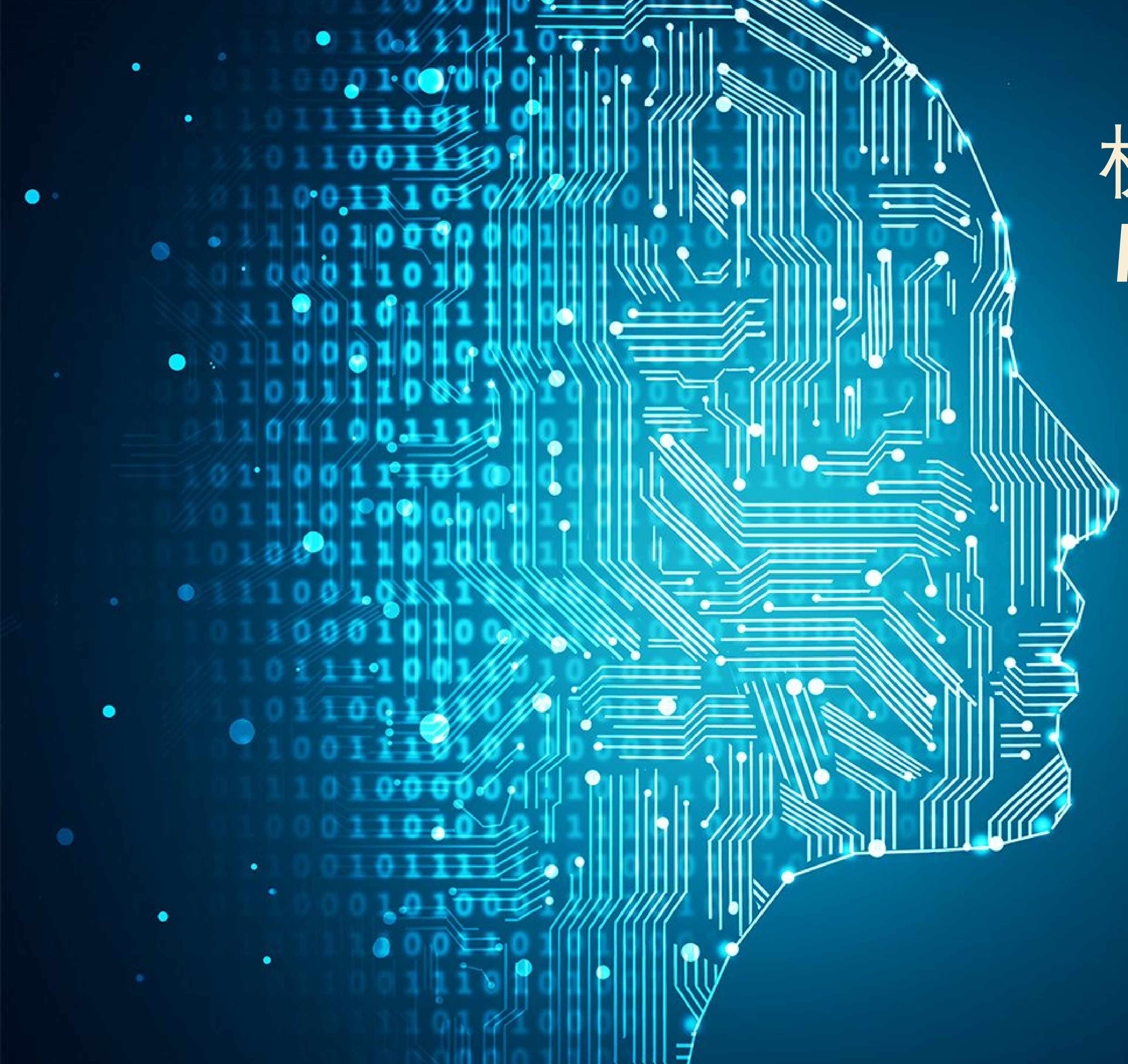


机器学习与人工智能
Machine Learning and Artificial Intelligence

Lecture 11 Review

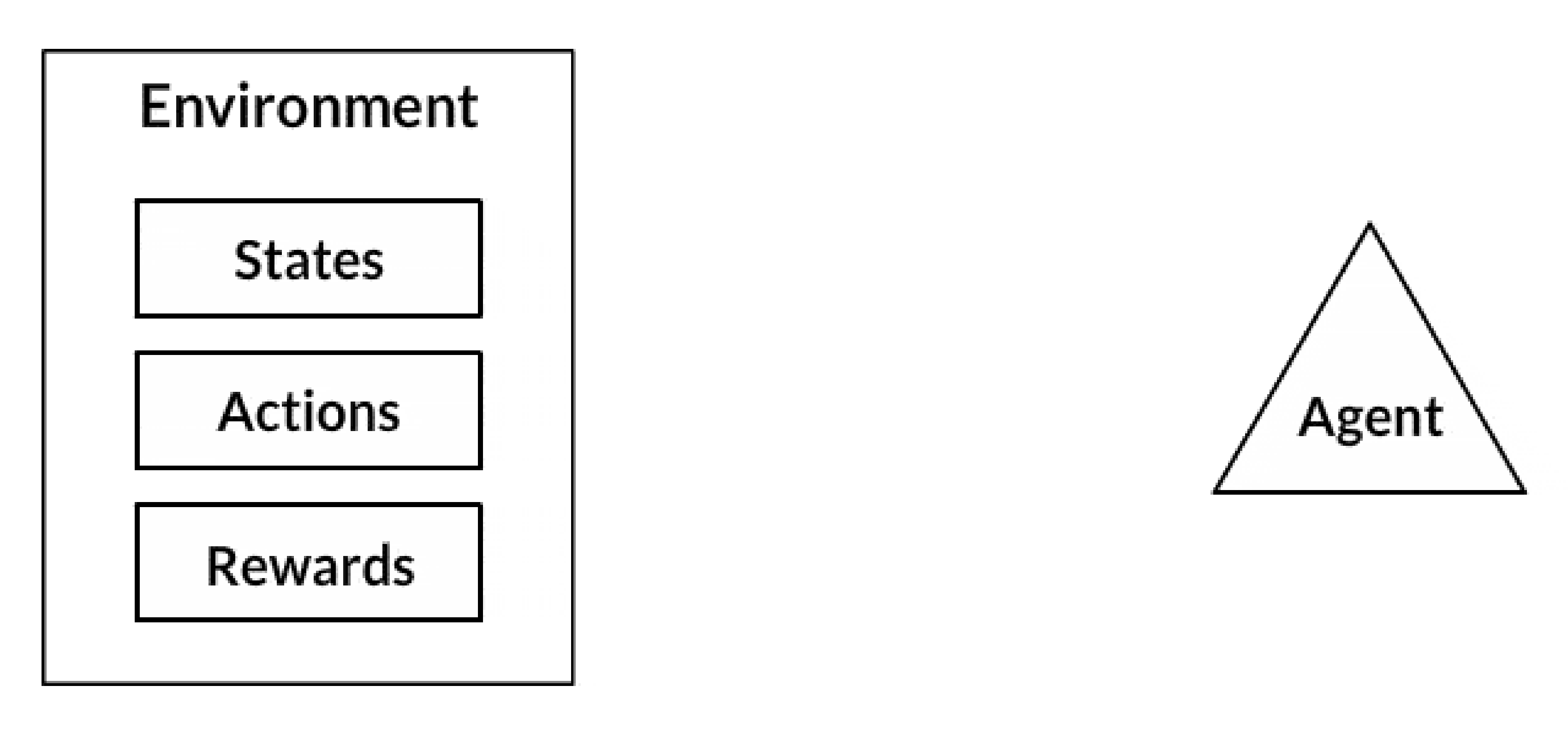
Yingjie Zhang (张颖婕)

Peking University

yingjiezhang@gsm.pku.edu.cn

2021 Fall

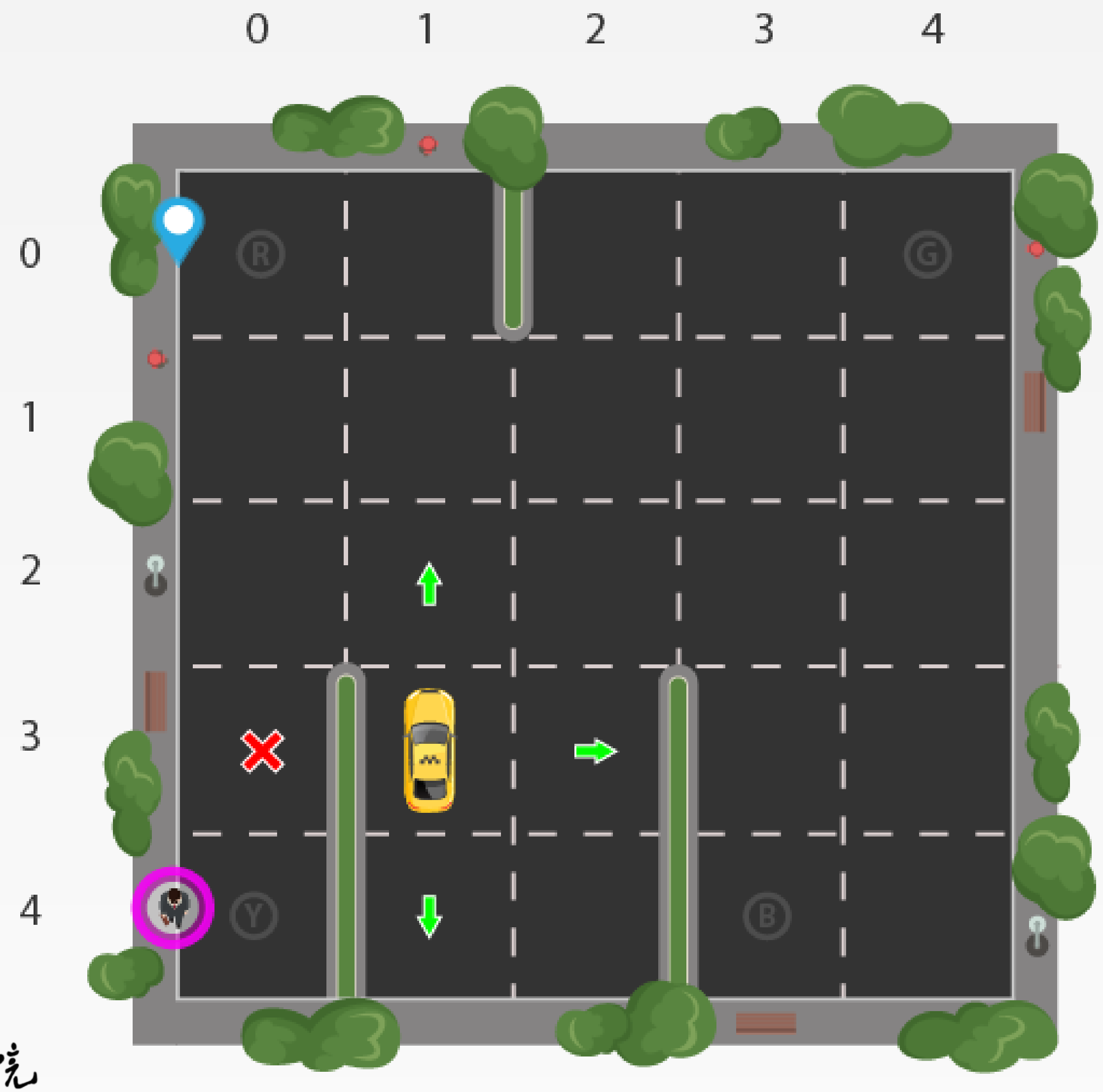
Reinforcement Learning



Open AI Gym

- https://gym.openai.com/
- “Open source interface to reinforcement learning tasks. The gym library provides an easy-to-use suite of reinforcement learning tasks.”
- “We provide the environment; you provide the algorithm.”

Taxi-v3




```
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
```

Basic Setup

```python
import gym

env = gym.make("Taxi-v3").env
env.render()

env.reset() # reset environment to
env.render()

print("Action Space {}".format(env
print("State Space {}".format(env.
```

```
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+

Action Space Discrete(6)
State Space Discrete(500)
```

Basic Setup

```python
state = env.encode(3, 1, 2, 0) #
print("State:", state)

env.s = state
env.render()


env.P[328]
```

```
{0: [(1.0, 428, -1, False)],
 1: [(1.0, 228, -1, False)],
 2: [(1.0, 348, -1, False)],
 3: [(1.0, 328, -1, False)],
 4: [(1.0, 328, -10, False)],
 5: [(1.0, 328, -10, False)]}
```

```
State: 328
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
```

Solve the problem without RL

```python
env.s = 328  # set environment to illustration's state

epochs = 0
penalties, reward = 0, 0

frames = [] # for animation

done = False

while not done:
    action = env.action_space.sample()
    state, reward, done, info = env.step(action)

    if reward == -10:
        penalties += 1

    # Put each rendered frame into dict for animation
    frames.append({
        'frame': env.render(mode='ansi'),
        'state': state,
        'action': action,
        'reward': reward
        }
    )

    epochs += 1
```

```
+---------+
|R: | : :G|
| : : : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
  (Dropoff)

Timestep: 1
State: 328
Action: 5
Reward: -10
```

Q-Learning (Training)

```
import numpy as np
q_table = np.zeros([env.observation_space.n, env.action_space.n])

import random
from IPython.display import clear_output

# Hyperparameters
alpha = 0.1
gamma = 0.6
epsilon = 0.1

# For plotting metrics
all_epochs = []
all_penalties = []
```

Q-Table

Initialized

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	327	0	0	0	0	0	0
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	499	0	0	0	0	0	0

Q-Table

Training

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603

Q-Learning (Training)

```python
for i in range(1, 100001):
    state = env.reset()

    epochs, penalties, reward, = 0, 0, 0
    done = False

    while not done:
        if random.uniform(0, 1) < epsilon:
            action = env.action_space.sample() # Explore action space
        else:
            action = np.argmax(q_table[state]) # Exploit learned values

        next_state, reward, done, info = env.step(action)

        old_value = q_table[state, action]
        next_max = np.max(q_table[next_state])

        new_value = (1 - alpha) * old_value + alpha * (reward + gamma * next_max)
        q_table[state, action] = new_value

        if reward == -10:
            penalties += 1

        state = next_state
        epochs += 1
```

Q-Learning (Evaluation)

```
total_epochs, total_penalties = 0, 0
episodes = 100

for _ in range(episodes):
    state = env.reset()
    epochs, penalties, reward = 0, 0, 0

    done = False

    while not done:
        action = np.argmax(q_table[state])
        state, reward, done, info = env.step(action)

        if reward == -10:
            penalties += 1

        epochs += 1

    total_penalties += penalties
    total_epochs += epochs
```

Comparison

Measure	Random agent's performance	Q-learning agent's performance
Average rewards per move	-3.9012092102214075	0.6962843295638126
Average number of penalties per episode	920.45	0.0
Average number of timesteps per trip	2848.14	12.38

Exam Logistics

Final Exam

- **December 2, 2021 3:10-5:10 pm**
- Location: **光华101教室**
  - Online (4 students, Teams)

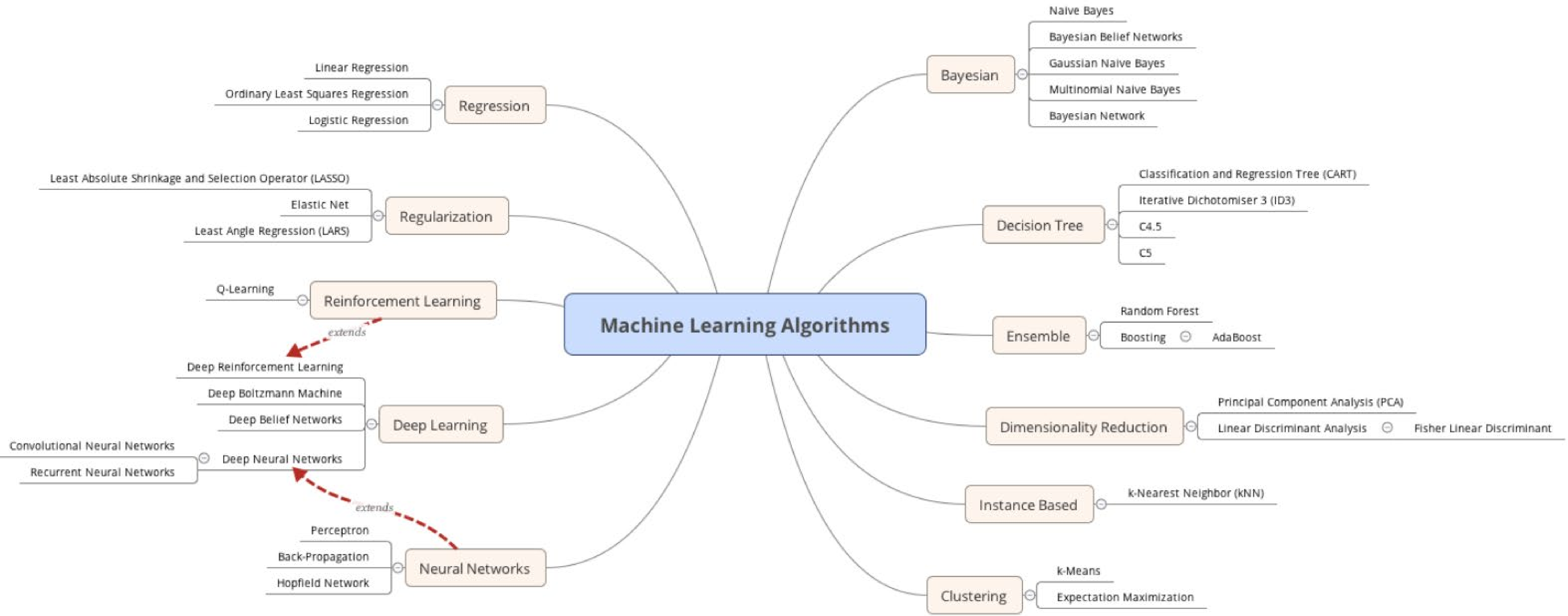
- Closed-book
- One-page (A4 size) cheating paper
- A calculator is allowed

Final Exam

- Total points: 100 + 5 bonus

- Formats of questions:
  - True/false
  - Multiple choice questions
  - Short answer questions
  - Interpreting figures

- Either Chinese or English is OK.



Machine Learning Algorithms
Regression
Linear Regression
Ordinary Least Squares Regression
Logistic Regression
Regularization
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Reinforcement Learning
Q-Learning
extends
Deep Learning
Deep Reinforcement Learning
Deep Boltzmann Machine
Deep Belief Networks
Deep Neural Networks
Convolutional Neural Networks
Recurrent Neural Networks
extends
Neural Networks
Perceptron
Back-Propagation
Hopfield Network
Bayesian
Naive Bayes
Bayesian Belief Networks
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5
Ensemble
Random Forest
Boosting
AdaBoost
Dimensionality Reduction
Principal Component Analysis (PCA)
Linear Discriminant Analysis
Fisher Linear Discriminant
Instance Based
k-Nearest Neighbor (kNN)
Clustering
k-Means
Expectation Maximization

Topics

- Important concepts:
  - Supervised vs. Unsupervised
    - Generative vs. Discriminative
  - Optimization
    - Gradient Descent
  - Model selection and evaluation
  - Overfitting
  - Regularization
  - Applications

Types of ML Systems

Criteria Whether or not they are trained with human supervision

Supervised Learning

Fraud detection
Prediction of stock markets

Unsupervised Learning

Customer segmentation
Recommendation

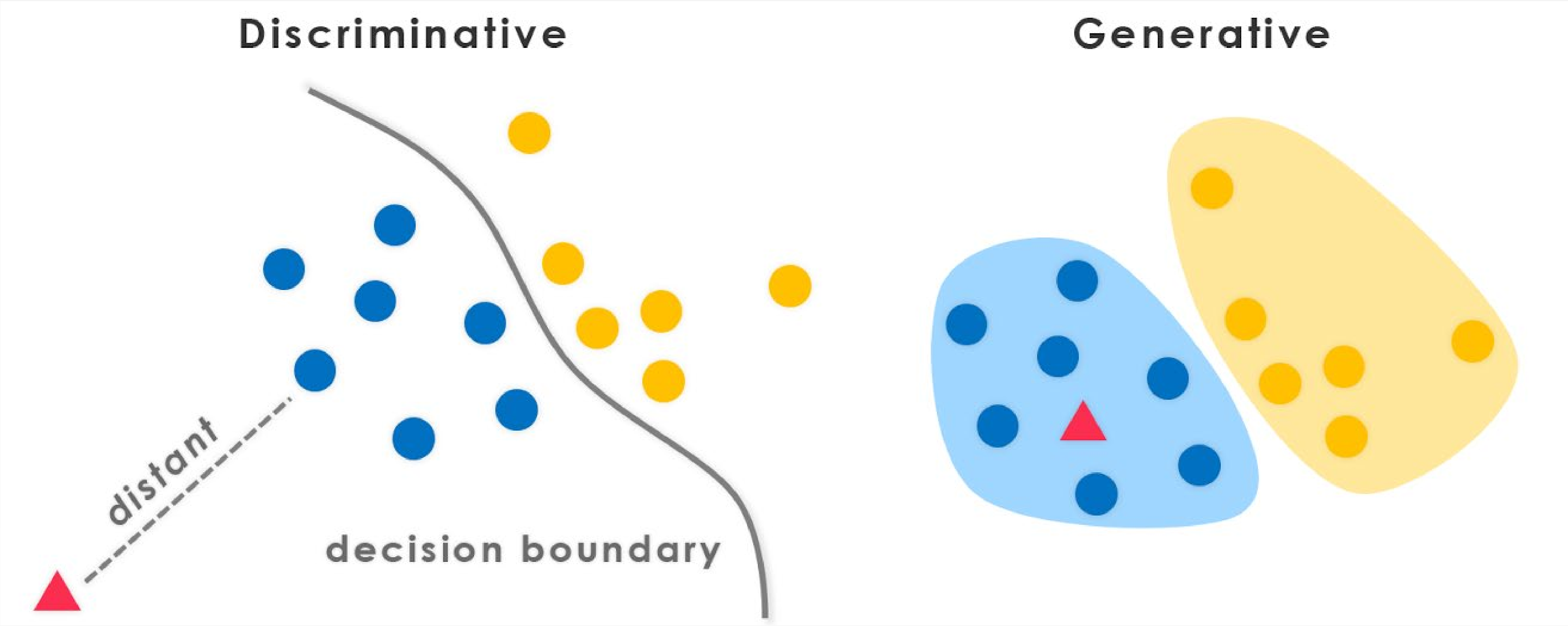
Semi-supervised Learning

Photo-hosting service
Speech analysis
Web-content classification

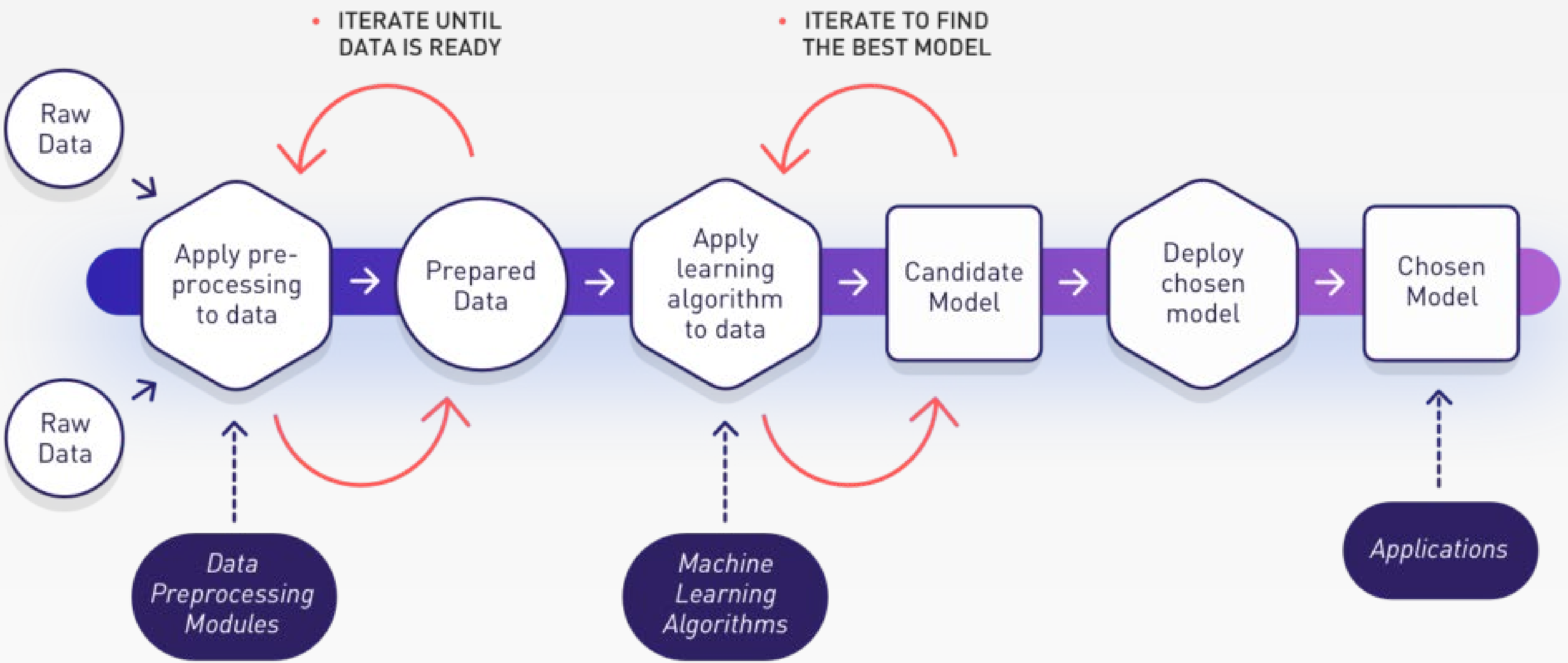
Reinforcement Learning

Robotics
Go games
Self-driving cars

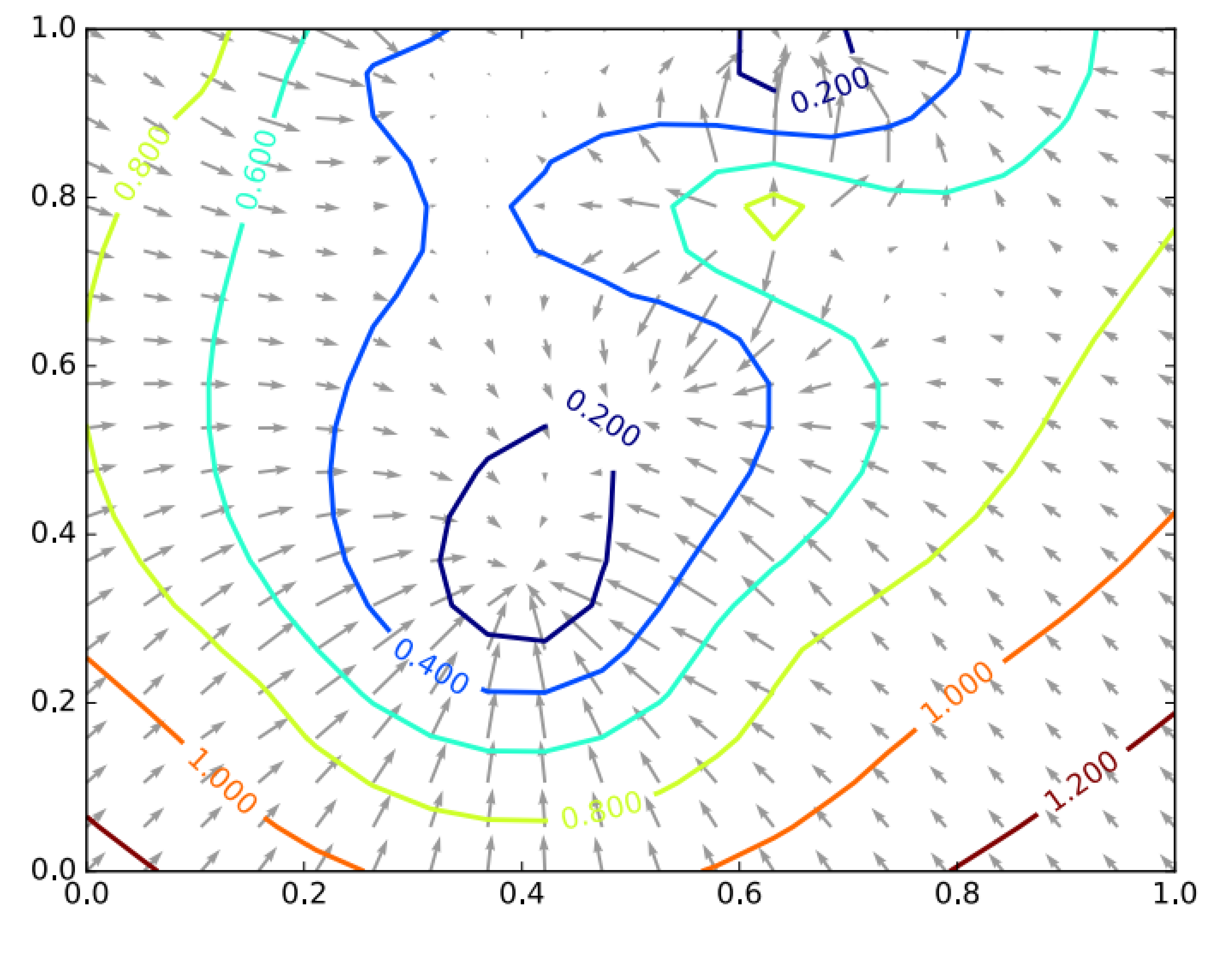
Generative vs. Discriminative



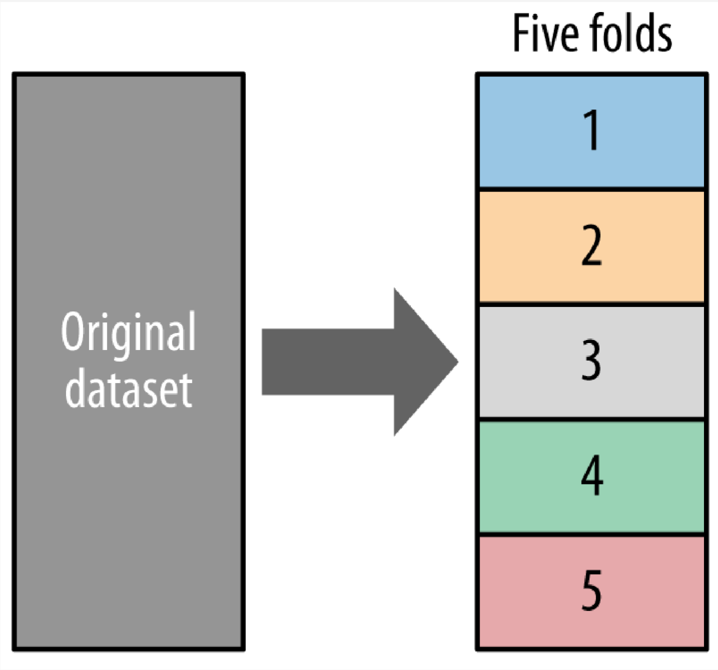
Machine Learning Workflow

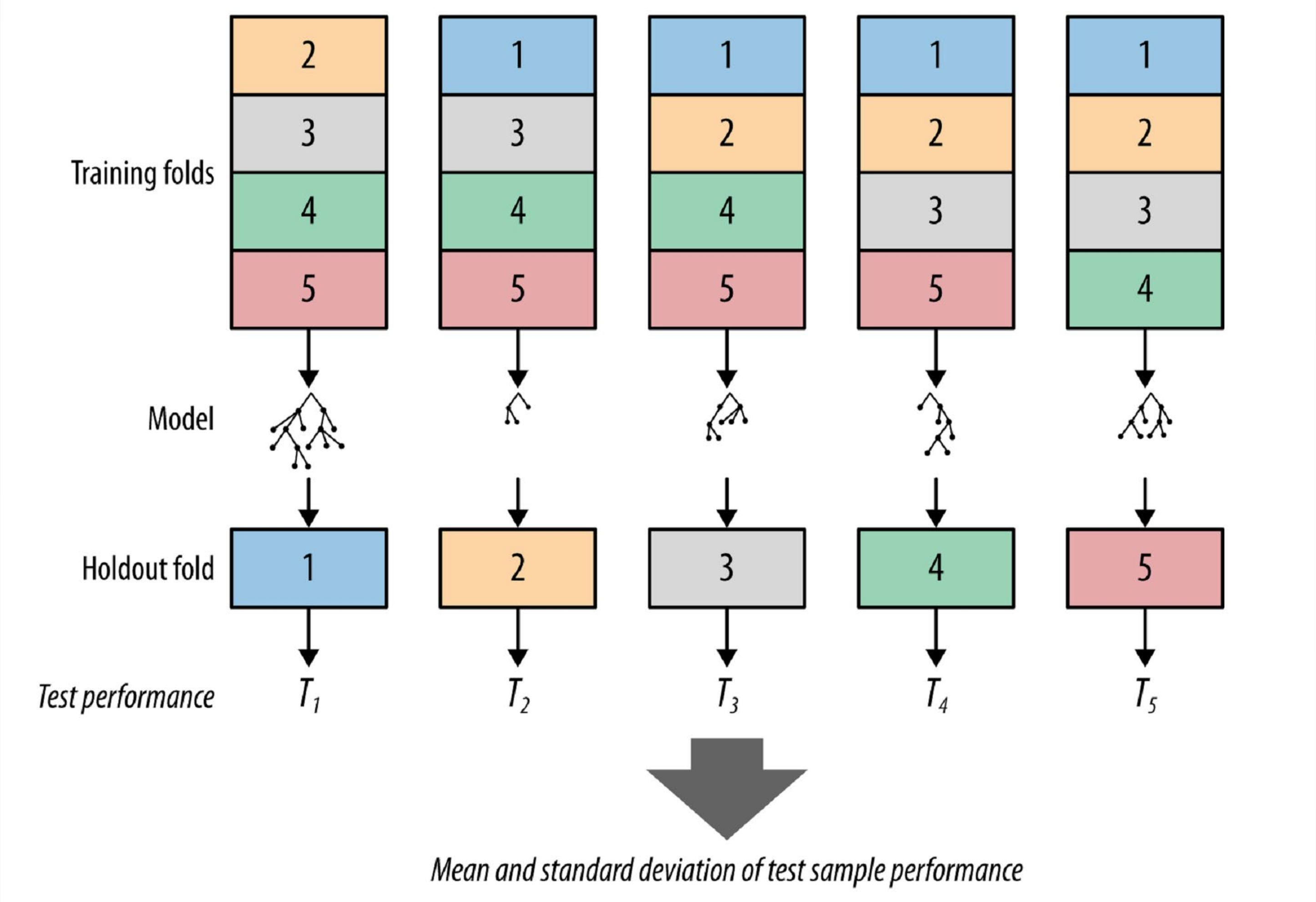


Gradient Descent

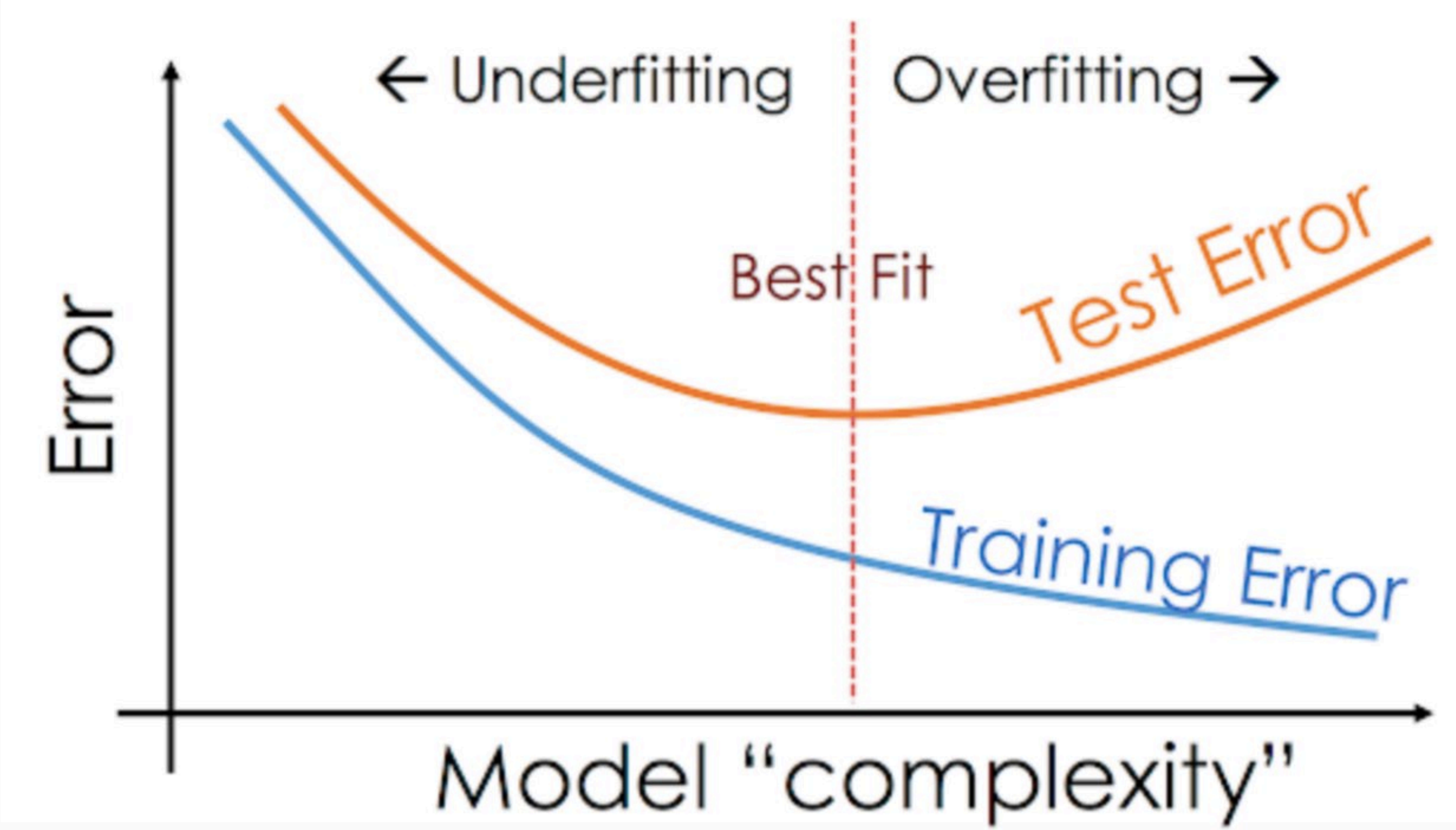


N-fold Cross Validation





Overfitting



Regularization

- Goal: optimize some combination of fit and simplicity
  - Penalize the magnitude of coefficients of features
  - Minimize the error between predicted and actual examples

- Ridge Regression:
  - L2-norm: adds penalty equivalent to square of the magnitude of coefficients
- Lasso Regression:
  - L1-norm: adds penalty equivalent to absolute value of the magnitude of coefficients

Topics

- Models:
  - KNN
  - Regressions
    - Linear; Logistic; Polynomial; Ridge; LASSO
  - Decision Trees
  - SVM

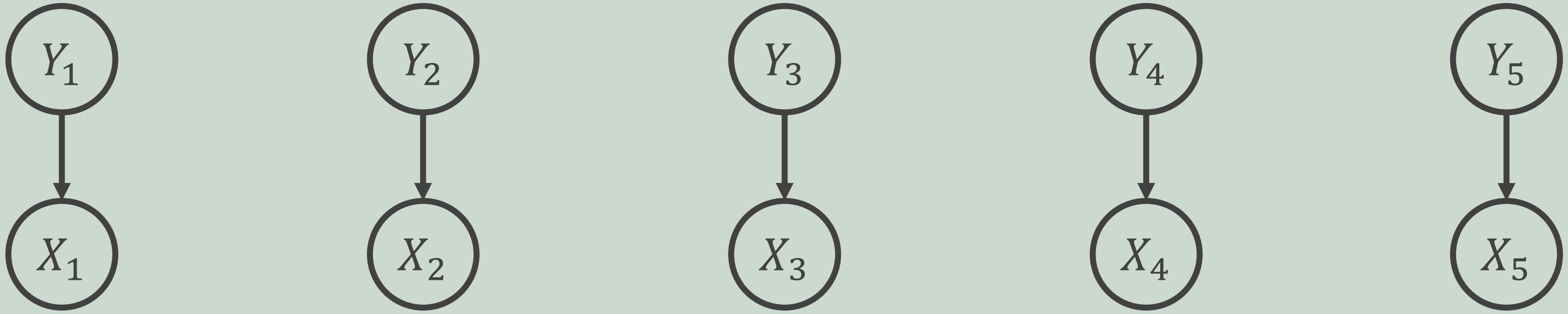
  - Naïve Bayes
  - Hidden Markov Model

  - Ensemble Models

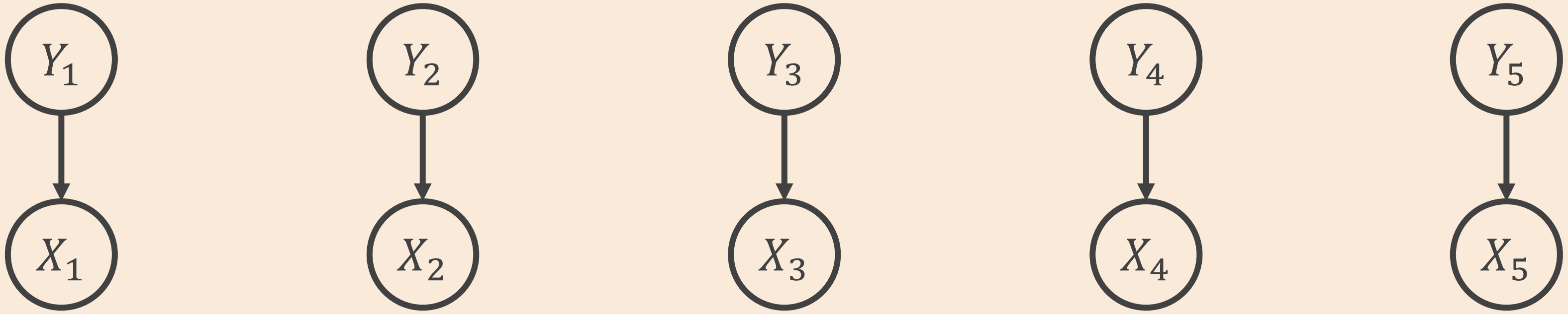
DT Comparison

Features	*ID3*	*C4.5*	*CART*
Formula	Information Gain	Gain Ratio	Gini
Pruning	No	Yes	Yes
Type of data	Categorical	Categorical / Continuous	Categorical / Continuous
Missing Values	Can't	Can	Can
Prediction	Classification	Classification	Classification / Regression

HMM

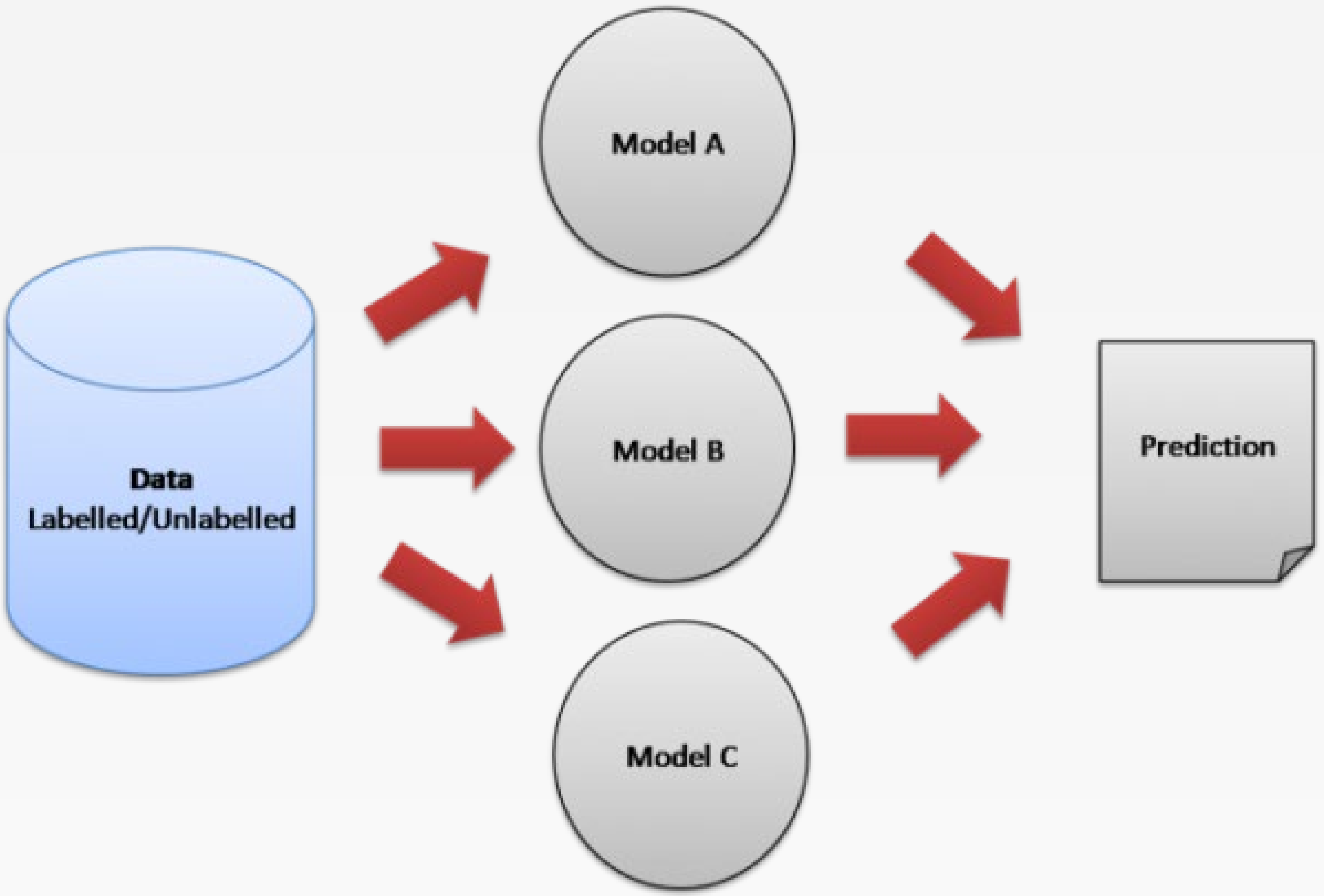


Naïve Bayes: $P(\boldsymbol{X},\boldsymbol{Y}) = \prod_{t=1}^{T} P(X_t|Y_t)p(Y_t)$



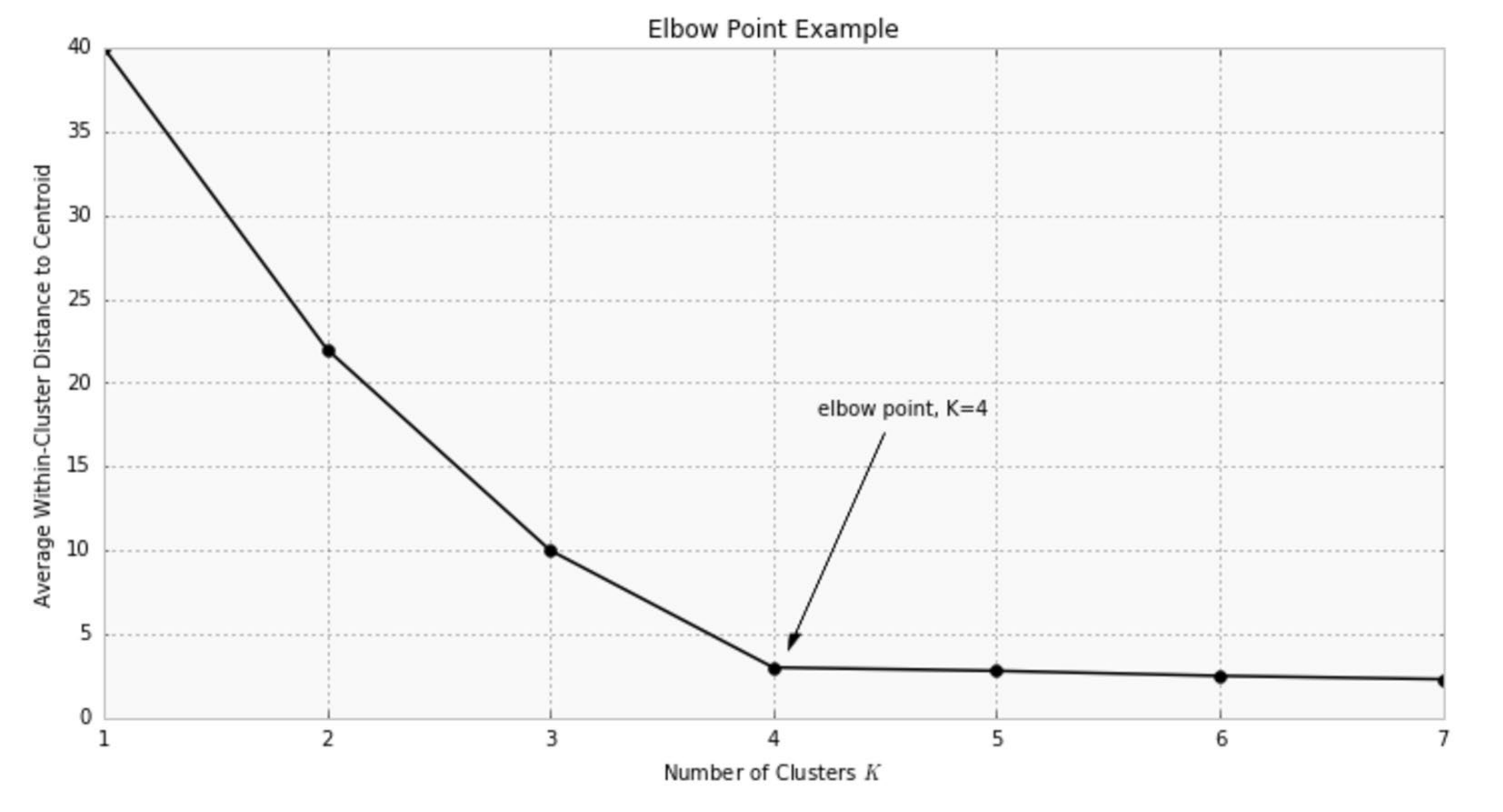
HMM: $P(\boldsymbol{X},\boldsymbol{Y}|Y_0) = \prod_{t=1}^{T} P(X_t|Y_t)p(Y_t|Y_{t-1})$

Wisdom of the Crowd

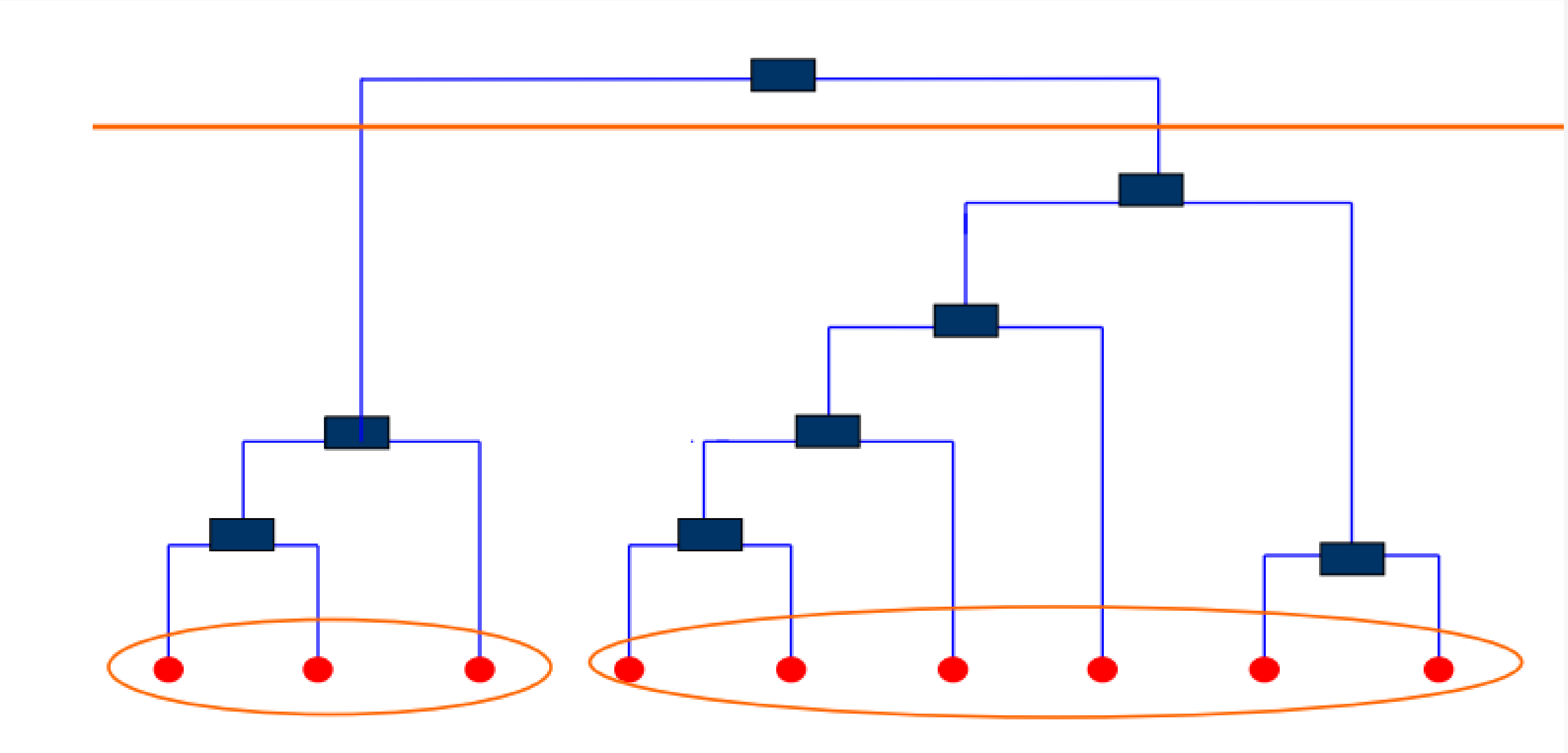


Topics

- Models:
  - Clustering
  - PCA

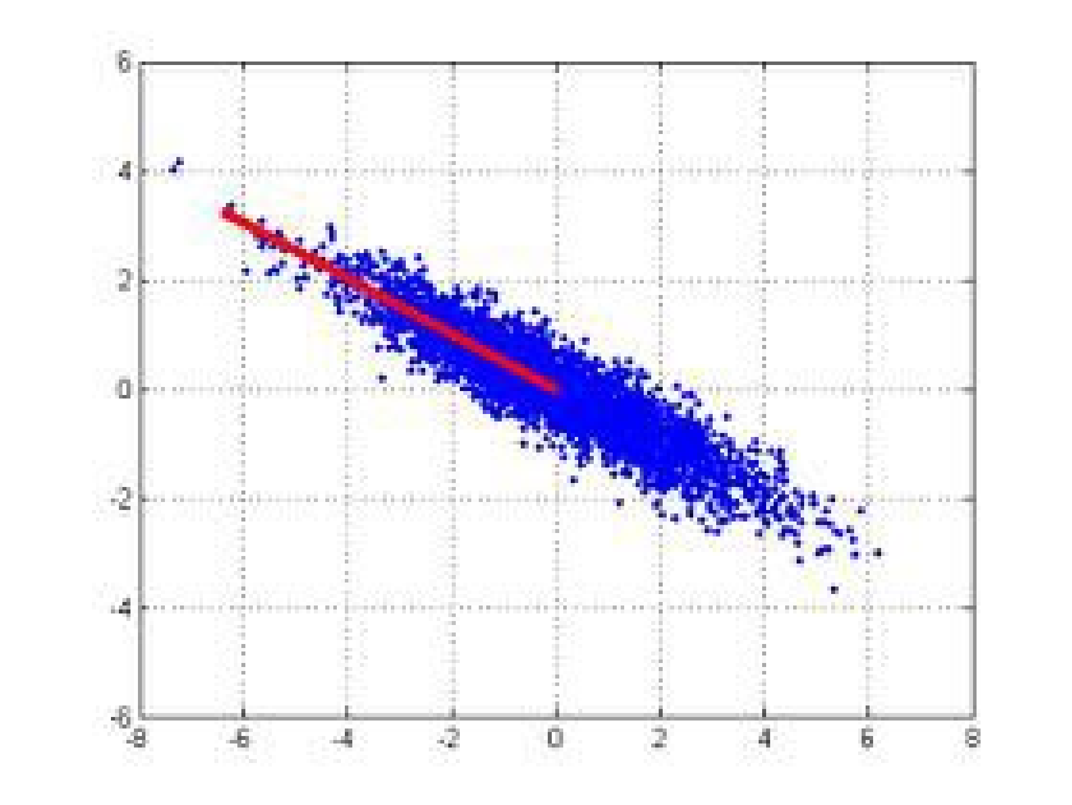

Elbow Point Example
Average Within-Cluster Distance to Centroid
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
7
Number of Clusters K
elbow point, K=4

Dendrogram



PCA

First find the direction of maximum variance, labeled “Component 1”



Along this direction:

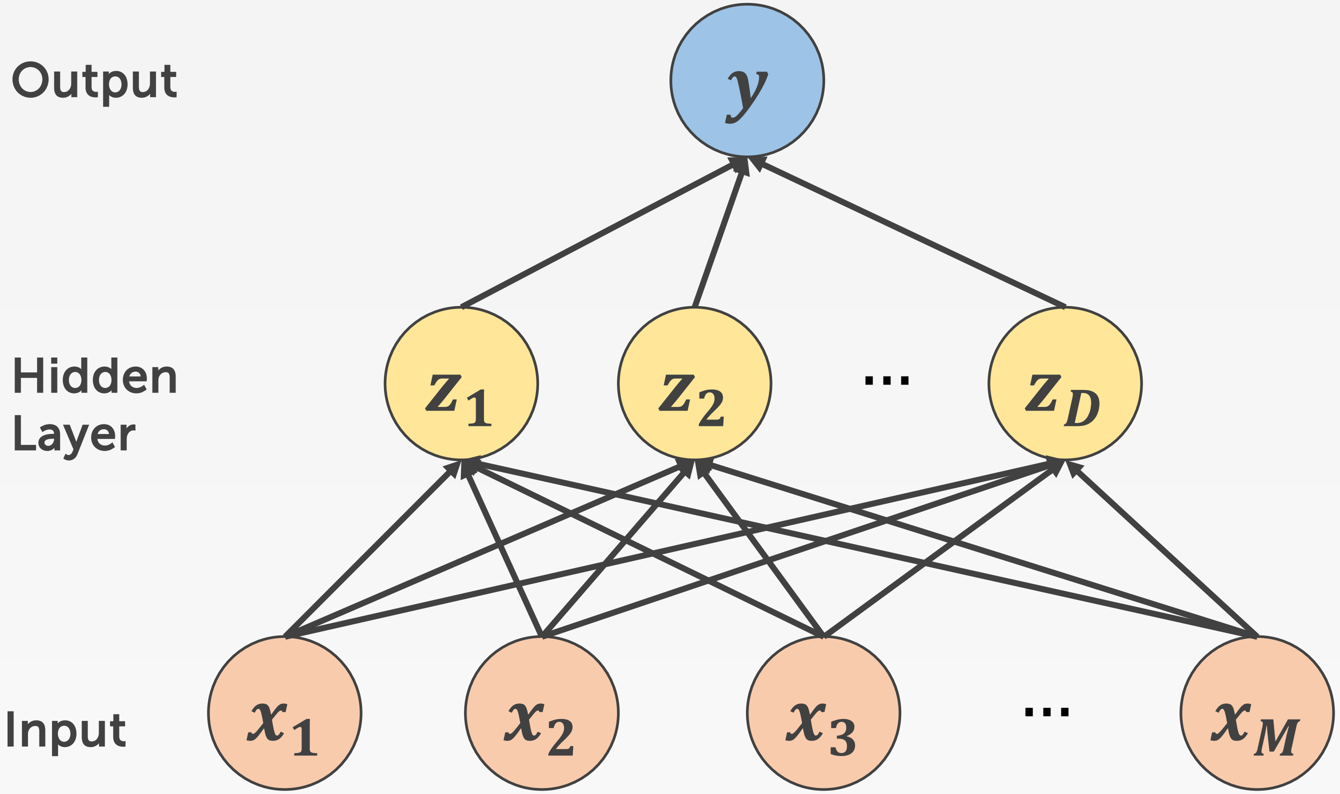
- Features are most correlated with each other
- Contains the most of the information

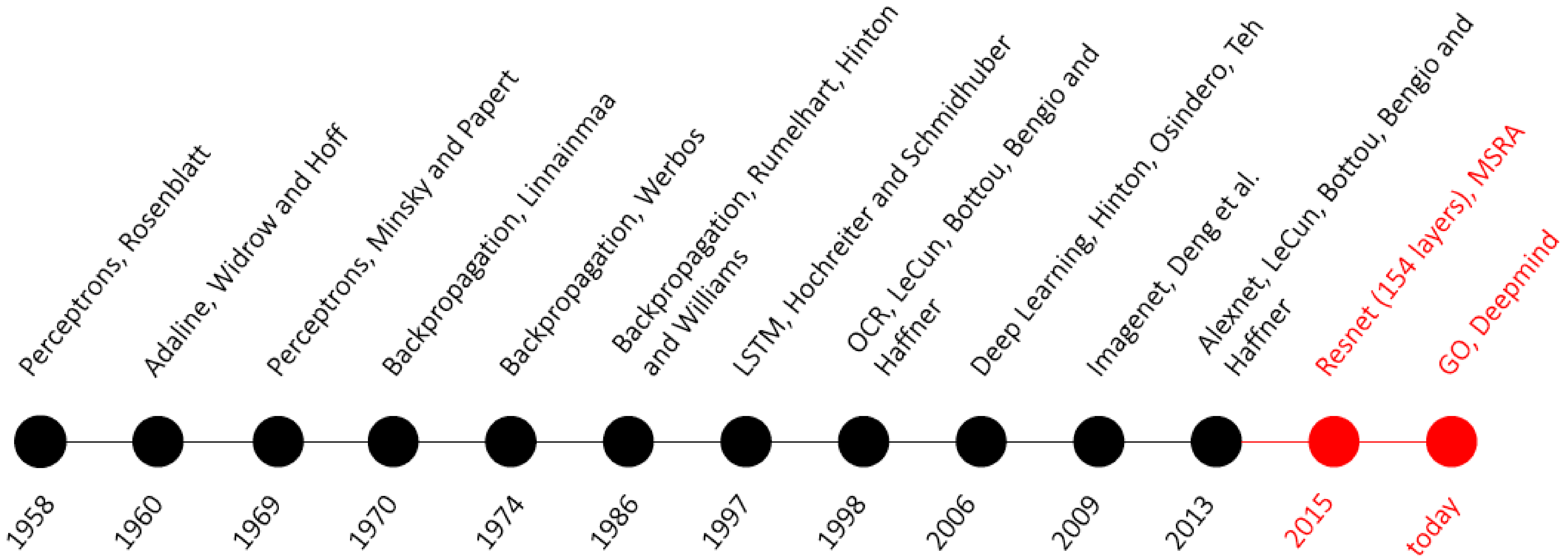
Topics

- Deep Learning Introduction
  - Neural Network
  - Backpropagation
  - Basic NN architectures

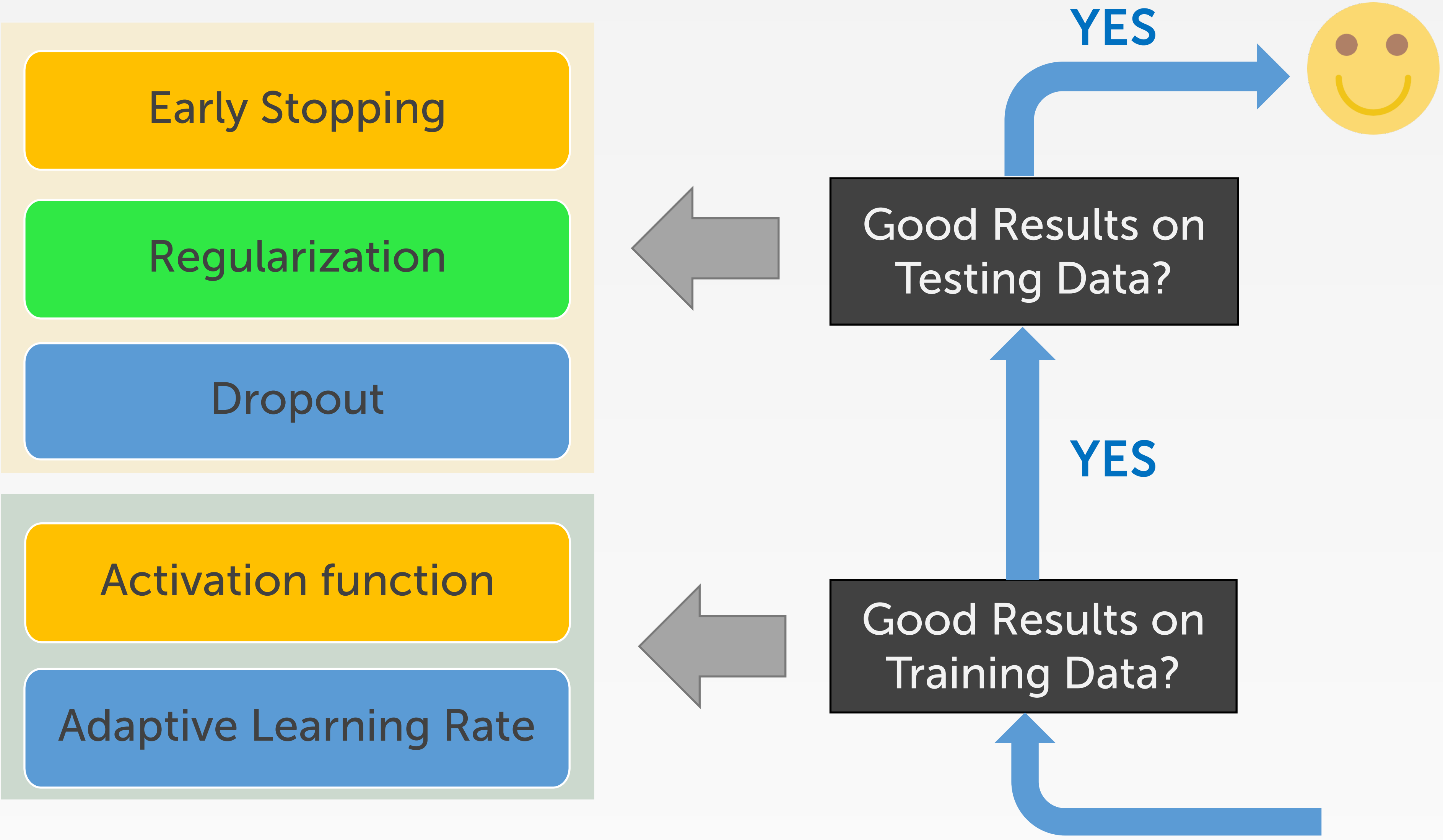
- Reinforcement Learning
  - Basic concepts
  - Value iteration vs. policy iteration
  - Q-learning

Neural Network

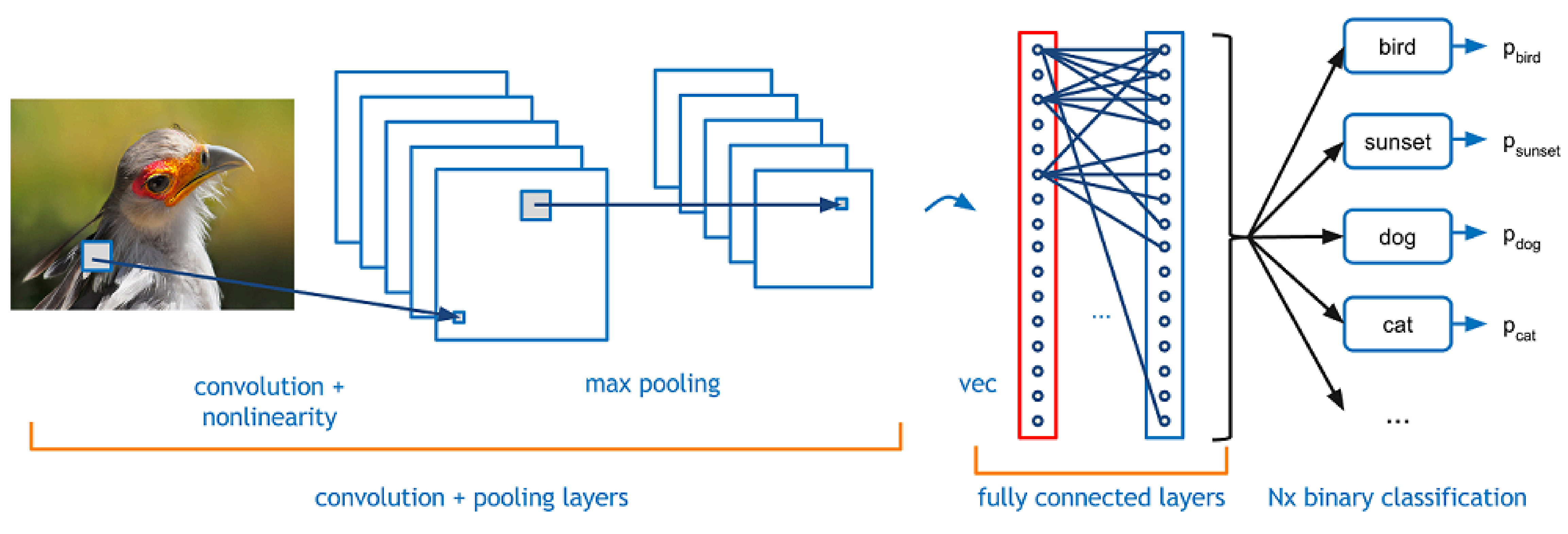



Perceptrons, Rosenblatt
1958
Adaline, Widrow and Hoff
1960
Perceptrons, Minsky and Papert
1969
Backpropagation, Linnainmaa
1970
Backpropagation, Werbos
1974
Backpropagation, Rumelhart, Hinton and Williams
1986
LSTM, Hochreiter and Schmidhuber
1997
OCR, LeCun, Bottou, Bengio and Haffner
1998
Deep Learning, Hinton, Osindero, Teh
2006
Imagenet, Deng et al.
2009
Alexnet, LeCun, Bottou, Bengio and Haffner
2013
Resnet (154 layers), MSRA
2015
GO, Deepmind
today

Recipe of Deep Learning



CNN Architecture



光华管理学院
Guanghua School of Management

Elements of RL

- Environment:
  - Physical world in which the agent operates
- State:
  - Current situation of the agent
- A policy
  - A map from state space to action space.
  - May be stochastic.
- A reward function
  - It maps each state (or, state-action pair) to a real number, called reward.
- A value function
  - Value of a state (or, state-action pair) is the total expected reward, starting from that state (or, state-action pair).

Exploration and Exploitation

- Exploration: find more information
- Exploitation: maximize the reward by exploiting already known information

Sample Questions

Sample Question 1

True or False

A classifier that attains 100% accuracy on the training set and 70% accuracy on test set is better than a classifier that attains 70% accuracy on the training set and 75% accuracy on test set.

F

Sample Question 2

True or False

Each attribute can be used for a node split in a decision tree only once.

F

Sample Question 3

T or F

- Reinforcement learning differs from supervised learning because it has a temporal structure in the learning process, whereas, in supervised learning, the prediction of a data point does not affect the data you would see in the future

T

- Value iteration is better at balancing exploration and exploitation compared with policy iteration

F
(nothing to do with exploration)

Sample Question 4

Which technique(s) would be useful for the following business problem?
“Predict whether the loan applicants are likely to default”

A. Linear Regression
B. Decision Tree
C. Unsupervised learning models
D. Logistic Regression

Sample Question 5

Which of the following statement is true for k-NN classifiers?

A. k-NN requires an explicit training step

B. The classification accuracy is better with larger values of k → no such direct relationship

C. k-NN is efficient in handling large-scale training data

D. A small value of k might lead to the overfitting issue

Sample Question 6

In the linearly separable case, which of the following may happen to the decision boundary of a SVM classifier if one of the training samples is removed.

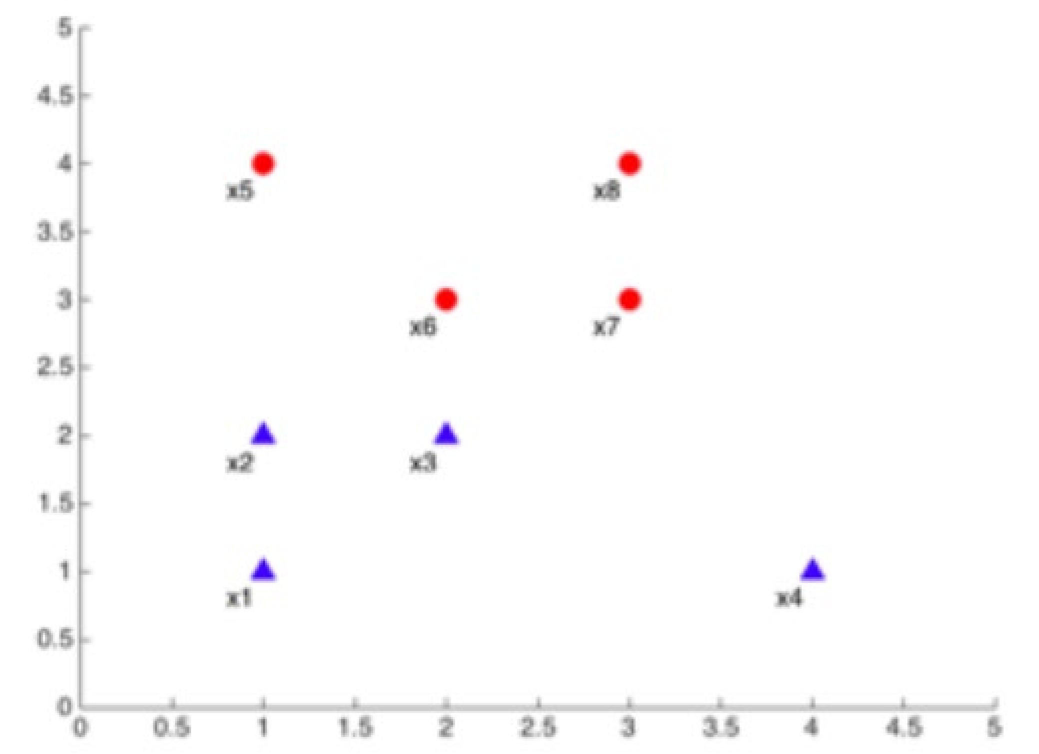
→ if support vector

A. Shifts toward the point removed

B. Shifts away from the point removed

C. Does not change

→ if no supp vector



Sample Question 7

We are given n data points, $x_1,\ldots, x_n$ and asked to cluster them using K-means. If you choose the value for k to optimize the objective function, how many clusters will be used?

A. 1

B. 2

C. n

D. $\log(n)$

Sample Question 8

Consider a regression model where we want to predict variable y from a single feature x. Consider two possible models to be estimated:

$y = \omega_0 + \omega_1 x$ (B.1)

$y = \omega_0 + \omega_1 x + \omega_2 x^2$ (B.2)

1. Which model is more likely to fit the training data better? Explain your reasoning.

B.2

2. Which model is more likely to fit the test data better? Explain your reasoning.

↳ Cannot be said

if few data, B.2 might overfit

much data, might ~~be~~ better

$w_2 = 0$

then same

hence //

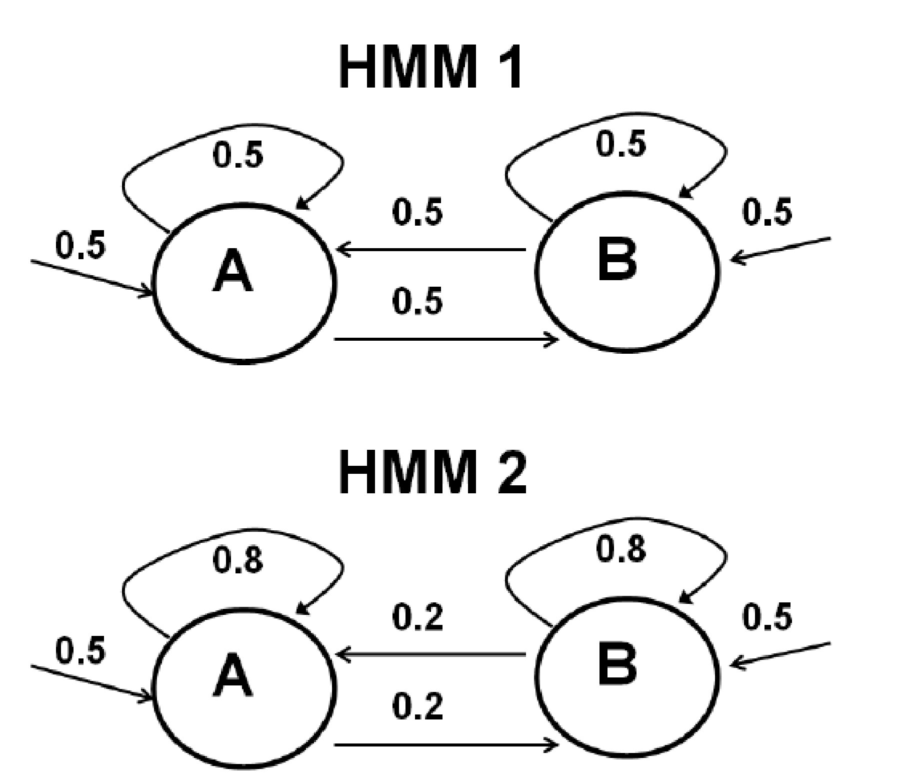
Sample Question 9

The figure above presents two HMMs. States are represented by circles and transitions by edges.

In both, emissions are deterministic and listed inside the states.

Transition probabilities and starting probabilities are listed next to the relevant edges.

For example, in HMM 1 we have a probability of 0.5 to start with the state that emits A and a probability of 0.5 to transition to the state that emits B if we are now in the state that emits A.



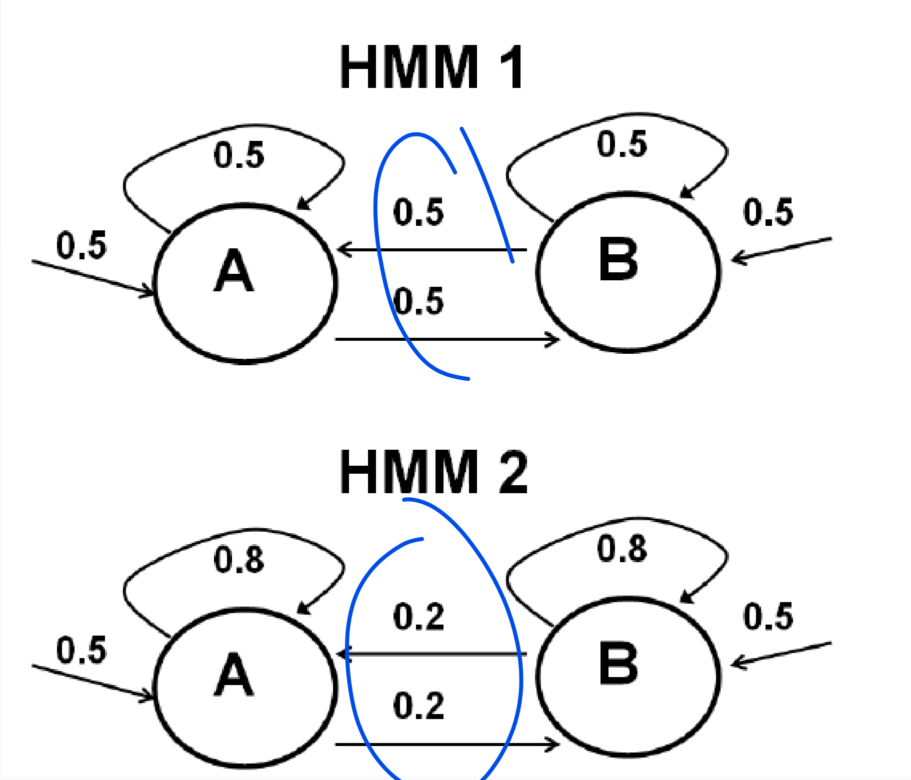
Guanghua School of Management

Sample Question 9 (cont.)

Let P_1 be: $P_1 = P(O_{100} = A, O_{101} = B, O_{102} = A, O_{103} = B)$for HMM1 and let P_2 be: $P_2 = P(O_{100} = A, O_{101} = B, O_{102} = A, O_{103} = B)$for HMM2. Choose the correct answer from the choices below and briefly explain.

$0.5^4 > 0.5 \cdot 0.2^2$

1. $P_1 > P_2$
2. $P_1 < P_2$
3. $P_1 = P_2$
4. Impossible to tell

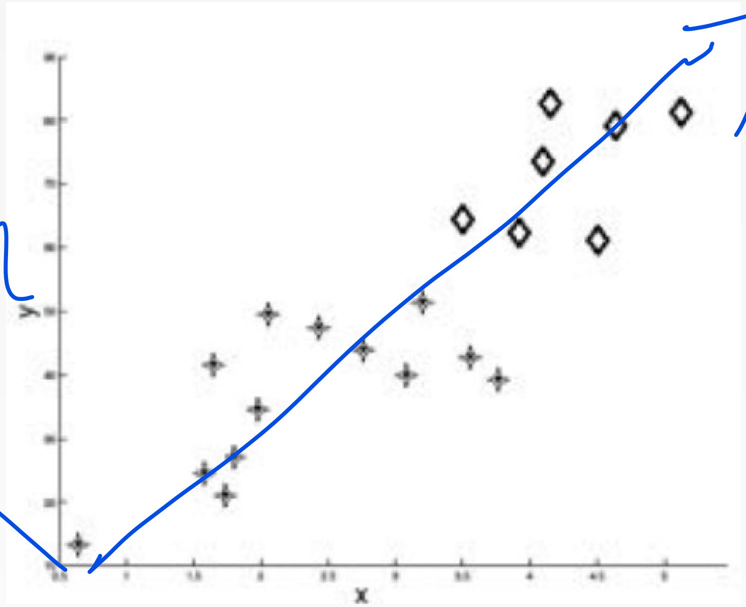


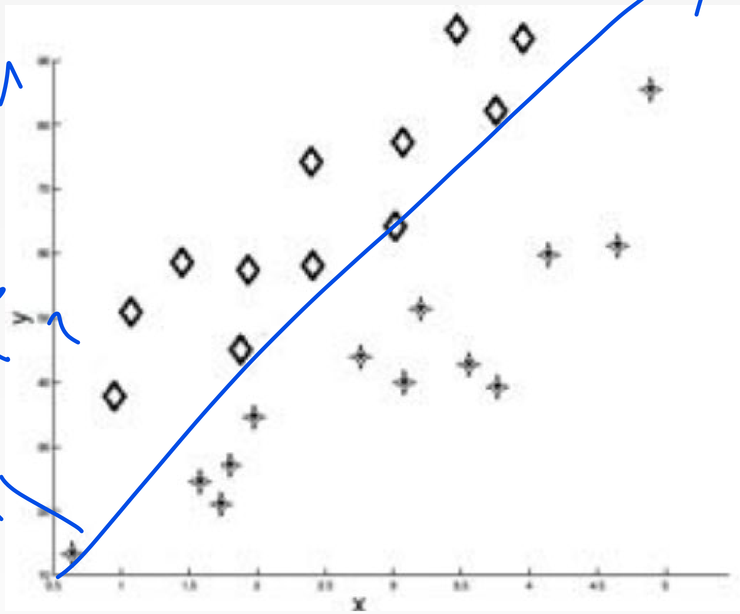
Sample Question 10

In the following plots, a train set of data points X belonging to two classes on R^2 are given, where the original features are the coordinates (x, y). For each, answer the following questions:

1. Draw all the principal components.

2. Can we correctly classify this dataset by using a threshold function after projecting onto one of the principal components?



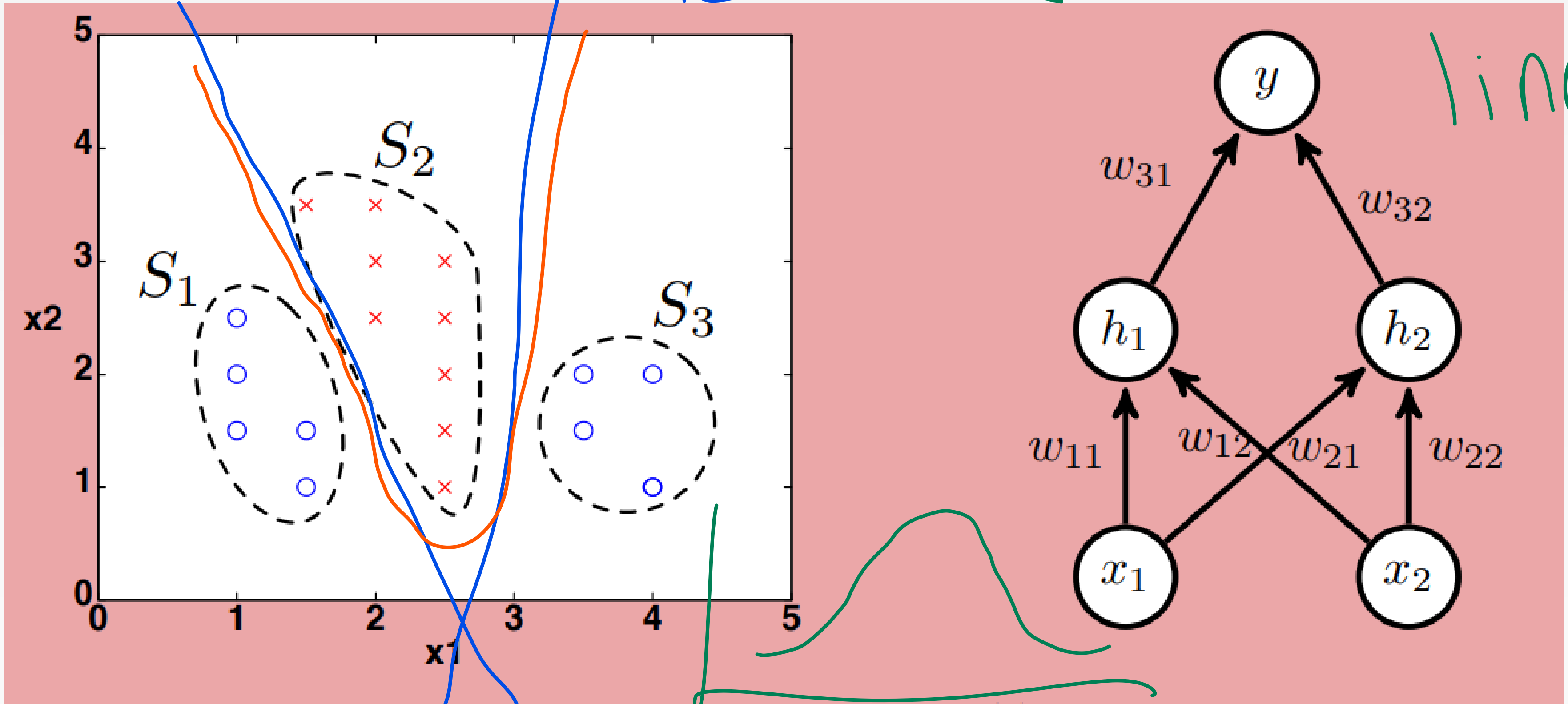


(PCAs are vectors so there is a direction!

Sample Question 11

Can the neural network in Figure (b) correctly classify the dataset given in Figure (a)?

h1 h2 Yes due to non-linearities



(a) The dataset with groups S_1, S_2, and S_3.

(b) The neural network architecture

Sample Question 12

You are given a data set of 10,000 students with their sex, height, and hair color. You are trying to build a classifier to predict the sex of a student, so you randomly split the data into a training set and a test set.

- Sex ∈ {male, female}
- Height ∈ $[0,300]$ centimeters
- Hair ∈ {brown, black, blond, red, green}
- 3240 men in the data set
- 6760 women in the data set

Under the assumption necessary for NB, answer the following questions

F → Gaussian NB

1. **T or F**: As height is a continuous valued variable, Naive Bayes is not appropriate since it cannot handle continuous valued variables
2. **T or F**: P (height |sex , hair) = P (height |sex). T

Good Luck!