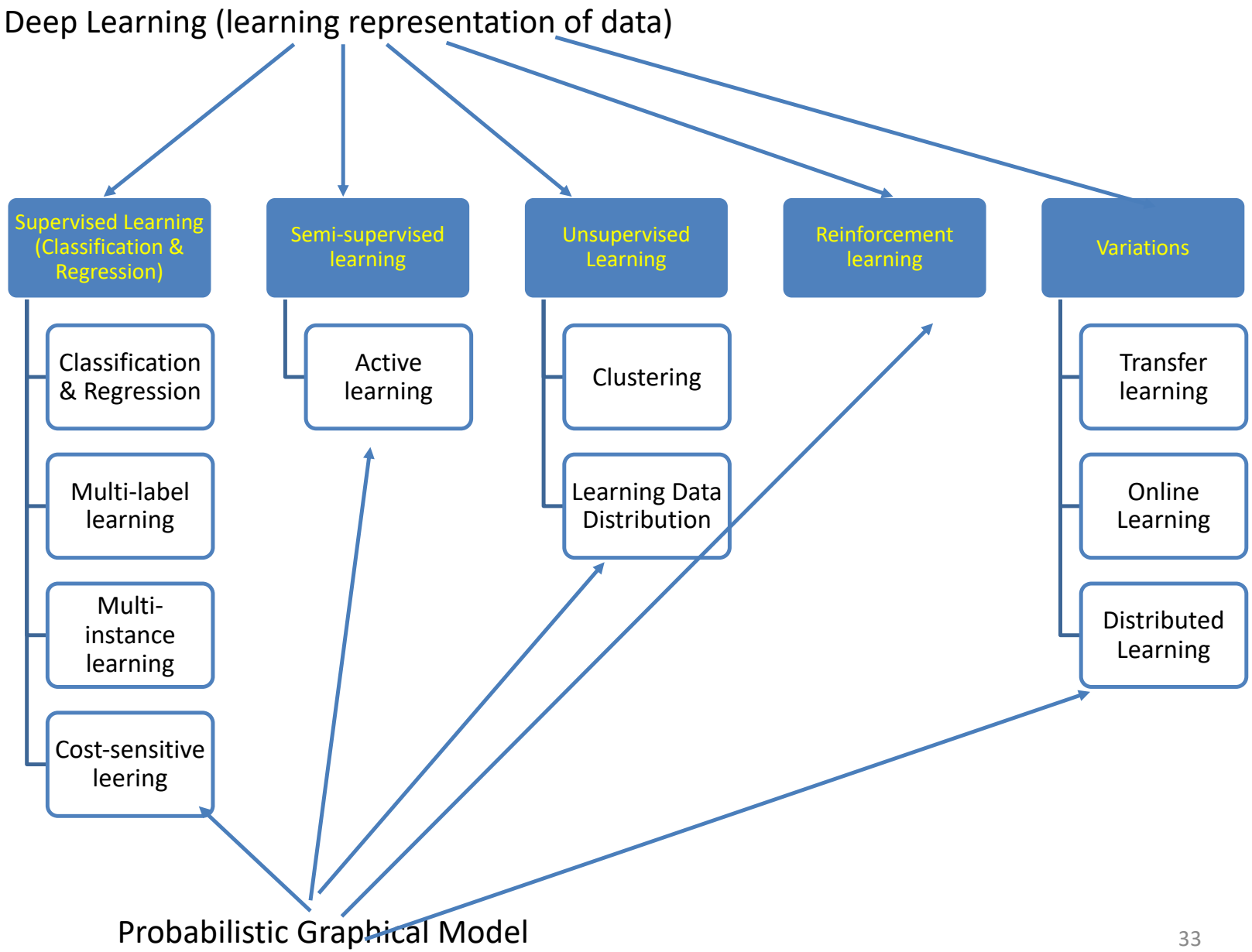


[Pan & Yang 2010]

Distributed Learning

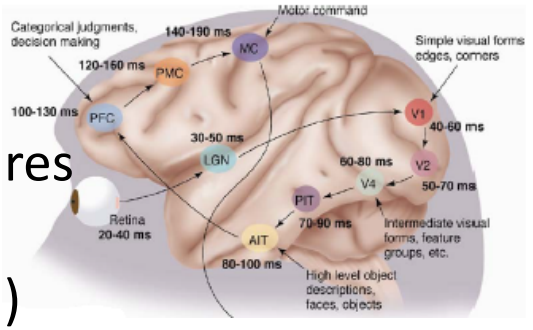
- Perform machine learning on multiple machines

Computation	Traditional Parallel Computing	Distributed Learning
# of machines	Few (10~100), communication cost can be ignored	Many (>1000), communication cost can be the bottleneck
Computational power	Powerful (cluster), dedicated	Ordinal (mobile), cannot be dedicated
Memory	Large	Small
Management	Strong control	Weak control


Deep Learning (learning representation of data)
Supervised Learning (Classification & Regression)
Semi-supervised learning
Unsupervised Learning
Reinforcement learning
Variations
Classification & Regression
Multi-label learning
Multi-instance learning
Cost-sensitive leering
Active learning
Clustering
Learning Data Distribution
Transfer learning
Online Learning
Distributed Learning
Probabilistic Graphical Model

Deep Learning

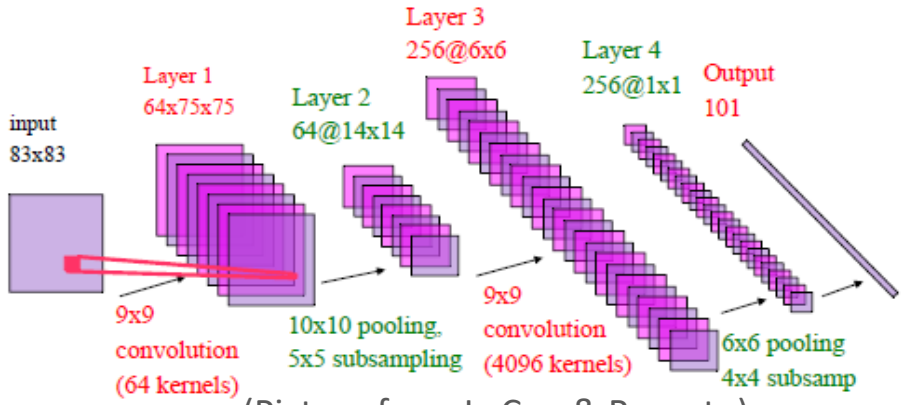
- Human brains perform much better than computers at recognizing natural patterns
- The brains learn to extract many layers of features
- Features in one layer are combined to form higher-level features in the layer above (Hinton)



- Multiple processing stages in the visual cortex (LeCun & Ranzato):

- Inspired by human brains, *deep learning* aims to learn multiple layers of representation from data, with increasing level of abstraction

Convolutional Network



(Picture from LeCun & Ranzato)

We will NOT cover most of the above topics

We will likely to cover the followings

- Learning embeddings
- Learning with partial inputs
- Learning time-series data
- Learning Language Models (sequence-to-sequence)
- Explaining ML models
- Other issues (selective)
  - Learning with small dataset
  - Unsupervised models
  - Reasoning Machines
  - Security & Privacy in learning
  - Scalability, efficiency, and other practical issues

Learning embeddings

- We would like to learn from data the vector representation of objects
- Why vector representations?
  - Easier to compute similarity
  - Embeddings can be pre-trained and stored
- What kind of embeddings can be trained
  - Word embeddings
  - Knowledge embeddings
  - Graph embeddings
  - …
- How to learn embeddings (embeddings can be learned supervisedly or unsupervisedly)?
  - Traditional solutions: Dimension-reduction techniques
  - DNN solutions: embedding layers

Learning with partial inputs

- Traditional Training in ML: X→Y, where X are fully observed
- What happens if there are a lot of (e.g. 99%) missing feature values in X?
- This is important for IoT data analysis and Recommender systems
- Factorization-based solution can be exploited here

Learning time-series data

- Forecasting time series has a wide range of application in
  - Financial (e.g. stock)
  - Environment (e.g. PM2.5)
  - Commercial products (e.g. TV ratings)
- It’s different from traditional ML tasks where the features are considered non-ordered.
  - How to exploit the order information becomes important

Learning Language Models (sequence-to-sequence)

- Language models usually capture the order information (in probability) of words ➔ P(Y|X)
- The main challenge lies in how to capture long-term and short-term dependency of data in a language model.
- Before RNN, people relies on n-gram language model for decades.
- RNN/LSTM/GLU provide a better solution to capture the longer-term dependency

Explaining ML models

- Machine learning models are generally complicated
- However, explaining why such results are produced is critical for human beings to accept them.
- There are several issues in this direction
  - Can we create a model that's easier to be explained?
  - Given certain model, can we explain why the results are produced?
  - What is a better way to explain a learning model?

Learning with small (labelled) datasets

- In real-world, large amount of labeled data are usually hard to obtain.
- How do improve the model given only small amount of labelled data?
  - Semi-supervised learning
  - Transfer learning
  - Active learning

Security & Privacy in learning

- ML models are proven fragile as they suffer the adversarial attack
- ML models can exploit user privacy, which becomes an ethical concern.

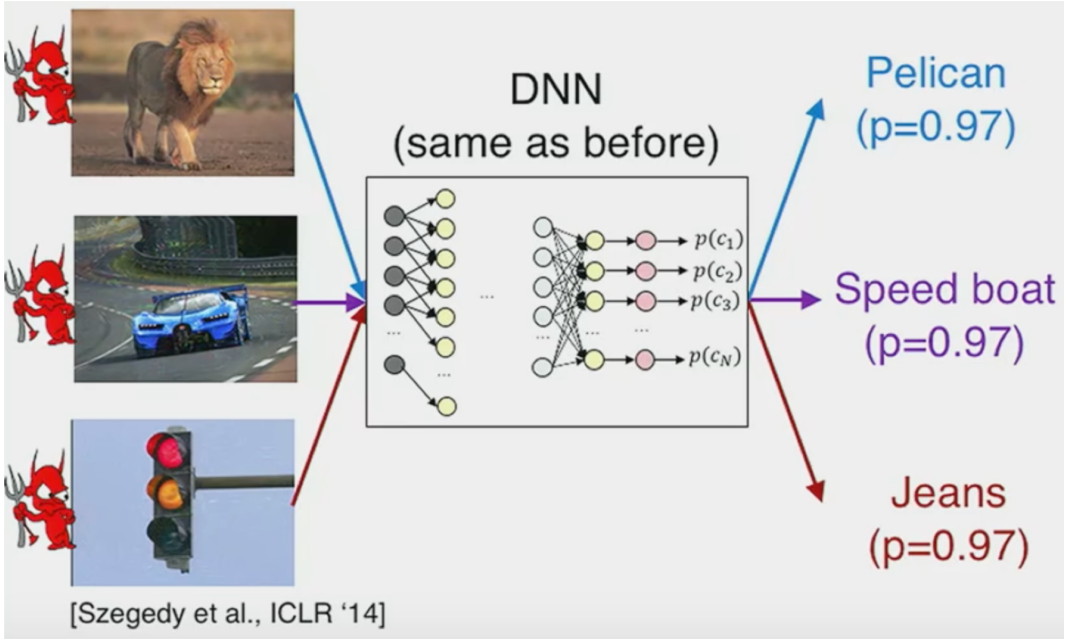


Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

Many other issues to be solved for ML

- Unsupervised Models
- Learning given incorrect labels
- Improving the efficiency of learning
  - Smaller model
  - Parallel learning
  - Better convergence
- Learning (data driven) with constraints (knowledge)
- Deep reasoning model
- …

Syllabus

9月13日	**Intro**	
9月20日	**Graph Embedding**	HW1 out
9月27日	**Document Embedding**	
10月4日	**Knowledge Embedding**	
10月11日	**HW1 presentation**	HW1 due
10月18日	**Recommendation & Factorization**	HW2 out
10月25日	**Advanced Recommendation**	
11月1日	**Time-series prediction**	
11月8日	**Sequence-to-sequence model**	
11月15日	**HW2 presentation**	HW2 due, HW3 out
11月22日	**break**	
11月29日	**Explanation of Deep Learning Model**	
12月6日	**Final project proposal presentation**	
12月13日	**HW3 presentation**	HW3 due
12月20日	**Selective topics (e.g. clustering)**	
12月27日	**Selective topics (e.g. deep reasoning model)**	
1月3日	**final presentation 1**	
1月10日	**final presentation 2**	Final Project Report Due

Grades

- 3 homework assignments (70%)
- 1 final project (30%)

About Homework Assignments (70%)

- All assignments require the hands-on coding work to create relevant models
- Each assignment contains two parts:
  - Basic task(s)
  - Advanced task(s)
- Some assignments will be in competition form
  - Please open a Kaggle account first
- The basic task(s) is an individual assignment, everybody needs to turn in their code and a report to describe the solution and results.
  - It will due sooner than the advanced tasks
  - Plagiarism will lead to the failure of this class (no exception)
  - Discussing is encouraged, but sharing code is forbidden
- For the advanced task, you can work as a team of 1~3 people.

Final Project (30%)

- Final Project needs to be done in a group of 3 persons
  - Find your teammates and discuss the potential topics ASAP
  - We will open a discussion board for team member matching
- You will decide your own topic for final projects
  - It has to be a problem related to ML
  - It has to be a problem originated from your team (or from our assignments)
  - It cannot be the work that you are doing in your lab (or for your thesis), nor a project for another class.
- Schedule
  - Proposal presentation on 12/6
  - Final presentation on 1/3 and 1/10
  - Final report (optional) due on 1/17

Addition to the Class

- Addition is possible, as long as you have already taken at least one
- ML or DL-related course
- Please email (subject: Addition to the SDML course) me and TA (r06922011@ntu.edu.tw) telling us which ML related course(s) you have taken, and we will send the registration code back to you
- Add at your own risk (this can be one of the toughest course you have ever taken)
  - Base on past experience, about 10% of the students will drop or fail