

L-8

Reinforcement Learning III and Project Clinic

MAIB7002
Fundamentals of AI Models and Technologies



Mock Exam Paper

■ University No.:

--	--	--	--	--	--	--	--	--	--

 Seat No.:

--	--	--	--

 ■

THE UNIVERSITY OF HONG KONG

MAIB7002
FUNDAMENTALS OF AI MODELS AND TECHNOLOGIES

MOCK EXAM PAPER

Suggested working time: two hours

This mock paper is provided for practice in applying course concepts under examination conditions. It is illustrative rather than exhaustive; the final examination may use different questions, contexts and topic emphases.

Answer **all five questions** on this question paper in the spaces provided. Write your University No. and Seat No. in the boxes at the top of each odd-numbered page.

Two A4 sheets of printed or handwritten notes, on both sides, are permitted. **Calculators are not permitted.**

For Question 1, select one answer per item and record it in the answer grid. There is no negative marking. For Questions 2-5, show working where requested and clearly identify your final answers. Exact fractions are acceptable.

Question	Examiner only						
1	<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			/ 60
2	<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			/ 10
3	<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			/ 10
4	<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			/ 10
5	<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			/ 10
Total	<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			/ 100

Reinforcement Learning III

6:30 pm – 7:30 pm

- Review: Value Iteration
- Learn from Experience: Q-Learning
- Exploration and Exploitation

Project Clinic

7:30 pm – 9:30 pm · 5 Minutes per Group

Neel

17 Groups

A, B, G

H, I, L

M, N, R

S, U, V

W, X

Beta, Epsilon, Eta

Dirk / Jack

16 Groups

C, D, E

F, J, K

O, P, Q

T, Y, Z

Alpha, Delta

Gamma, Zeta

L7 Review

- MDPs: States, Actions, Rewards and Policies
- Discounted Return: Immediate and Later Rewards
- Value Iteration: Plan with a Known Model
- Passive RL: Learn the Value of a Fixed Policy

What Is Value Iteration For?

			+1 Exit
			-1 Exit

Move reward 0 · $\gamma = 0.9$

Plan Before Acting

Given: a Known Model

Transitions T and rewards R

What Is Value Iteration For?

0.73	0.81	0.90	1
0.66		0.81	-1
0.59	0.66	0.73	0.66

Move reward 0 · $\gamma = 0.9$

How Good Can Each State Be?

Given: a Known Model

Transitions T and rewards R

Compute: Optimal Values V^*

Best expected discounted return

What Is Value Iteration For?

→	→	→	Exit
↑		↑	Exit
↑	→	↑	←

Move reward $0 \cdot \gamma = 0.9$

Which Action Should We Take?

Given: a Known Model

Transitions T and rewards R

Compute: Optimal Values V^*

Best expected discounted return

Choose: an Optimal Policy π^*

An optimal action in each state

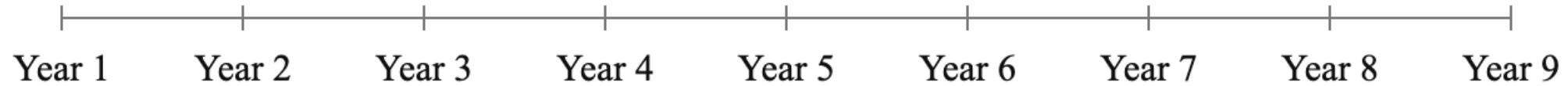
Value Iteration in Gridworld

Change the move reward, discount and transition probabilities

Compare the optimal values V^* and policy π^*

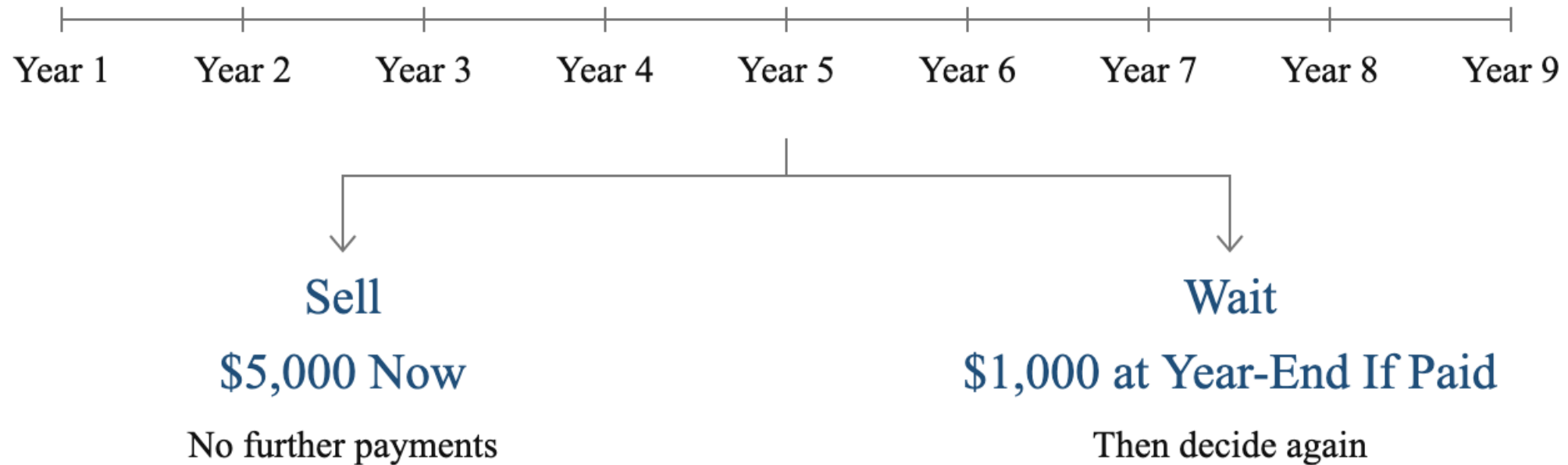
Sell Now or Keep Collecting?

You Own a Contract Promising \$1,000 Each Year for Nine Years



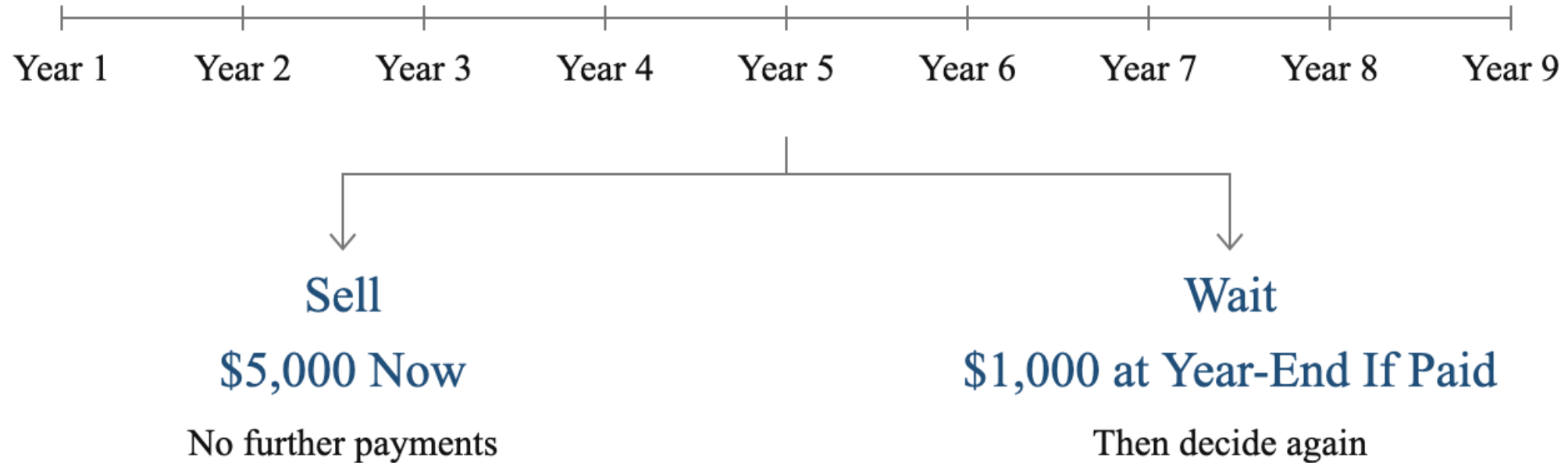
Sell Now or Keep Collecting?

You Own a Contract Promising \$1,000 Each Year for Nine Years



Sell Now or Keep Collecting?

You Own a Contract Promising \$1,000 Each Year for Nine Years

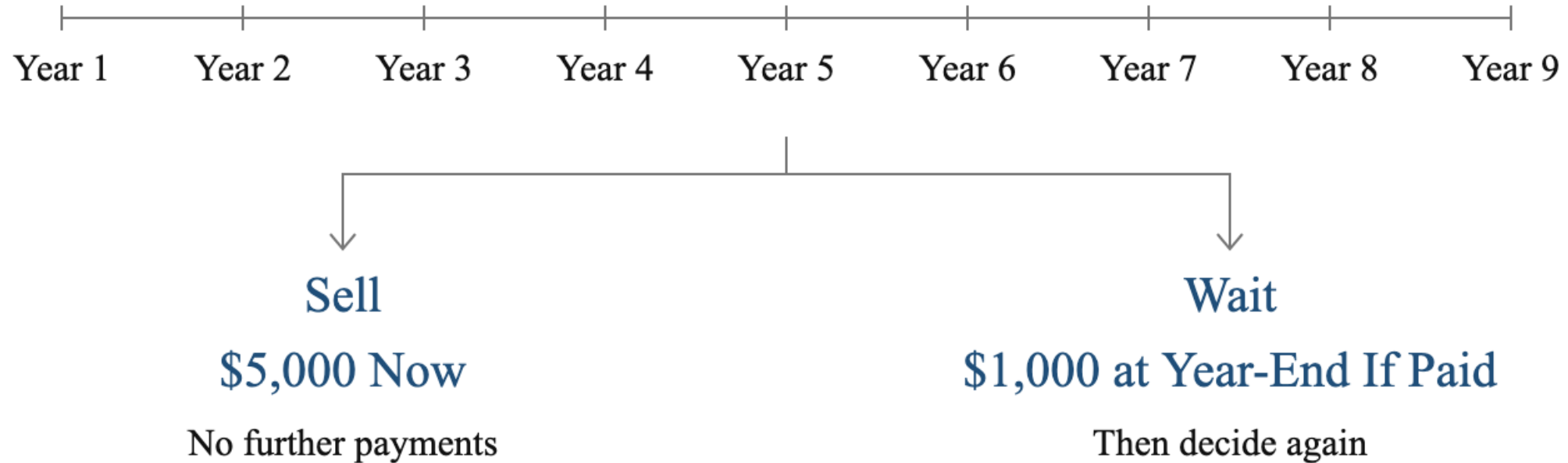


10% Chance of Default Each Year

If default occurs: no payment and all remaining payments are lost

Sell Now or Keep Collecting?

You Own a Contract Promising \$1,000 Each Year for Nine Years



10% Chance of Default Each Year

If default occurs: no payment and all remaining payments are lost

Would You Sell Now, or Wait?

Sell Now or Keep Collecting?

Each year: sell the remaining payments, or wait

Waiting may lead to a payment or permanent default

Use the known model to find optimal values and a policy

The Bellman Optimality Equation

$$V^*(s) = \max_a Q^*(s, a)$$

Best State Value = Highest Action Value

The Bellman Optimality Equation

$$V^*(s) = \max_a Q^*(s, a)$$

Best State Value = Highest Action Value

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Take Action a, Then Act Optimally

The Bellman Optimality Equation

$$V^*(s) = \max_a Q^*(s, a)$$

Best State Value = Highest Action Value

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Take Action a, Then Act Optimally

$$\pi^*(s) = \arg \max_a Q^*(s, a)$$

Choose an Action That Achieves the Maximum

State Values and Action Values

Inspect the action values Q^* in a state

V^* is the highest action value; π^* selects a maximising action

State Values and Action Values

Inspect the action values Q^* in a state

V^* is the highest action value; π^* selects a maximising action

Value Iteration

The Bellman Optimality Equation

$$V^*(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Value Iteration

The Bellman Optimality Equation

$$V^*(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Use the Previous State Values

$$V_k(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_{k-1}(s')]$$

Start with $V_0 = 0$ · Update every state

Q-Value Iteration

The Bellman Optimality Equation

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q^*(s', a')]$$

Q-Value Iteration

The Bellman Optimality Equation

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q^*(s', a')]$$

Use the Previous Action Values

$$Q_k(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q_{k-1}(s', a')]$$

Start with $Q_0 = 0 \cdot V_k(s) = \max_a Q_k(s, a)$

Q-Value Iteration in Gridworld

Start with $Q_0 = 0$

Use + / - to inspect successive synchronous Q-value iteration sweeps

When T and R Are Not Supplied

Planning with a Known MDP

With a model, calculate the update

$$Q_k(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q_{k-1}(s', a')]$$

When T and R Are Not Supplied

Planning with a Known MDP

We cannot directly calculate this update

$$Q_k(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q_{k-1}(s', a')]$$

T and R: Not supplied

When T and R Are Not Supplied

Planning with a Known MDP

We cannot directly calculate this update

$$Q_k(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q_{k-1}(s', a')]$$

T and R: Not supplied

Reinforcement Learning

Take action a $\longrightarrow$ observe r, s'

Learn values from experience

Q-Learning

$$Q_{\text{updated}} = (1 - \alpha)Q_{\text{old}} + \alpha Q_{\text{step}}$$

Combine the old estimate with this step's estimate

Q-Learning

$$Q_{\text{updated}} = (1 - \alpha)Q_{\text{old}} + \alpha Q_{\text{step}}$$

Combine the old estimate with this step's estimate

$$Q_{\text{step}} = \text{Reward} + \gamma \times \text{Best next-action estimate}$$

Q-Learning

$$Q_{\text{updated}} = (1 - \alpha)Q_{\text{old}} + \alpha Q_{\text{step}}$$

Combine the old estimate with this step's estimate

$$Q_{\text{step}} = \text{Reward} + \gamma \times \text{Best next-action estimate}$$

$$0 \leq \alpha \leq 1$$

0: Keep the old estimate · 0.5: Move halfway · 1: Use this step's estimate

With noisy outcomes, decreasing α can help estimates settle

Only the experienced state–action pair is updated

Q-Learning with Random Actions

A random agent explores the same Gridworld

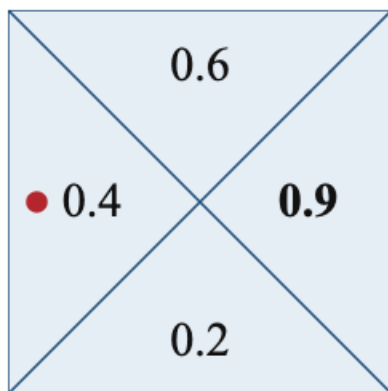
Watch each old estimate, this step's estimate and updated estimate

Epsilon-Greedy Exploration

Can we earn more reward while we learn?

With Probability ϵ

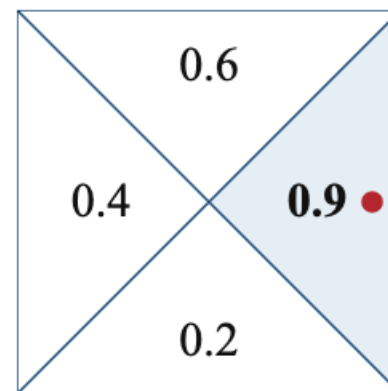
Explore



Each action is equally likely

With Probability $1 - \epsilon$

Exploit



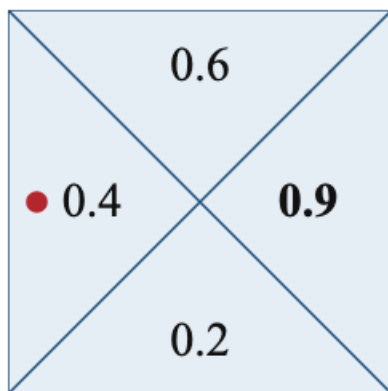
Choose the highest Q estimate

Epsilon-Greedy Exploration

Can we earn more reward while we learn?

With Probability ϵ

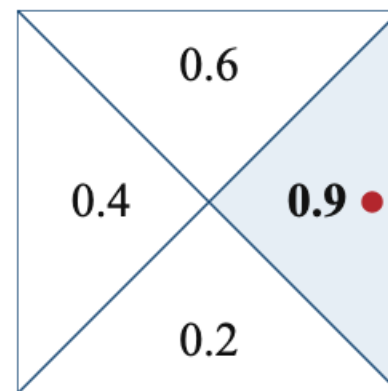
Explore



Each action is equally likely

With Probability $1 - \epsilon$

Exploit



Choose the highest Q estimate

α changes how much we learn from an outcome

ϵ changes which actions we try

Same Q-learning update; different action selection

Q-Learning with ϵ -Greedy Actions

Q-Learning with ϵ -Greedy Actions

Keep the learned values; change how actions are chosen

Quiz

Scan the QR Code to Open Ed

Answer the Questions

Share Any Questions You Have

Fundamentals of AI Models and Technologies

Supervised Learning

Predict outcomes

Learn from labelled examples

Hotel booking outcomes

Unsupervised Learning

Discover structure

Find patterns without target labels

Customer groups

Reinforcement Learning

Improve sequential
decisions

Learn from actions and rewards

Notification timing

Fundamentals of AI Models and Technologies

Supervised Learning

Predict outcomes

Learn from labelled examples

Hotel booking outcomes

Unsupervised Learning

Discover structure

Find patterns without target labels

Customer groups

Reinforcement Learning

Improve sequential
decisions

Learn from actions and rewards

Notification timing

Business Question → Data → Python Model → Evaluation

Explain the value, evidence and limitations