Foundation of Artificial Intelligence

人工智能基础

李翔宇

软件学院

Email: lixiangyu@bjtu.edu.cn

Machine Learning

5.1 Perspectives about Machine Learning

5.2 Tasks in Machine Learning

5.3 Paradigms in Machine Learning

5.4 Models in Machine Learning

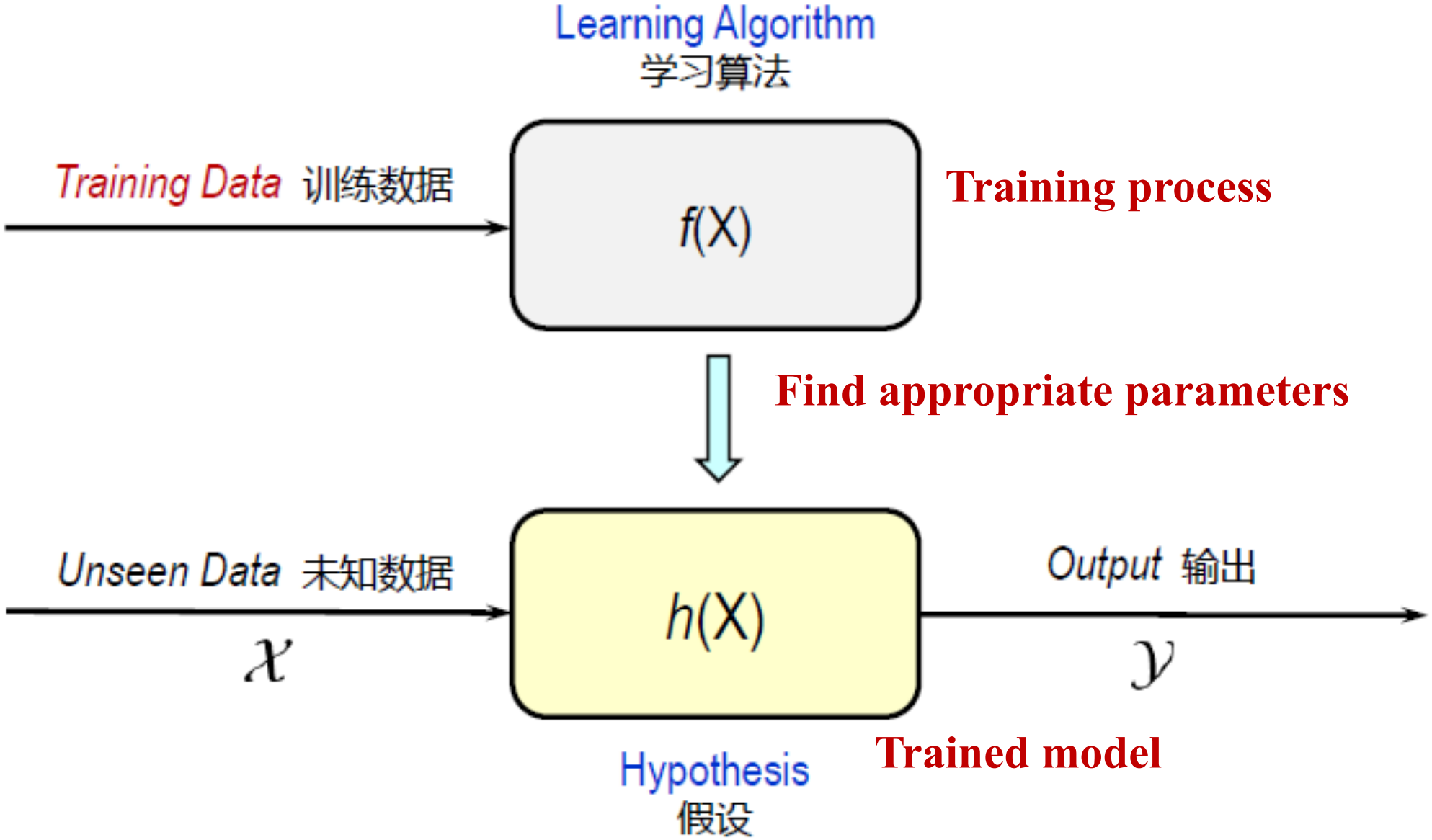
Paradigms in Machine Learning

5.3.1 Supervised Learning
5.3.2 Unsupervised Learning
5.3.3 Reinforcement Learning

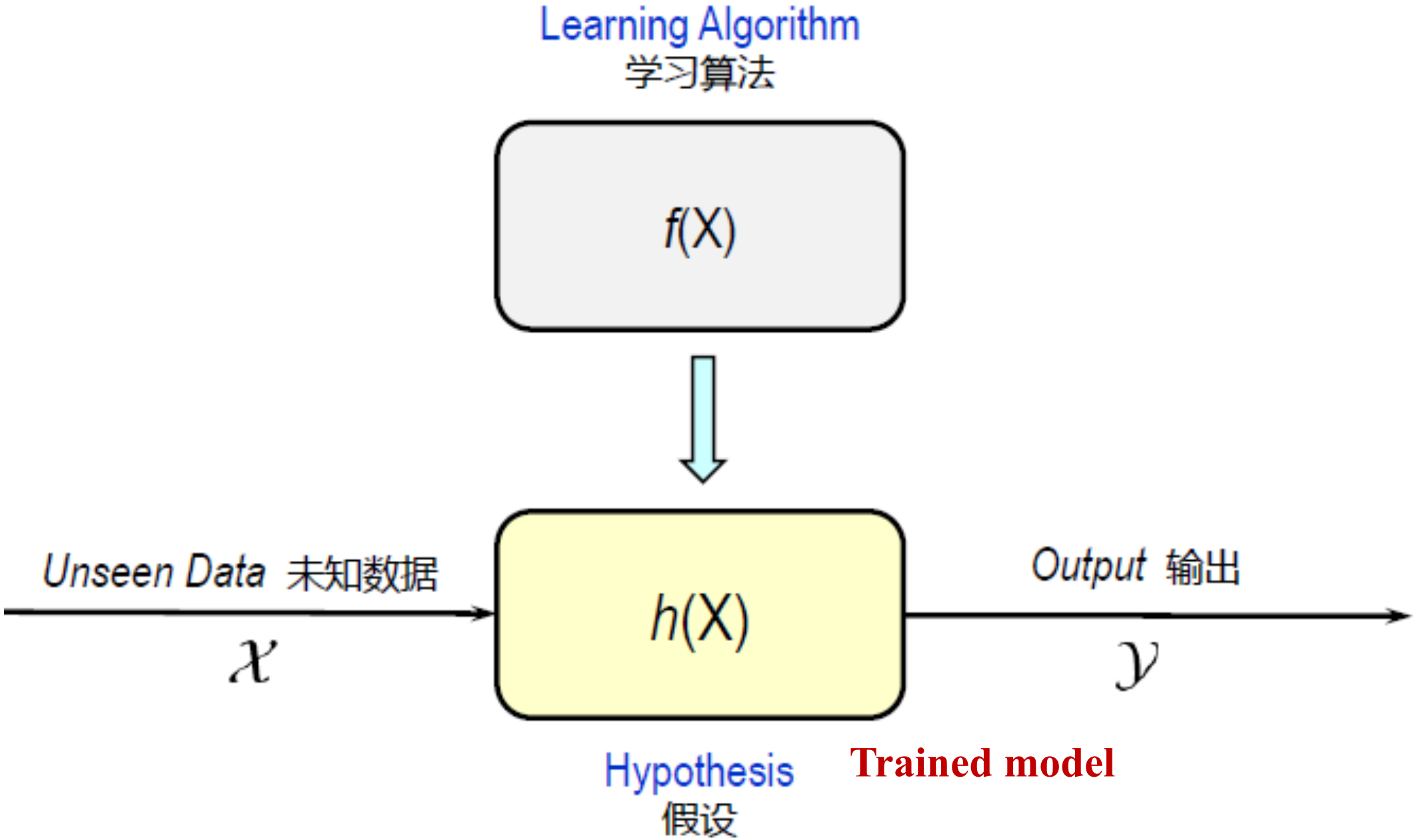
Teaching Objectives 教学目的

In this chapter we will discuss in detail about the learning paradigms that have been proposed in machine learning.

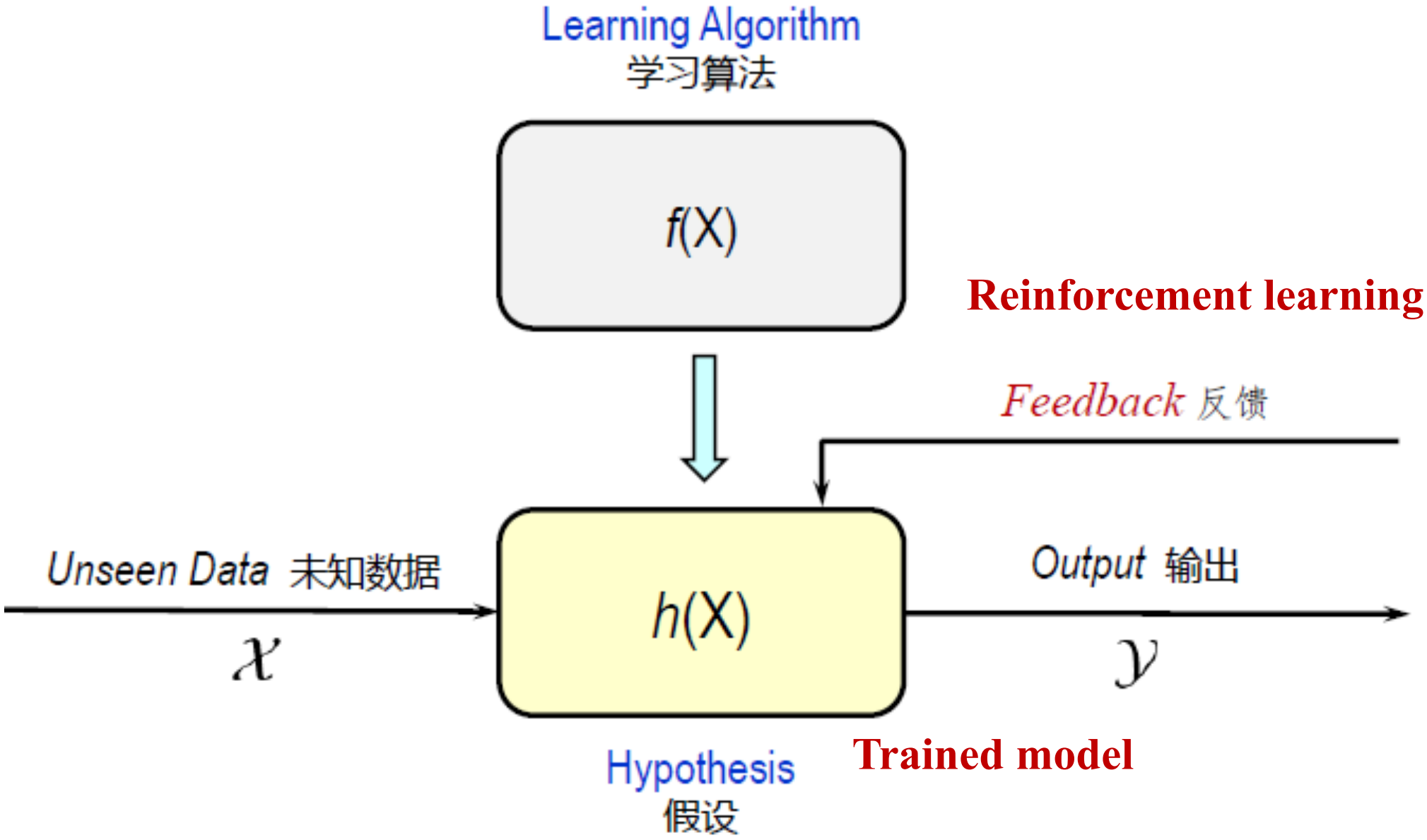
How Does Machine Learning Work
机器学习是如何工作的



How Does Machine Learning Work



How Does Machine Learning Work



Typical Paradigms in Machine Learning

Paradigms 范式	Brief Statements 简短描述	Typical Algorithm 典型算法
Supervised 有监督	The algorithm is trained by a set of labeled data, and makes predictions for all unseen points. 算法采用一组标注数据进行训练，再对所有的未知点做出预测。	Support vector machines 支撑向量机
Unsupervised 无监督	The algorithm exclusively receives unlabeled data, and makes predictions for all unseen points. 算法仅接收未标注的数据，再对所有的未知点做出预测。	*k*-means *k*-均值
Reinforcement 强化	The algorithm interacts with environment, and receives an reward for each action. 算法与外部环境交互，每个动作得到一个回报。	Q-learning

Supervised Learning Paradigm

What is Supervised Learning

- The agent receives a set of **labeled** examples as training data, and makes predictions for all unseen points.

- This approach attempts to generalize a function or mapping from inputs to outputs by training, which can then be used speculatively to generate an output for previously unknown data.

It is a way of "teaching" the learning algorithm, like that a "teacher" gives the classes (courses).

这是一种“教”学习算法的方式，就像“老师”讲授课程那样。

What is Supervised Learning

- The training data in supervised learning:
  - each training data has a **known label** as an input data,
  - the label is a **pair** consisting of an input object and a desired output value

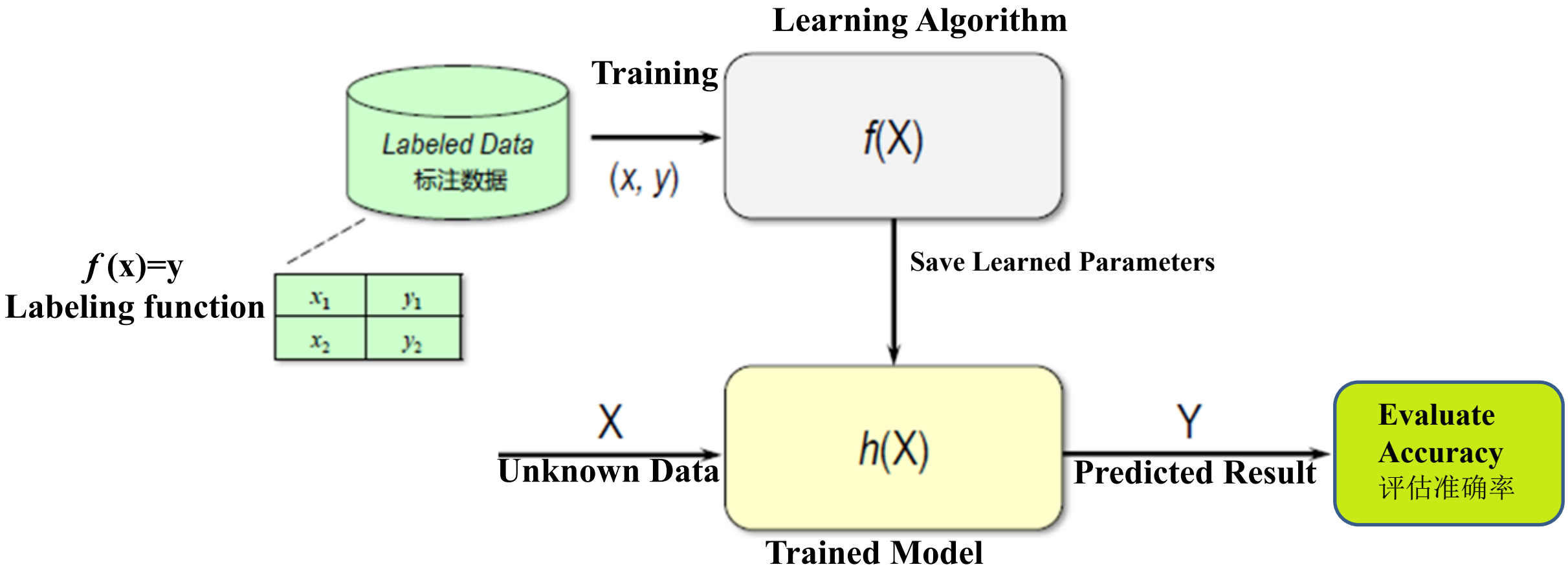
  (such as spam/not-spam, or a stock price at a time)

- An hypothesis function after training:
  can be used for mapping new unseen data.

desired output: **cat**

desired output: **dog**

Steps by Supervised Learning



Steps for Supervised Learning

1) Gather a training set and a validation set

- Collect two separate set of the labeled data to be used as a **training set and a validation set respectively**
- E.g., for handwriting recognition, that may be a single handwritten character, an entire handwritten word, or an entire line of handwriting.

2) Determine the feature extraction approach

- Typically, there are two kind of approaches to extract the feature from input data:
- *Handcrafted* feature extraction: by some feature descriptor.
  - SIFT (Scale-invariant feature transform)
  - HOG (Histogram of Oriented Gradient)
- *Automated* feature extraction: by some deep neural network.

Steps for Supervised Learning

3) Design the learning algorithm to the task

This depends on what your task is.

- E.g., for classification, you may choose to use SVM, decision tree, Softmax, etc.
- For clustering：k-means

4) Training the learning algorithm

- Run the learning algorithm on the gathered training set.
- Some algorithms require the user to determine certain control parameters.
- These parameters may be adjusted by optimizing performance on a subset of the training set.

Steps for Supervised Learning

5) Evaluate the accuracy 评估其精确性

After **parameter** adjustment and learning, the performance of the resulting function should be measured on a **validation set** that is separate from the training set.

A Formal Description for Supervised Learning

Let X denote input space, Y denote output space, and *D*an unknown distribution over X $\times$Y.

- Let target labeling function:

$$f : X \rightarrow Y$$

- Training set (a labeled object set):

$$\mathcal{S} = \{(x^{(i)}, y^{(i)}) \mid (x, y) \in \mathcal{X} \times \mathcal{Y}, i \in [1, m]\}$$

- Given a model set *H*, to find a model $h \in H$ that is the mapping:

$$h : X \rightarrow Y$$

Tasks Associated with Supervised Learning

- **Classification**
  output space Y is a set of **categories**.

- **Regression**
  output space Y is a set of **real continuous numbers**.

- **Ranking**
  output space Y is a set **with relative order**.

Some Applications of Supervised Learning

- Object recognition in computer vision
- Optical character recognition (OCR)
- Handwriting recognition
- Information retrieval
- Learning to rank
- Spam detection
- Speech recognition
- Bioinformatics
- Cheminformatics

Some Examples of Supervised Learning

- **Spam Detection**
  Mapping email to {Spam, Not Spam}
- **Digit Recognition**
  Mapping handwriting digit to{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- **Price Prediction for Used Cars**
  Mapping a used car to a real price, based on the historical data collected from used car market.

Typical Classification and Regression Algorithms

Algorithm 算法	Task Types 任务类型	Predictive accuracy 预测精度	Training speed 训练速度
AdaBoost 自适应增强	Either 两者	Higher 高	Slow 慢
Artificial neural network 人工神经网络	Either 两者	Higher 高	Slow 慢
k-Nearest neighbor *k*近邻	Either 两者	Lower 低	Fast 快
Linear regression 线性回归	Regression 回归	Lower 低	Fast 快
Logistic regression 逻辑回归	Classification 分类	Lower 低	Fast 快
Naive Bayes 朴素贝叶斯	Classification 分类	Lower 低	Fast 快
Decision tree 决策树	Either 两者	Lower 低	Fast 快
Random Forests 随机森林	Either 两者	Higher 高	Slow 慢
Support vector machines 支撑向量机	Either 两者	Higher 高	Slow 慢

K-nearest Neighbor （KNN） Algorithm

- KNN is a typical supervised learning algorithm, which can be used for both classification and regression.
- The basic idea of the algorithm is very simple.

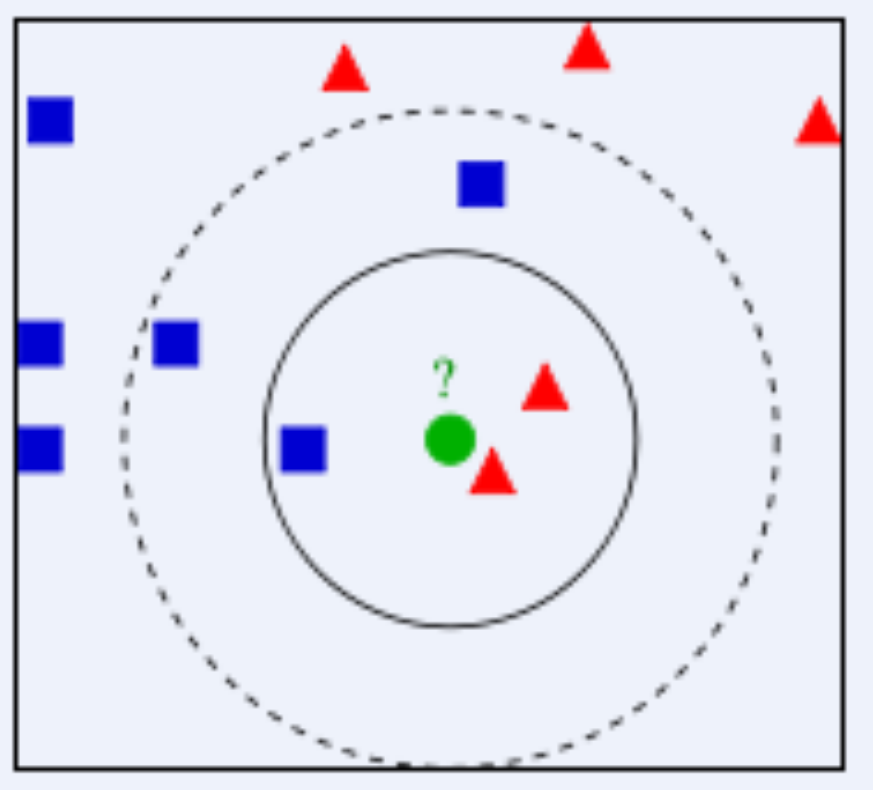
  - **Classification:** Given a sample A to be tested, the k samples most similar to A (that is, the nearest neighbor in feature space) are found in the feature space, and then the number of samples belonging to all kinds of samples is counted, and the category with the largest number of samples is found. Then sample A belongs to this class.

  - **Regression**: Find out the k most similar samples of sample A in the feature space, assign the average value of the attributes of the k samples to sample A, then we can get the attribute of sample A.

K-nearest Neighbor （KNN） Algorithm

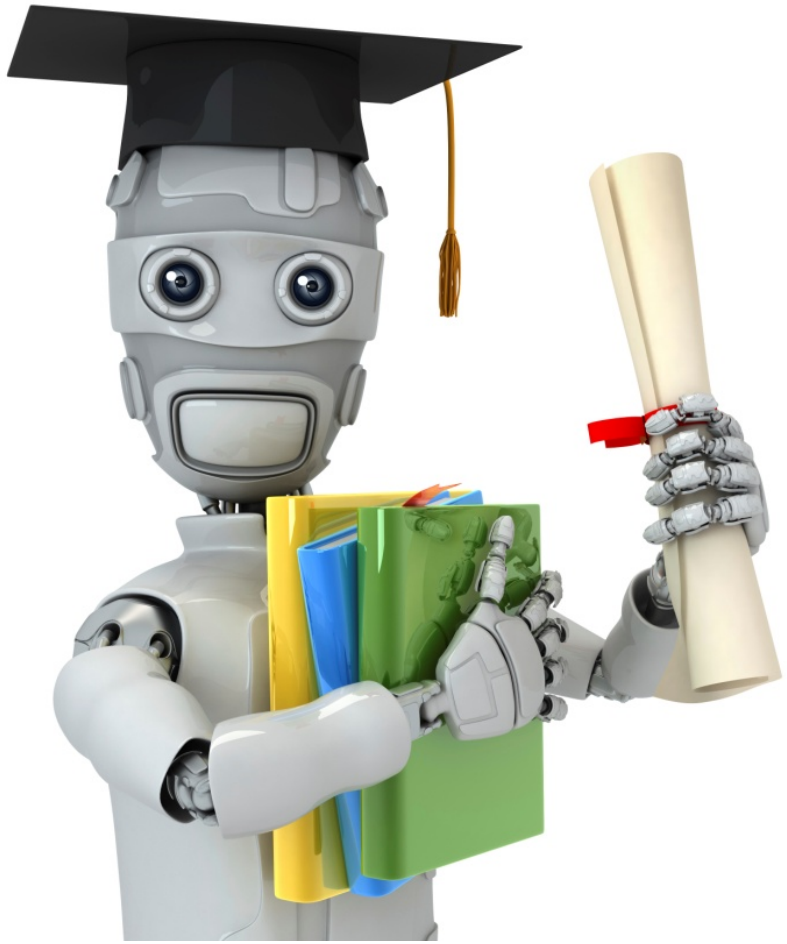
KNN algorithm process:

1. Calculate the distance from sample A to each sample in the training set
2. Arrange all samples in an ascending order of the distance from A
3. Sample A's nearest *k* training samples were selected as *k* neighbors of the test sample A.
4. Count the class frequency of these *k* neighbors
5. Find the category with the highest frequency among *k* neighbors, that is, the category of the test sample



Problem： how to determine the value of k?

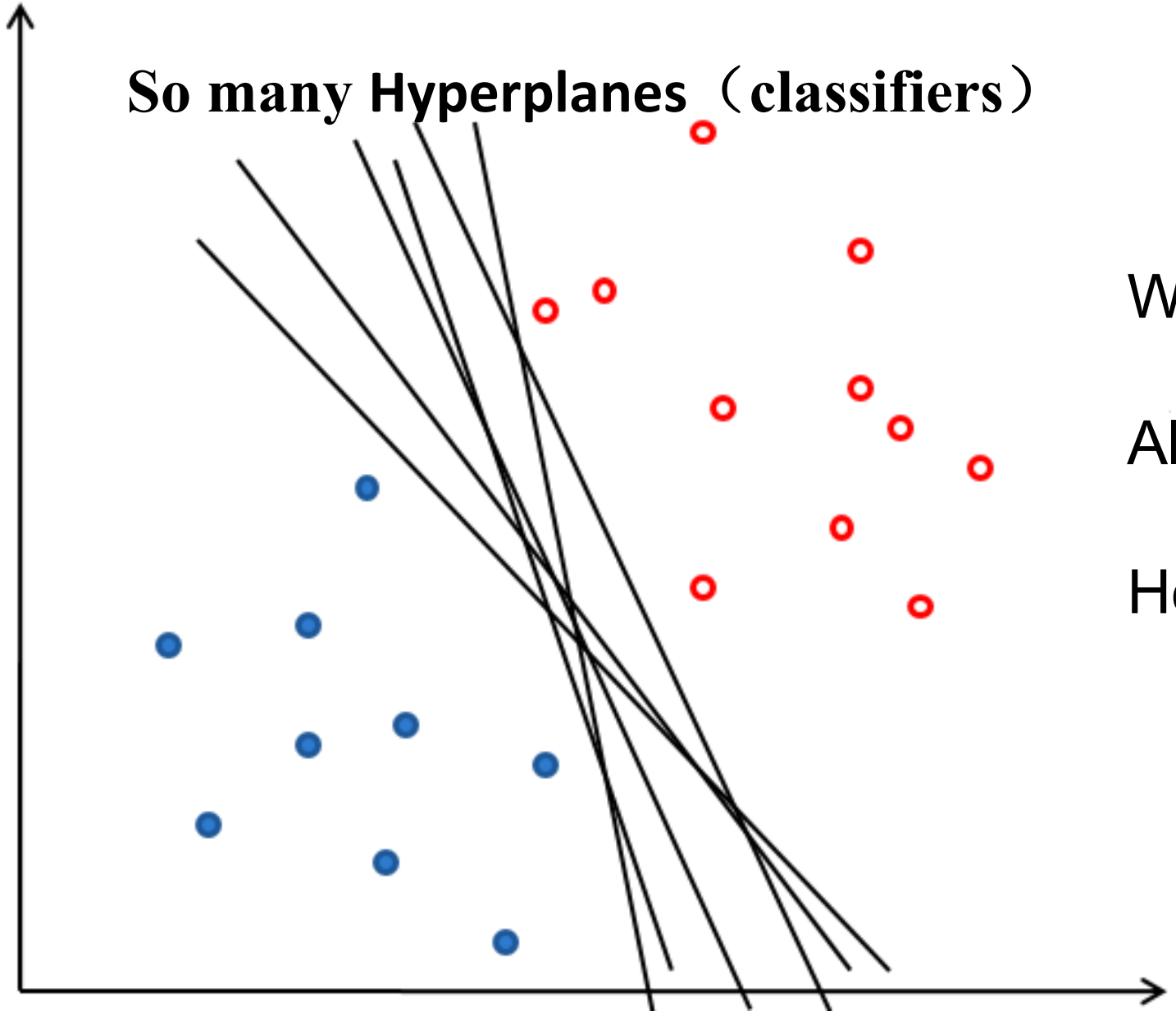
It is a empirical value.



Machine Learning

Support Vector Machines （SVM）

Selection of Classifiers

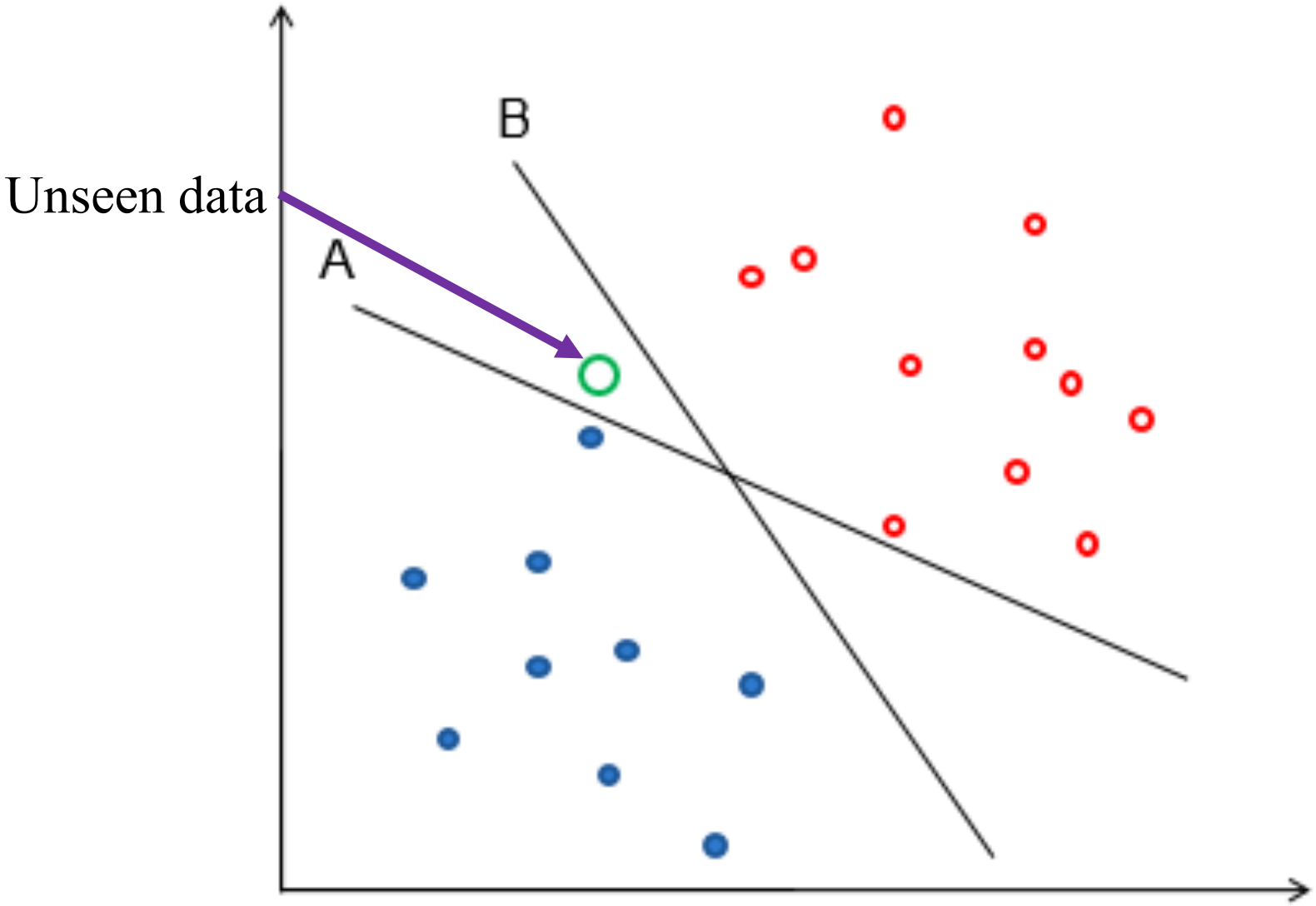


Which classifier is the best?

All have the same training error.

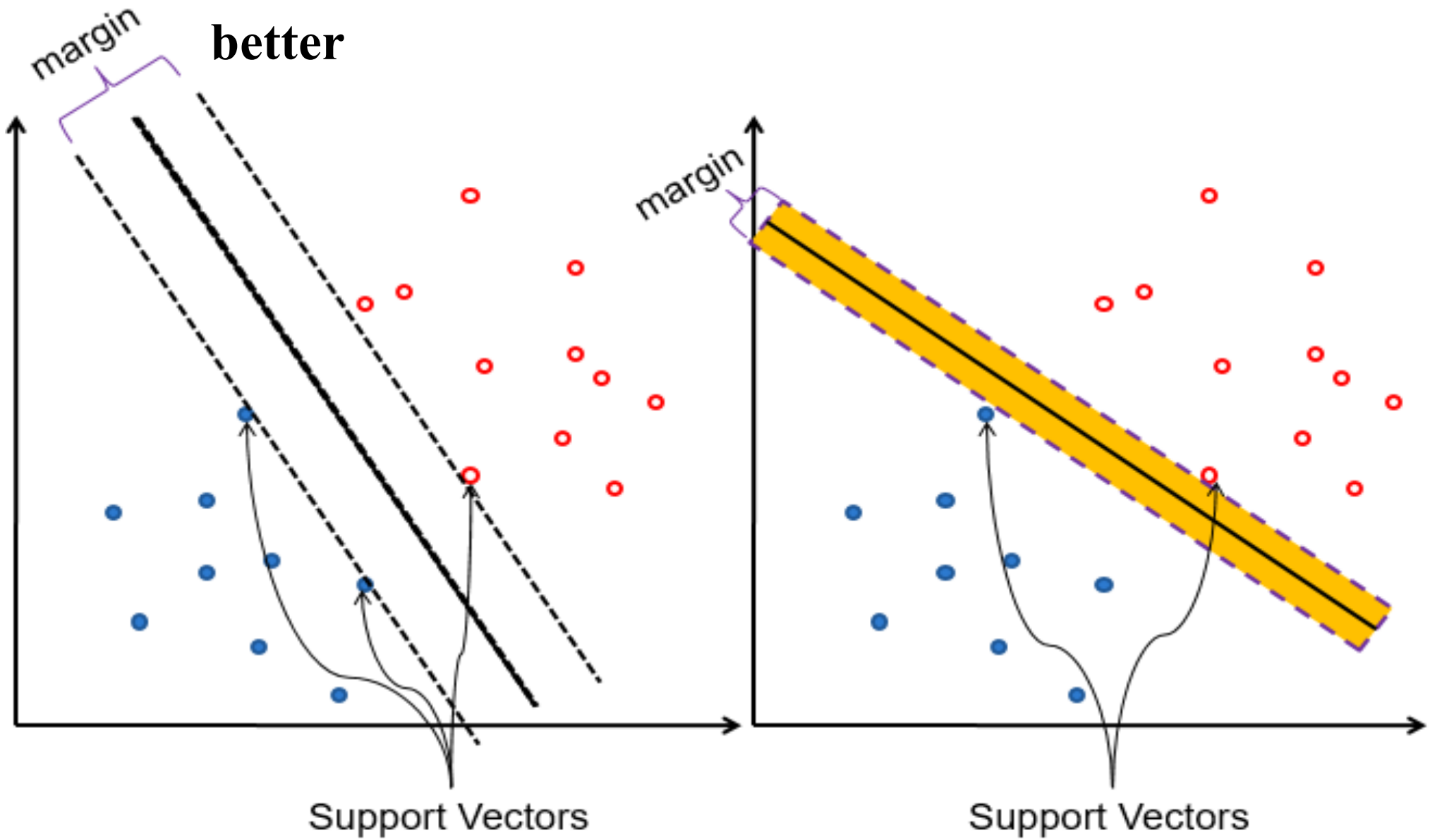
How about generalization?

Unknown objects



Classifier B is better because it divides the space more consistently (unbiased).

Margins



Margins

- The margin of a linear classifier is defined as the width that the boundary could be increased by before hitting a data point.

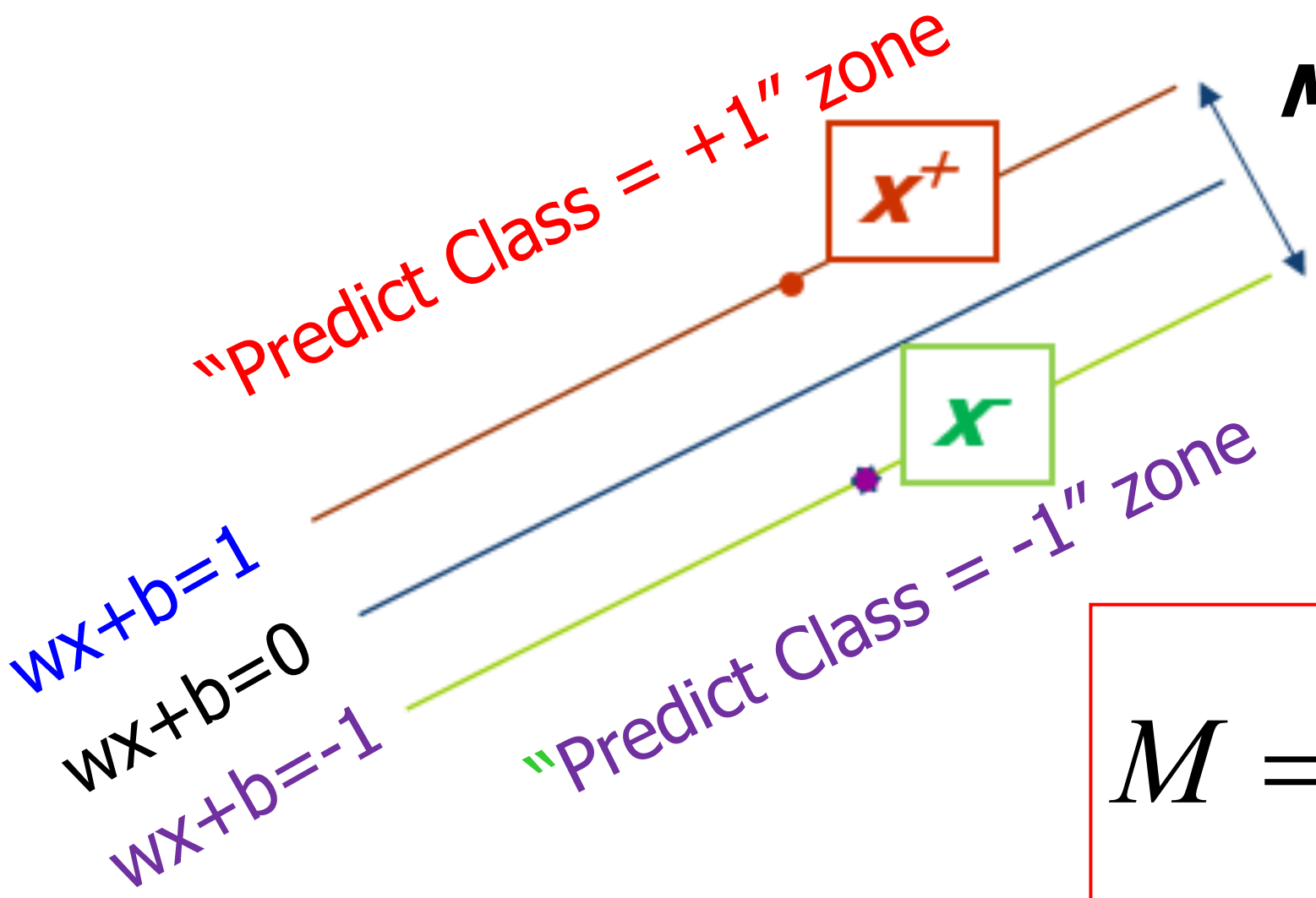
- Intuitively, it is safer to choose a classifier with a larger margin.

- The hyperplane is decided by only a few data points.

  – Support Vectors
  – Others can be discarded!
- Select the classifier with the maximum margin.

  – Linear Support Vector Machines (LSVM)
- How to specify the margin formally?

Margins



M=Margin Width

$$M = \frac{2}{\| w \|}$$

Large margin classifier

Distance between two parallel lines

Two parallel lines： ***Ax+By+c1=0*** **and**

Ax+By+c2=0

Distance between two lines:

$$\frac{|C_1 - C_2|}{\sqrt{A^2 + B^2}}$$

Two parallel lines： ***wx+b-1=0*** **and** ***wx+b+1=0***

Distance between two lines（c1=b-1, c2= b+1）：

$$M = \frac{2}{\| w \|}$$

Objective Function

- Correctly classify all data points:

$$w \cdot x_i + b \geq 1 \quad if \; y_i = +1$$
$$w \cdot x_i + b \leq -1 \quad if \; y_i = -1$$

$$y_i(w \cdot x_i + b) - 1 \geq 0$$

- Maximize the margin:

$$\max M = \frac{2}{\|w\|} \Rightarrow \min \frac{1}{2} w^T w$$

- Quadratic Optimization Problem
  - Minimize

$$\Phi(w) = \frac{1}{2} w^T w$$

  - Subject to

$$y_i(w \cdot x_i + b) \geq 1$$

Unsupervised Learning Paradigm

What is Unsupervised Learning ?

- The agent exclusively receives unlabeled data, and makes predictions for all unseen points.
- The objective is discovering commonalities in the data, or reducing the number of random variables under consideration.

It is a way of "teaching by itself", without a "teacher".

Supervised vs. unsupervised learning 有监督与无监督学习

Supervised learning

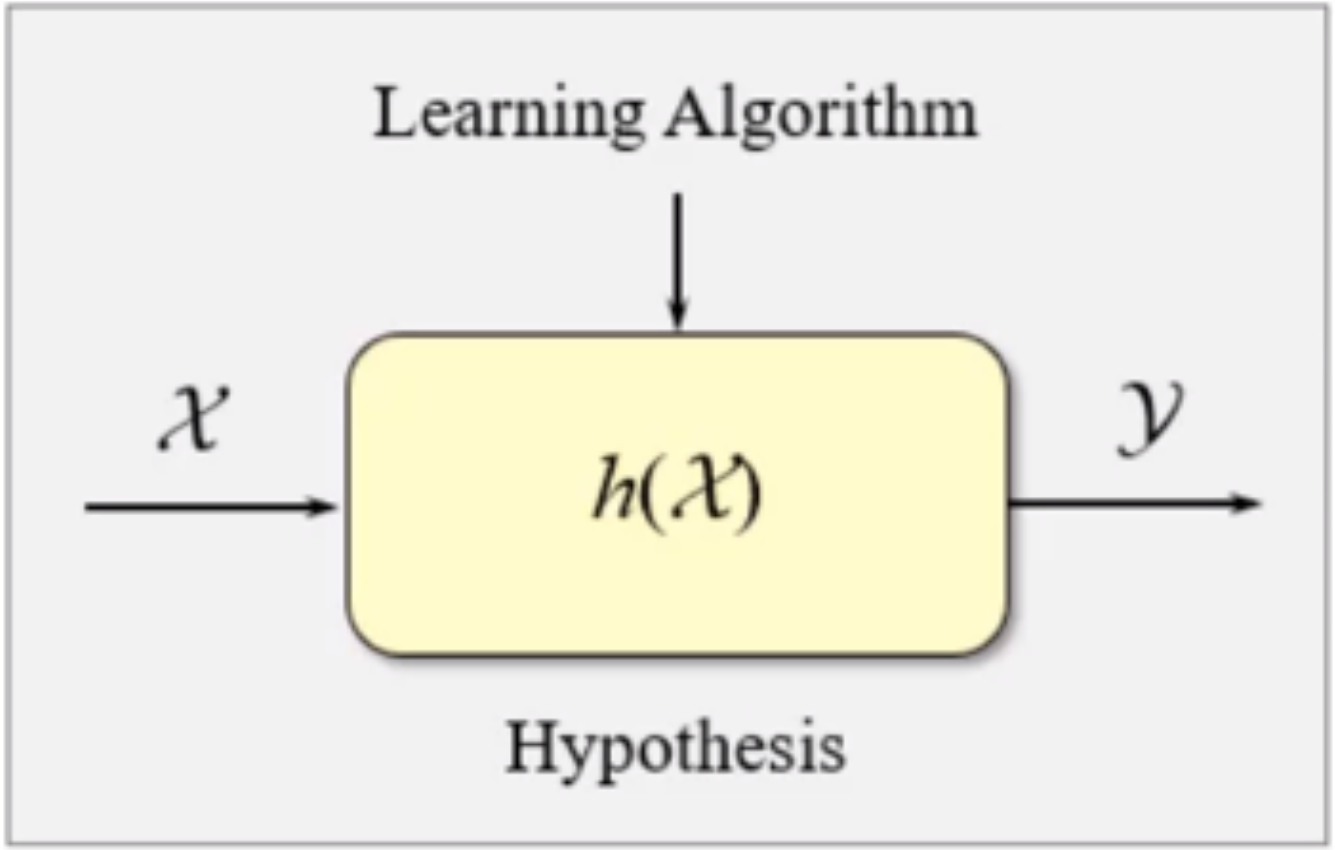
- the examples given to the learner are labeled,
- the examples are used for training the algorithm.

Unsupervised learning

- the examples given to the learner are unlabeled,
- there is no training process.

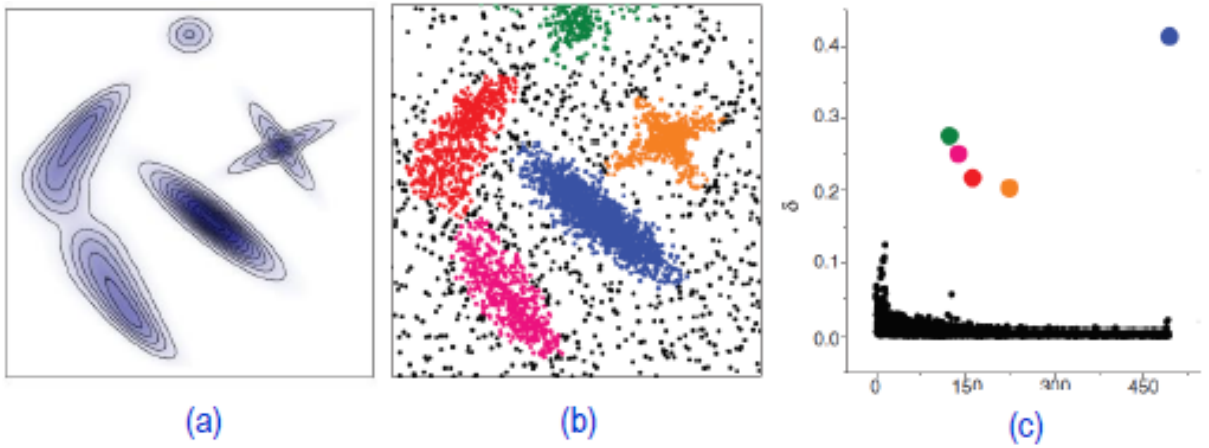
Tasks Associated with Unsupervised Learning

- **Clustering**
- Density estimation
- Dimensionality reduction



Typical Clustering Algorithms

- Single-linkage clustering
- Conceptual clustering
- ***k*-means**
- Fuzzy clustering
- Clustering by density peaks
- Clustering by density peaks



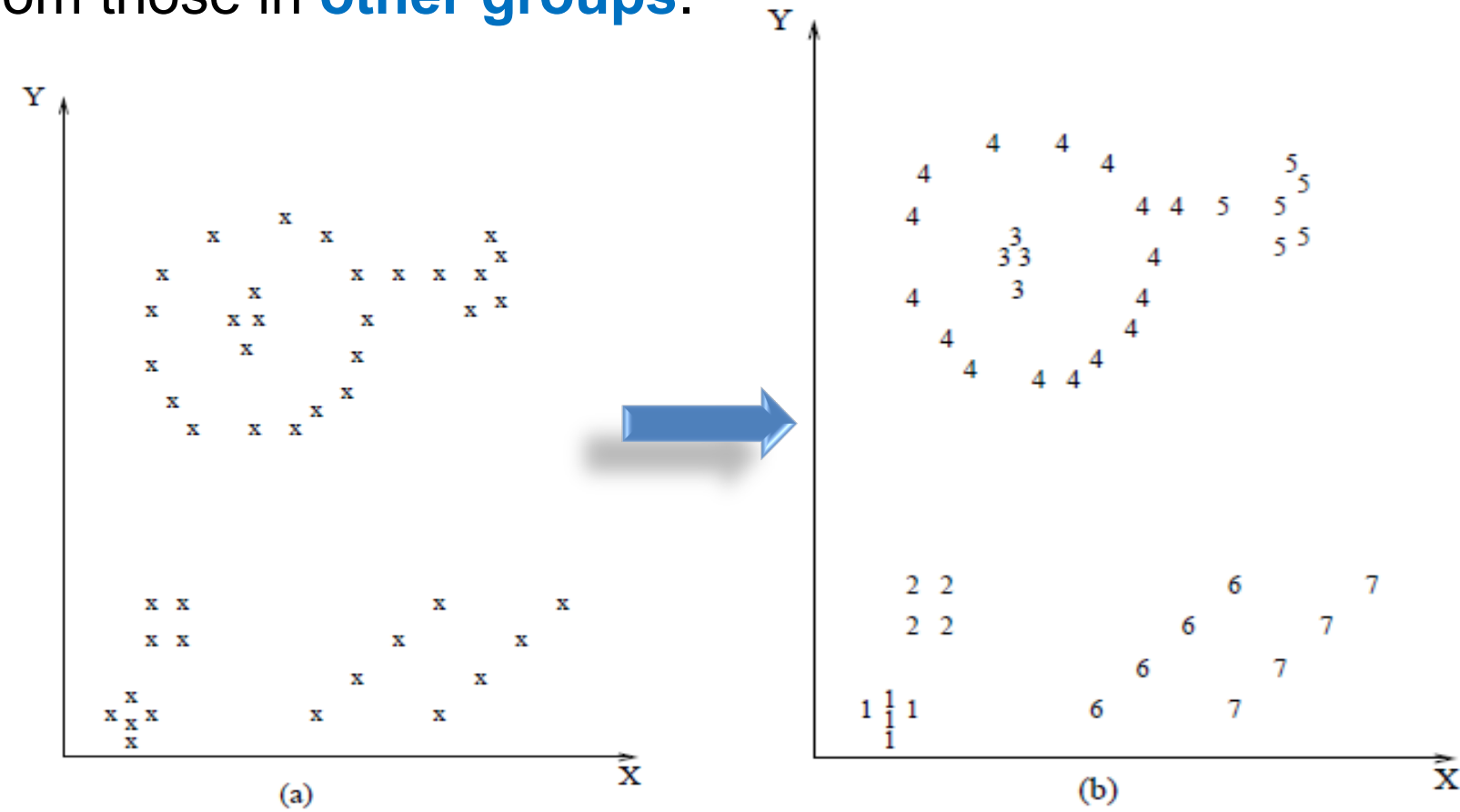
Yann LeCun's Comment

Source: Yann LeCun, "*Predictive Learning*", invited talk, NIPS 2016
(Conference and Workshop on Neural Information Processing Systems)

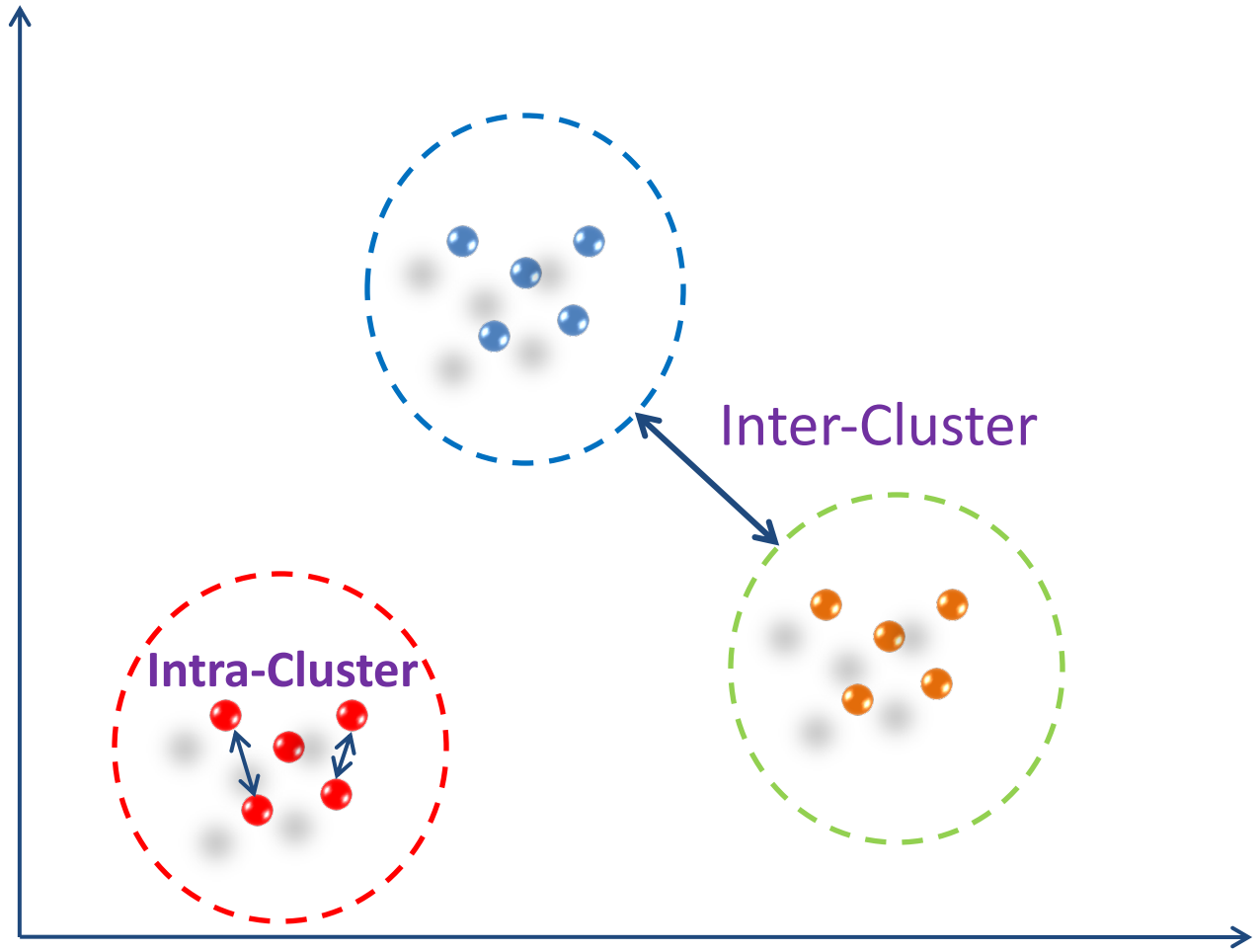
- If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake.
- We know how to make the icing and the cherry, but we don't know how to make the cake.
- We need to solve the unsupervised learning problem before we can even think of getting to true AI.

What is cluster analysis?

- **Finding groups of objects**
  - Objects **similar** to each other are in the **same group**.
  - Objects are **different** from those in **other groups**.
- **Unsupervised Learning**
  - No labels
  - Data driven



Clusters

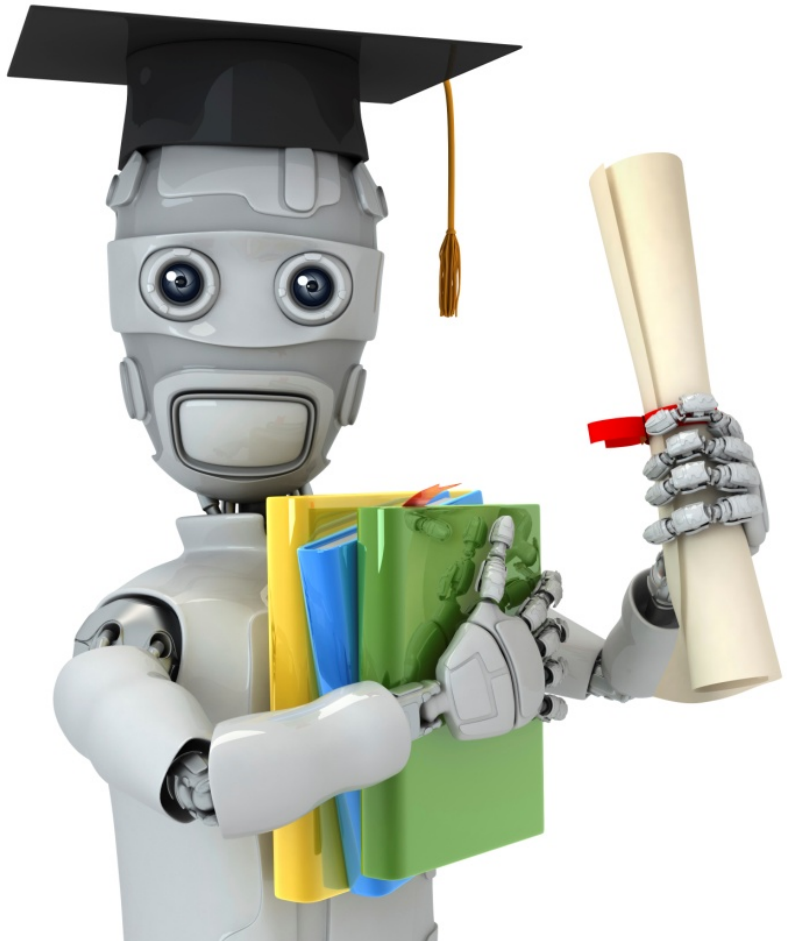


Applications of Clustering

- Marketing
  - Finding groups of customers with similar behaviours.
- Biology
  - Finding groups of animals or plants with similar features.
- Bioinformatics
  - Clustering microarray data, genes and sequences.
- Earthquake Studies
  - Clustering observed earthquake epicenters to identify dangerous zones.
- WWW
  - Clustering weblog data to discover groups of similar access patterns.
- Social Networks
  - Discovering groups of individuals with close friendships internally.

Image Segmentation

Segment the image according to the color (RGB) similarity.



Machine Learning

Clustering

K-means algorithm

K-means algorithm

- **Basic idea**: Given the parameter k as the number of clusters, *m* objects are divided into k clusters, so that the objects within one cluster have a higher similarity, while the objects between the clusters have a lower similarity.

- **Input**:
  - *K* (the number of clusters)
  - Training set *D*= $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ $x^{(i)} \in \mathbb{R}^n$
- **Output**: *K* clusters

K-means algorithm

- The processing of the k-means algorithm is as follows

(1) From the data set *D*, k objects are randomly selected as the centroids of the initial clusters;

(2) The distance between each object and the centroids of k clusters are calculated, partition the object into the nearest cluster.

(3) Recalculate the centroids of *k* new clusters (that is, the average of all data points in the cluster)

(4) Repeat steps (2)-(3) until the objects in all the clusters no longer change.

K-means algorithm

- Generally, the square error criterion is used to minimize the sum of squares of the Euclidean distance from each object to the nearest centroid.

$$J_e = \sum_{i=1}^{k} \sum_{x \in D_i} \left\| x - C_i \right\|^2, \qquad C_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

where ***k*** is the number of clusters, C_i is the centroid of the i^{th} cluster, J_e is the square error.

- **Goal：** finally achieve such a state that "The points within the same class are close enough, and the points between different classes are far enough".

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Repeat { //m is the number of samples

for i = 1 to m // find the cluster centroid $c^{(i)}$ closest to the i^{th} object

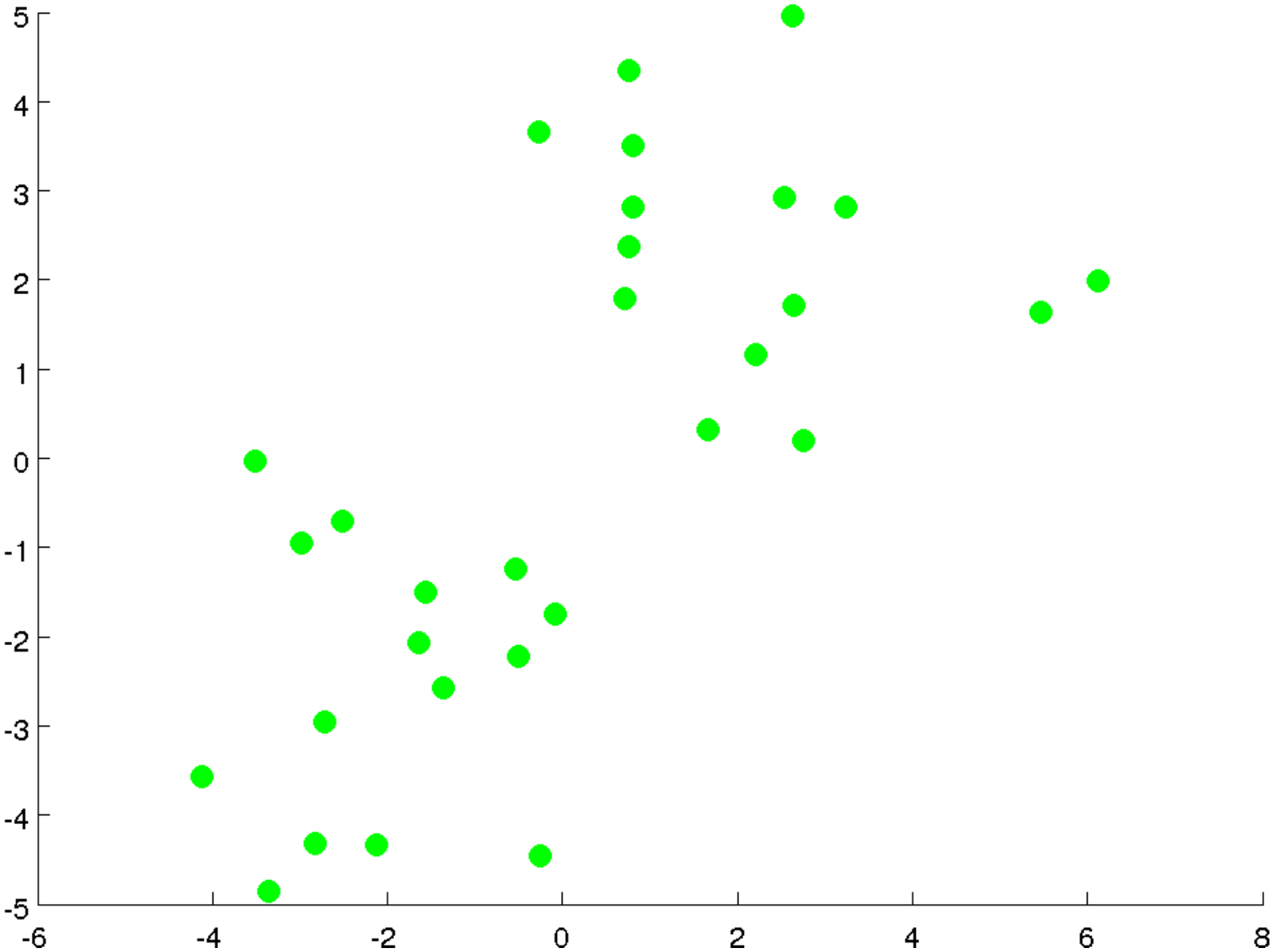
$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for k = 1 to K //recalculate the centroids of K clusters

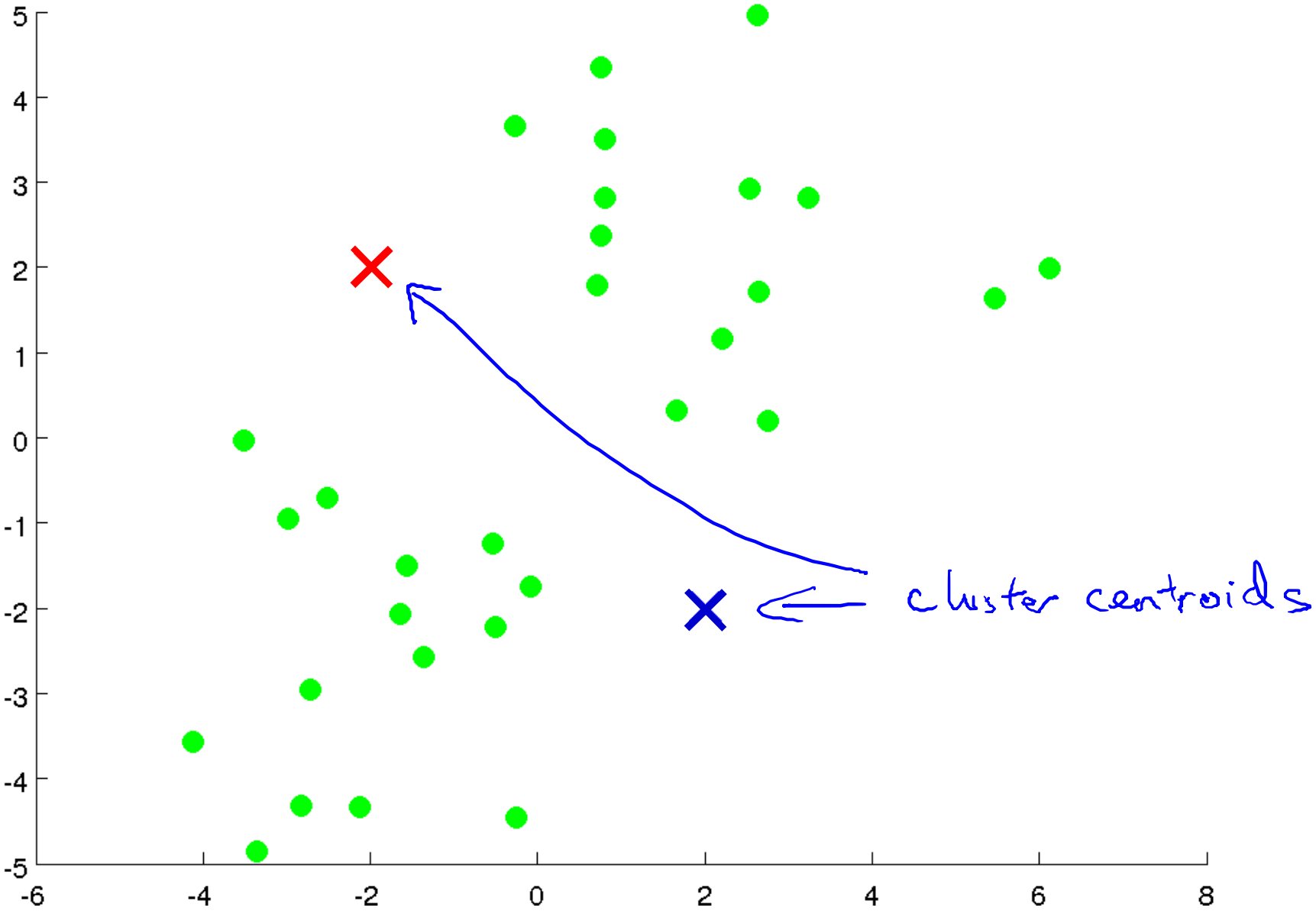
μ_k := average (mean) of points assigned to cluster k

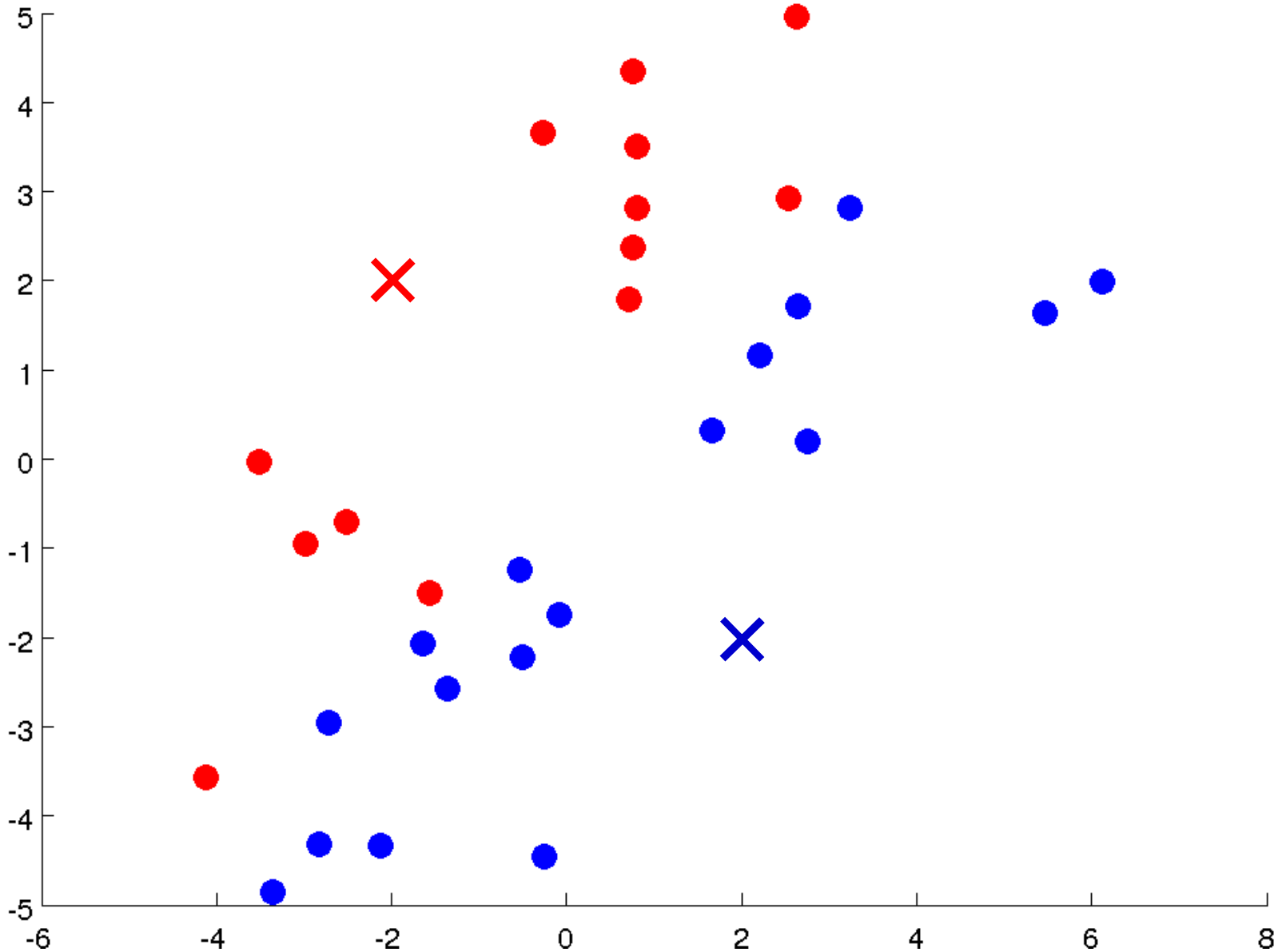
}

K=2

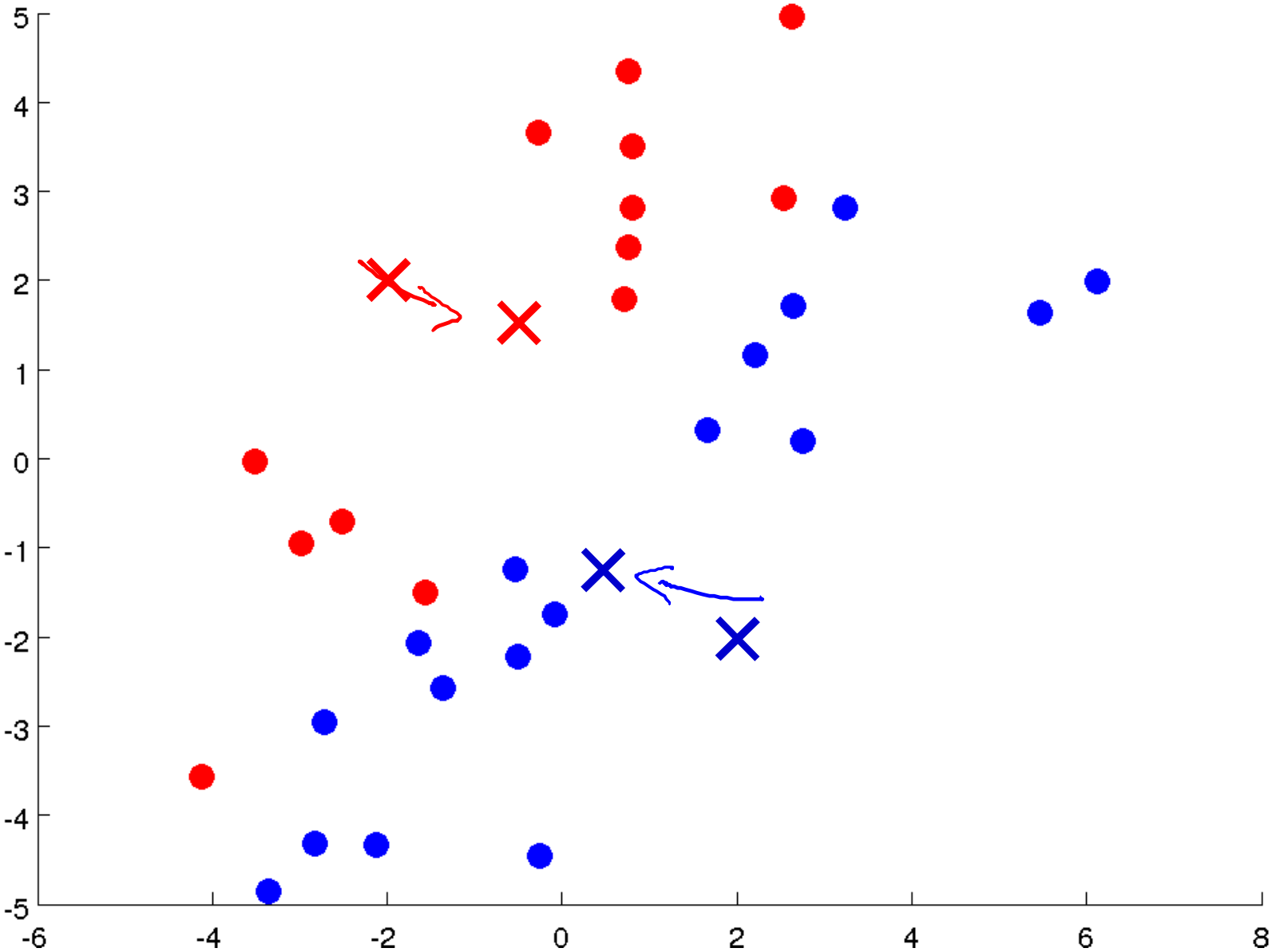


K=2

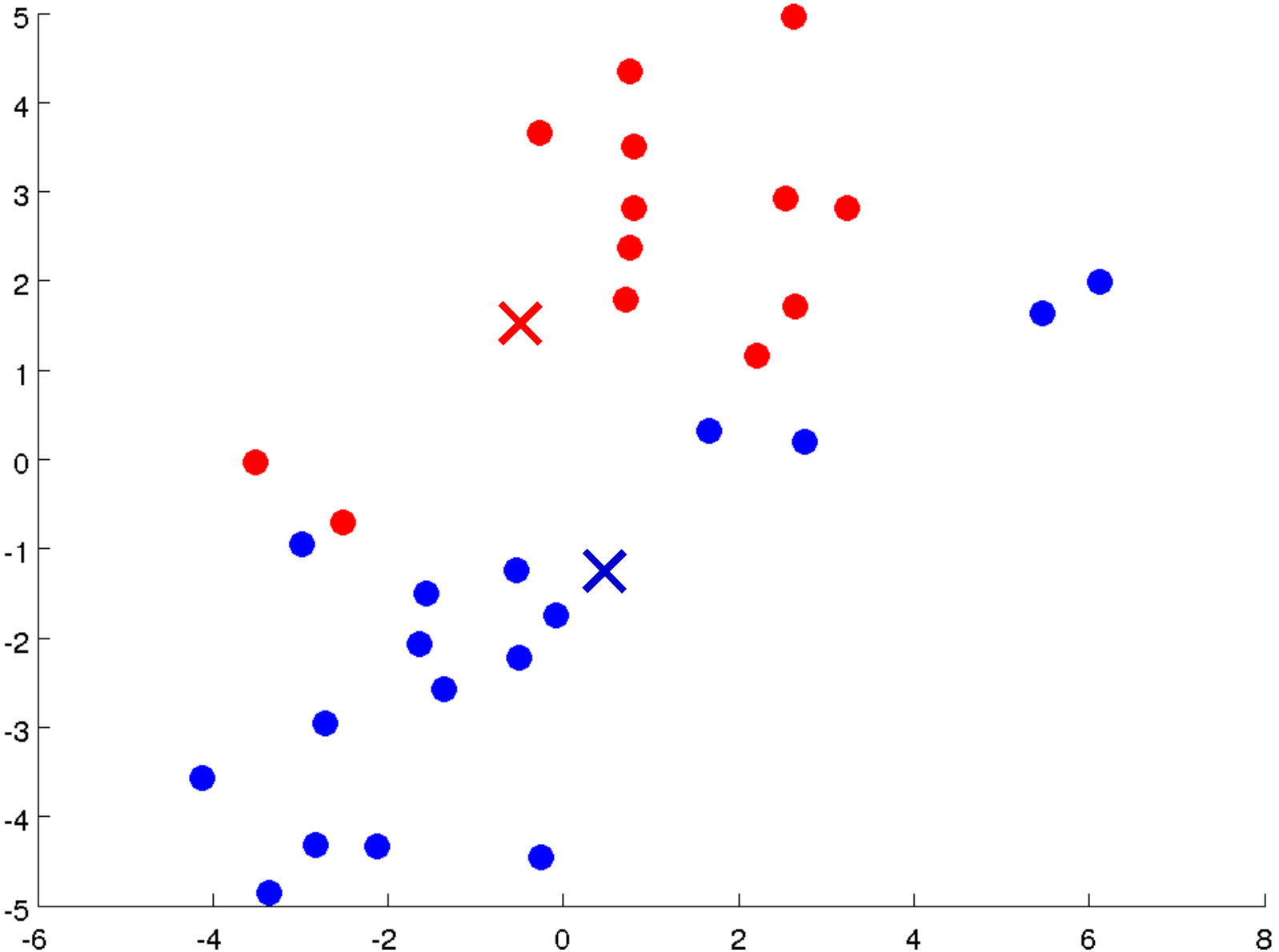



5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

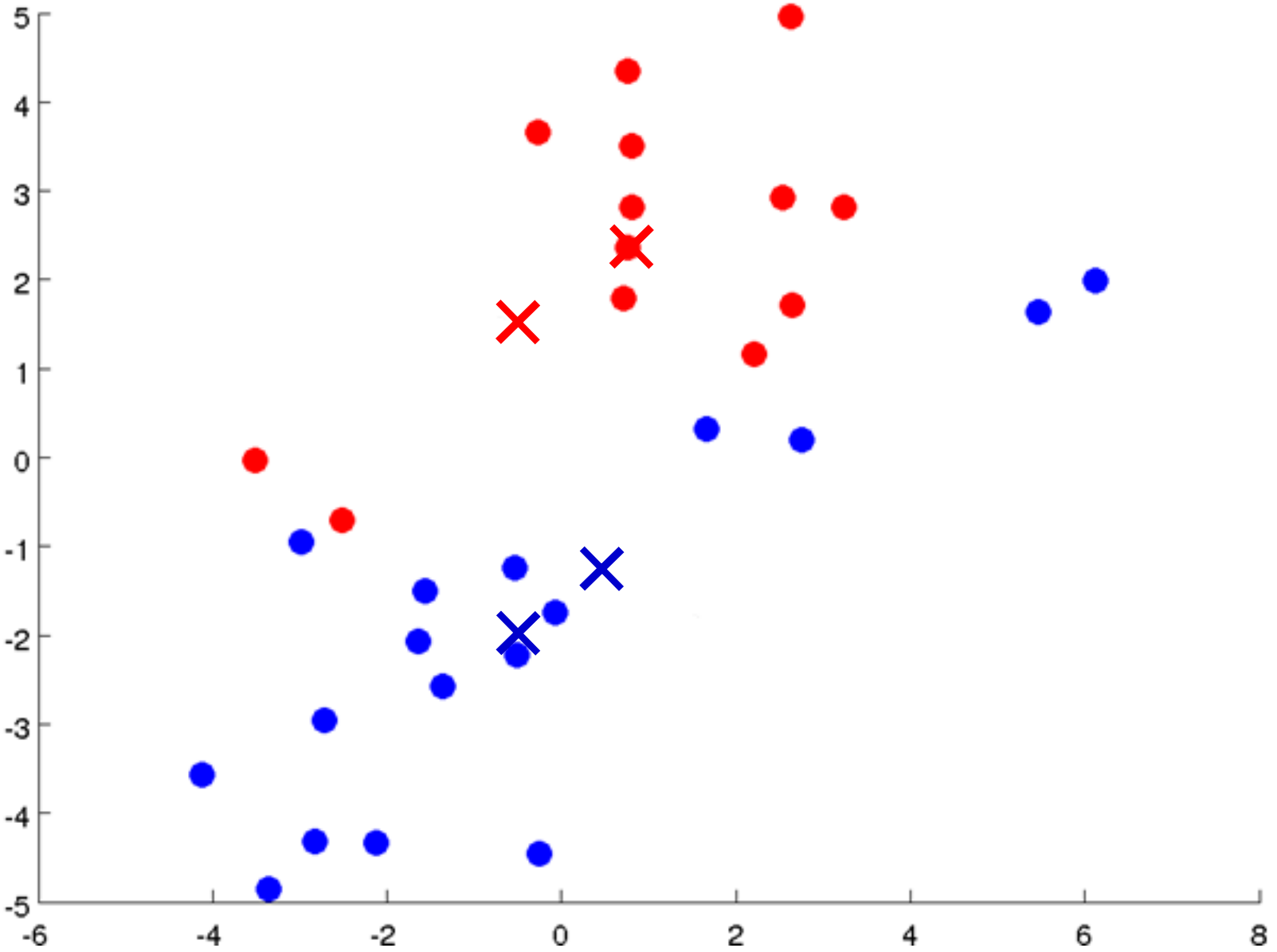
K=2



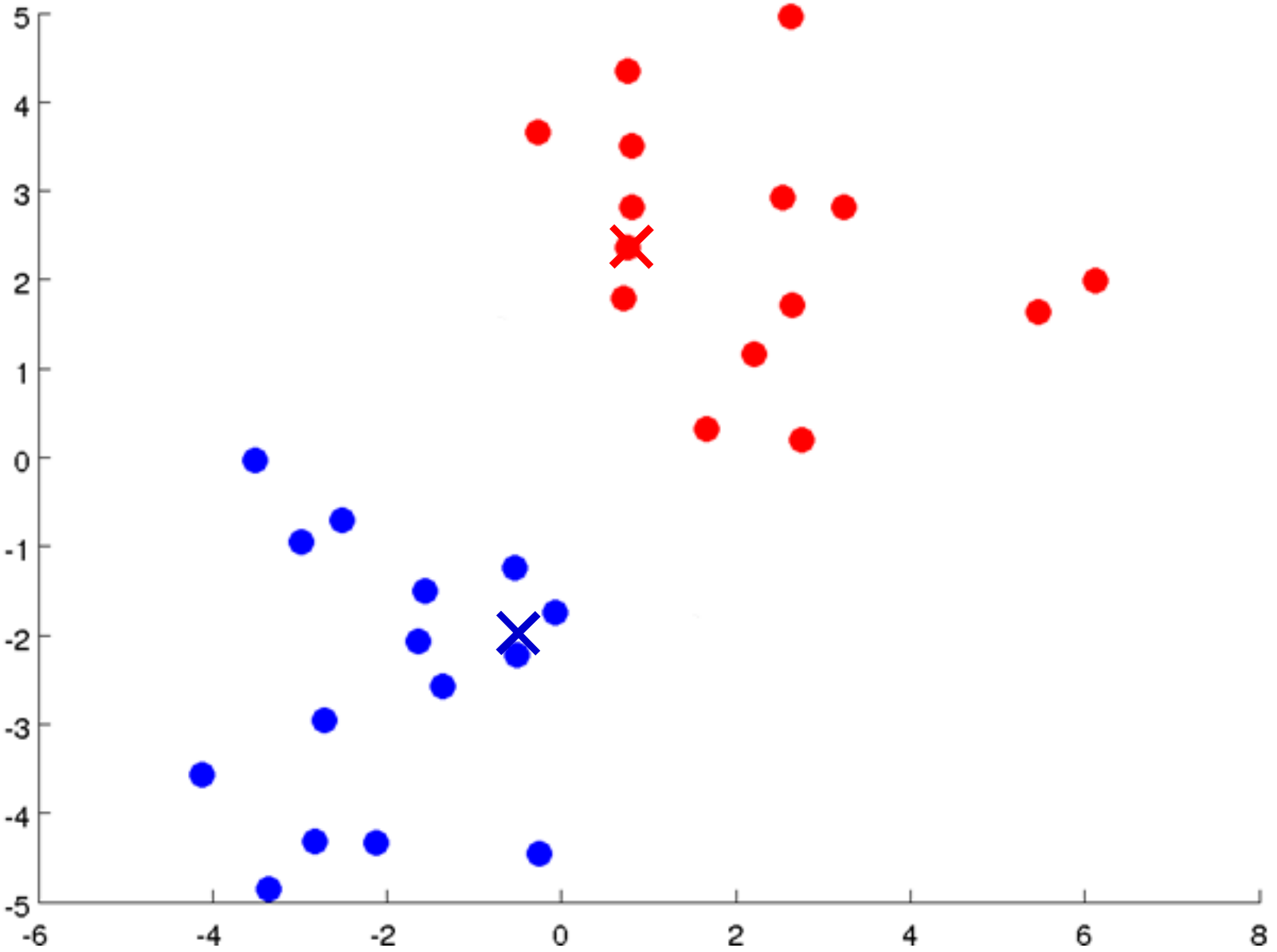
K=2



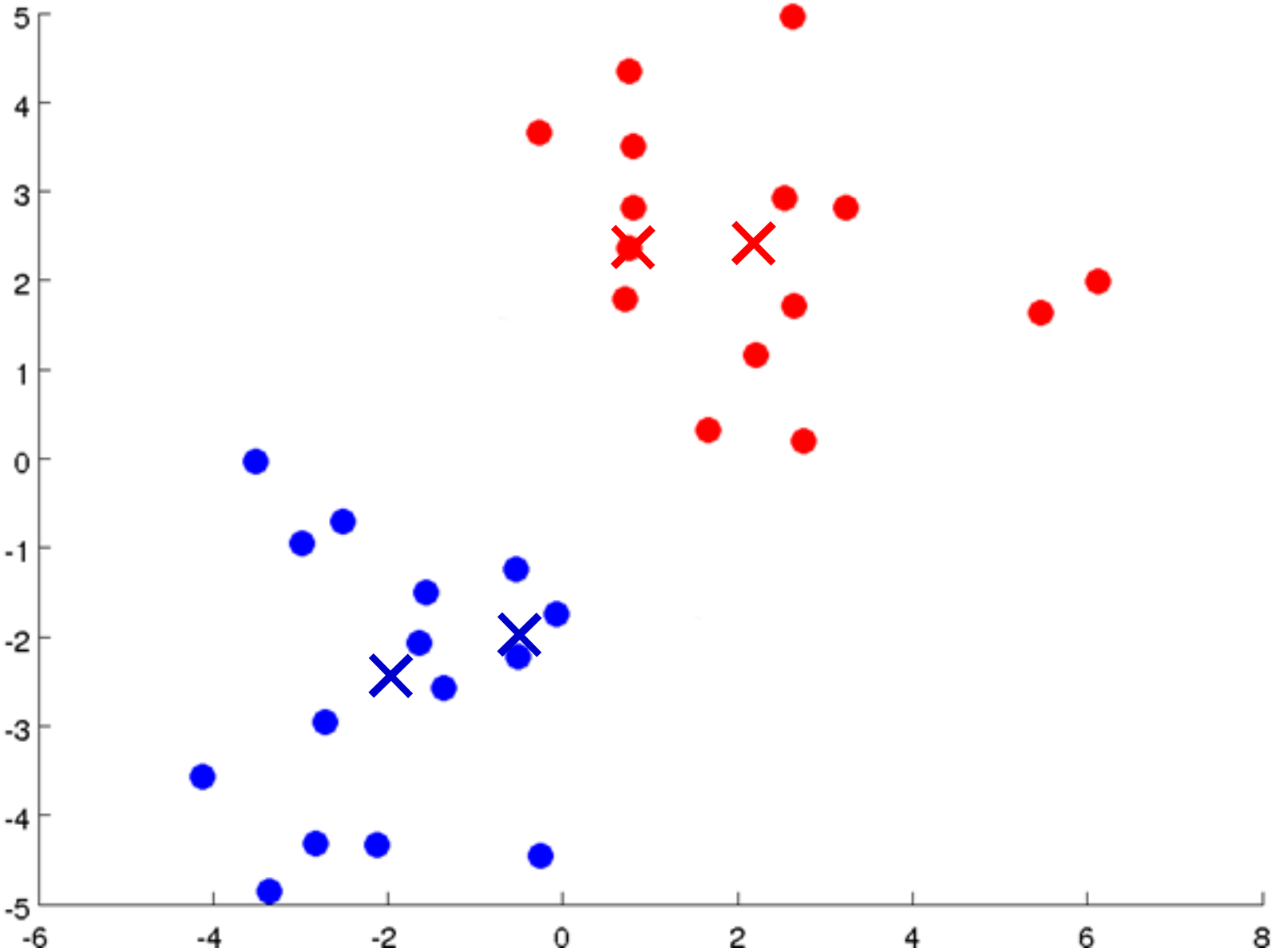
K=2



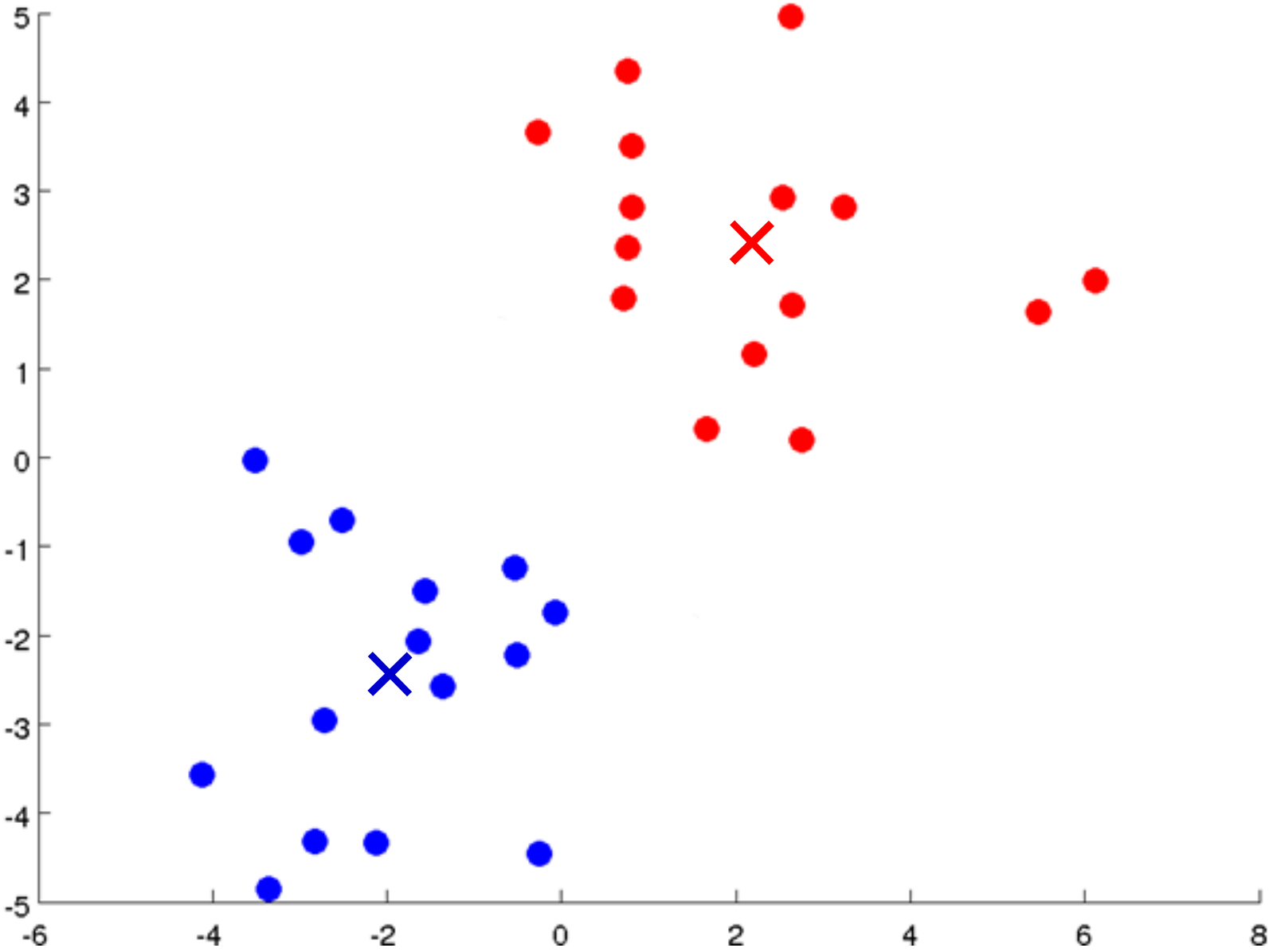
K=2



K=2



K=2



Reinforcement Learning Paradigm

- Overview of Reinforcement Learning
- Types of Reinforcement Learning
- New Algorithms of Reinforcement Learning
- Applications of Reinforcement Learning

What is Reinforcement Learning

- Reinforcement Learning is inspired by behaviorist psychology.
- Concerned with how agents take actions in an environment so as to maximize the cumulative reward.
- The reinforcement signal provided by the environment in RL is a kind of evaluation (usually scalar signal) of the action produced by agent, rather than telling agent how to produce the correct action.
- Because the external environment provides little information, agent must learn from its own experience.
- In this way, agent acquires knowledge in an action--evaluation environment and improves action to adapt to the environment.

Reinforcement learning

Learn to ride a bicycle, but nobody teach how to ride
(learn by oneself)

Trial and Error

+reward：ride stably

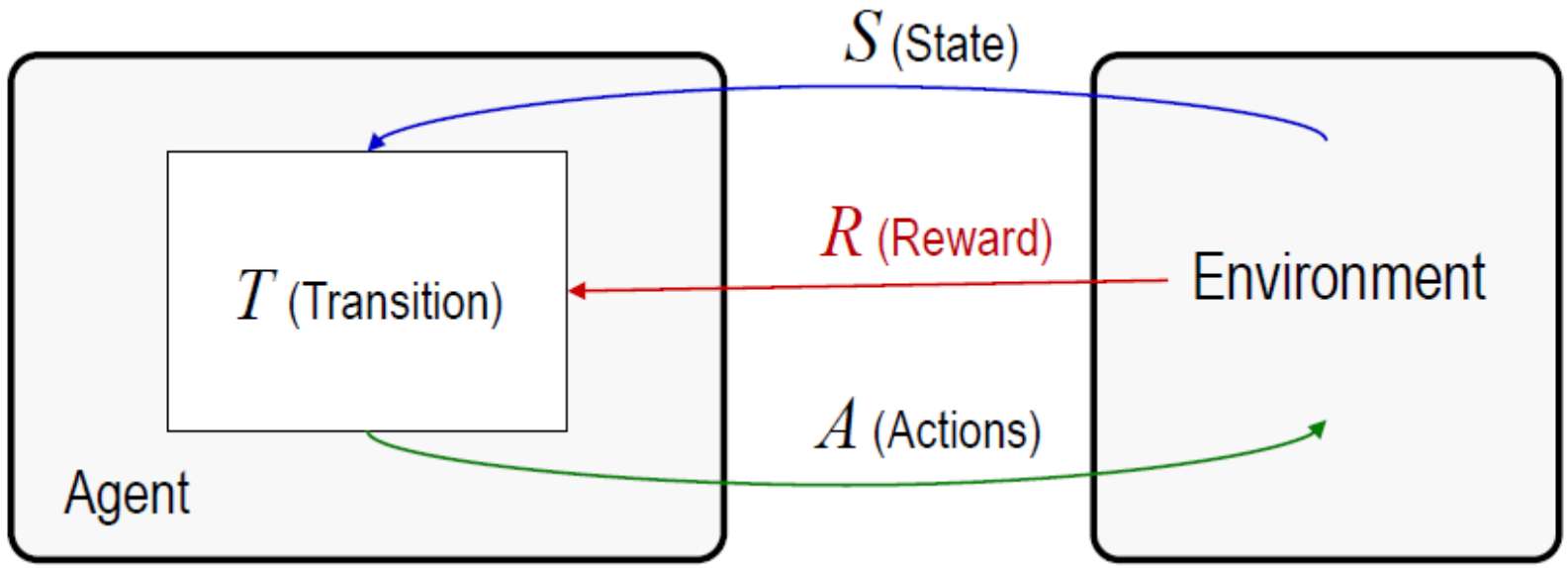
-punish：fall

The process of maximizing rewards

Basic idea of RL: get more rewards for the next action by constantly trying and **maximize rewards.**

What is Reinforcement Learning

- In reinforcement learning (RL), the learner is a decision-making agent, that takes actions in an environment and receives rewards for its actions.
- After a set of trial-and-error runs, the agent should learn the best policy.
- The policy is to maximize his reward over a course of actions and iterations with the environment.



Formalization of Reinforcement Learning

- Reinforcement learning consists of:
  - a set of agent **states**, $s_t \in S$;
  - a set of the agent **actions**, $a_t \in A$;
  - a **transition** from states to actions, $\boldsymbol{T(s_t, a_t, s_{t+1})}$
  - a **reward** function, $\boldsymbol{R(s_t, a_t, s_{t+1})}$.
- To look for a policy, $\pi(s_t)$.
- Don't know *T*or *R*
  - i.e. don't know which states are good or what the actions do.
  - Must actually try actions and states out to learn.
  - learning from state and reward, take better action to the environment.

Supervised vs. Unsupervised vs. Reinforcement Learning

- ***Supervised learning***
  - Input/output pairs are presented by **labeled data** (training examples).
  - *Learn-by-examples*
- ***Unsupervised learning***
  - To find the structure hidden in collections of unlabeled data.
  - *Learning-by-itself*
- ***Reinforcement learning***
  - Input/output pairs are never presented, focus on online performance.
  - *Feedback-learning*

Types of Reinforcement Learning

1) **Model-based**

building a model of the environment.

- First acting in Markov decision process (MDP)and learning *T*, *R*;
- Then doing value iteration or policy iteration with learned *T*, *R.*

2) **Model-free**

learning a policy without any model.

- Bypassing the need to learn *T*, *R*, using direct evaluation policy.
- Prediction-based temporal difference (TD) methods.

Q-learning

Q-learning is a Model-free Method.

- Q-learning uses Q estimation to make decisions and update learning.
- Q estimation, denoted as Q (*s*, *a*), is the expectation obtained by taking an action *a* under a state *s*;

- The environment will give feedback to Agent with a reward *r* based on the actions taken by the agent.

- The main idea of Q-Learning is to construct a Q-table using states and actions and store the Q value, and then select the actions which can obtain the maximum benefit according to the Q value.

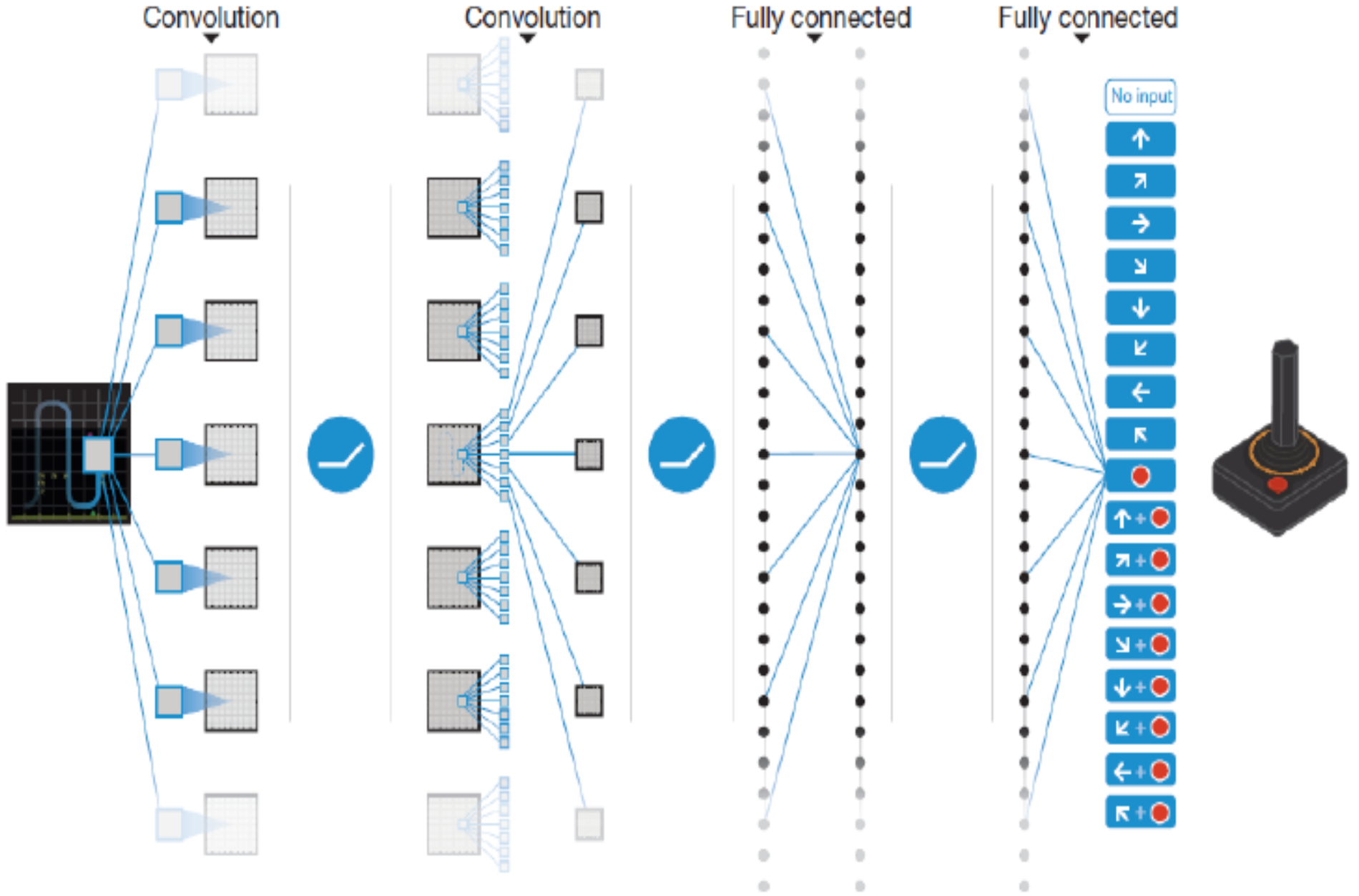
New Algorithms of Reinforcement Learning

- Feb. 2015, Google DeepMind published **Deep Q-Network**, the human-level control through deep reinforcement learning.

- **Deep Q-Network (DQN)**
  It combines CNN with Q-learning, (NIPS'13, Nature'15).

Case Study: Deep Reinforcement Learning

The input is raw pixels and the output is a value function estimating rewards.



Typical Applications of Reinforcement Learning

- **Robots**
  - Robotic arms
    be controlled to find the most efficient motor combination.
  - Robot navigation
    collision avoidance behavior can be learned by negative feedback.
- **Computer games**
  - Backgammom,
  - Chess,
  - Go.

Models in Machine Learning

- **Probabilistic Models**
- **Geometric Models**
- **Logical Models**
- **Networked Models**
  - **Artificial Neural Networks (ANN)**
  - **Convolutional Neural Networks (CNN)**
  - **Deep Neural Networks (DNN)**