Introduction to Machine Learning

- Prof. Balaraman Ravindran | IIT Madras

Problem Solving Session (Week-12)

Shreya Bansal

PMRF PhD Scholar
IIT Ropar

Week-12 Contents

---

1. Introduction to Theory in Machine Learning
2. VC Dimension
3. Reinforcement Learning

Introduction to Theory in Machine Learning

- In computer science, "theory" often refers to:
  - Problem hardness (how difficult a problem is to solve)
  - Space/time complexity
  - Approximability (how close solutions are to optimal)
- Goal: Apply similar theoretical frameworks to machine learning.
- Key questions:
  - How hard is it to approximate a solution?
  - How do we measure the hardness of ML problems?

Generalization Error

- **Generalization error ($\varepsilon(h)$):** Probability that hypothesis h makes a mistake on a new data point (x,y) sampled from distribution D:

$$\epsilon(h)=P_{(x,y)\sim D}(h(x)\neq y))$$

- **Empirical error ($\hat{\varepsilon}(h)$):** Error measured on training data:

$$\epsilon\hat{}(h)=(1/m)\ \sum_{i=1}^{m}1(h(xi)\neq yi)$$

- **Challenge:** Estimate generalization error using empirical error.

Empirical Risk Minimization (ERM)

- **Most learning algorithms aim to minimize empirical error:**

$$\hat{h} = \arg\min_{h \in H} \hat{\epsilon}(h)$$

- **Hypothesis class H:**
- **Linear classifiers: Defined by parameters θ.**
- **Neural networks: Defined by architecture + weights.**
- **Practical issue: ERM may not find the true minimizer (e.g., due to optimization challenges).**

Key Theoretical Tools

---

1. **Union Bound**: For events $A_1, \ldots, A_k$:

$$P\left(\bigcup_{i=1}^{k} A_i\right) \leq \sum_{i=1}^{k} P(A_i)$$

2. **Hoeffding's Inequality**: For iid Bernoulli random variables $Z_1, \ldots, Z_m$ with mean ϕ:

$$P(|\hat{\phi} - \phi| \geq \gamma) \leq 2e^{-2\gamma^2 m}$$

   - $\hat{\phi}$: Empirical mean.
   - γ: Error tolerance.

Uniform Convergence

Apply the union bound over all k hypotheses:

$$P\left(\exists h \in H, |\hat{\epsilon}(h) - \epsilon(h)| > \gamma\right) \leq \sum_{h \in H} P\left(|\hat{\epsilon}(h) - \epsilon(h)| > \gamma\right) \leq k \cdot 2e^{-2\gamma^2 m}.$$

Simplify:

$$P\left(\text{Any } h \text{ violates } |\hat{\epsilon}(h) - \epsilon(h)| \leq \gamma\right) \leq 2ke^{-2\gamma^2 m}.$$

Uniform Convergence

- Goal: Bound $|\epsilon(h) - \hat{\epsilon}(h)|$ for all $h \in H$.
- Apply Hoeffding + Union Bound:

$$P\left(\exists h \in H, |\epsilon(h) - \hat{\epsilon}(h)| > \gamma\right) \leq 2ke^{-2\gamma^2 m}$$

- **Implication**: With probability $\geq 1 - \delta$:

$$\epsilon(\hat{h}) \leq \min_{h \in H} \epsilon(h) + 2\sqrt{\frac{\log(2k/\delta)}{2m}}$$

Probabilistic Guarantee

Let $\delta = 2ke^{-2\gamma^2 m}$. This represents the "failure probability." Solve for γ:

$$\begin{aligned} \delta &= 2ke^{-2\gamma^2 m} \\ \implies e^{2\gamma^2 m} &= \frac{2k}{\delta} \\ \implies 2\gamma^2 m &= \log\left(\frac{2k}{\delta}\right) \\ \implies \gamma &= \sqrt{\frac{\log(2k/\delta)}{2m}}. \end{aligned}$$

Now, with probability $\geq 1 - \delta$, **for all** $h \in H$:

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq \sqrt{\frac{\log(2k/\delta)}{2m}}.$$

Bound for ERM Solution

Let:

- $h^* = \arg\min_{h \in H} \epsilon(h)$ (best true error).
- $\hat{h} = \arg\min_{h \in H} \hat{\epsilon}(h)$ (ERM solution).

From uniform convergence (Step 4), with probability $\geq 1 - \delta$:

1. For $\hat{h}$:

$$\epsilon(\hat{h}) \leq \hat{\epsilon}(\hat{h}) + \gamma.$$

2. For h^*:

$$\hat{\epsilon}(h^*) \leq \epsilon(h^*) + \gamma.$$

Since $\hat{h}$ minimizes empirical error:

$$\hat{\epsilon}(\hat{h}) \leq \hat{\epsilon}(h^*).$$

Combine the inequalities:

$$\epsilon(\hat{h}) \leq \hat{\epsilon}(\hat{h}) + \gamma \leq \hat{\epsilon}(h^*) + \gamma \leq \epsilon(h^*) + 2\gamma.$$

Substitute γ:

$$\epsilon(\hat{h}) \leq \epsilon(h^*) + 2\sqrt{\frac{\log(2k/\delta)}{2m}}.$$

Sample Complexity

- To ensure $\epsilon(\hat{h}) \leq \epsilon(h^*) + 2\gamma$ with probability $1 - \delta$:

$$m \geq \frac{\log(2k/\delta)}{2\gamma^2}$$

- **Trade-off**:
  - Small H (low k): Low sample complexity but high bias.
  - Large H (high k): High sample complexity (variance) but low bias.

Sample Complexity

Example

For $k = 100$ hypotheses, $\delta = 0.05$, and $\gamma = 0.01$:

$$m \geq \frac{\log(400)}{0.0002} \approx 29,950 \text{ samples.}$$

VC Dimension

- **VC Dimension (Vapnik-Chervonenkis Dimension) is a measure of the capacity (or complexity) of a statistical classification model, specifically in terms of its ability to shatter datasets.**

Key Concepts

- **Hypothesis Space**
- **This is the set of all possible functions (or classifiers) that a model can learn. For example, in linear classifiers, it's all possible straight lines (in 2D) that can divide the space into two classes.**

- **Shattering**
- **A hypothesis space shatters a set of points if it can correctly classify all possible labelings of those points.**
    - **For n points, there are 2^n possible ways to label them (each point can be +1 or -1).**
    - **If your model can classify all 2^n labelings correctly using some function in the hypothesis space, then it shatters those n points.**

- **VC Dimension**
- **The VC dimension of a hypothesis class is the maximum number of points that can be shattered by the hypothesis class.**

Examples -1

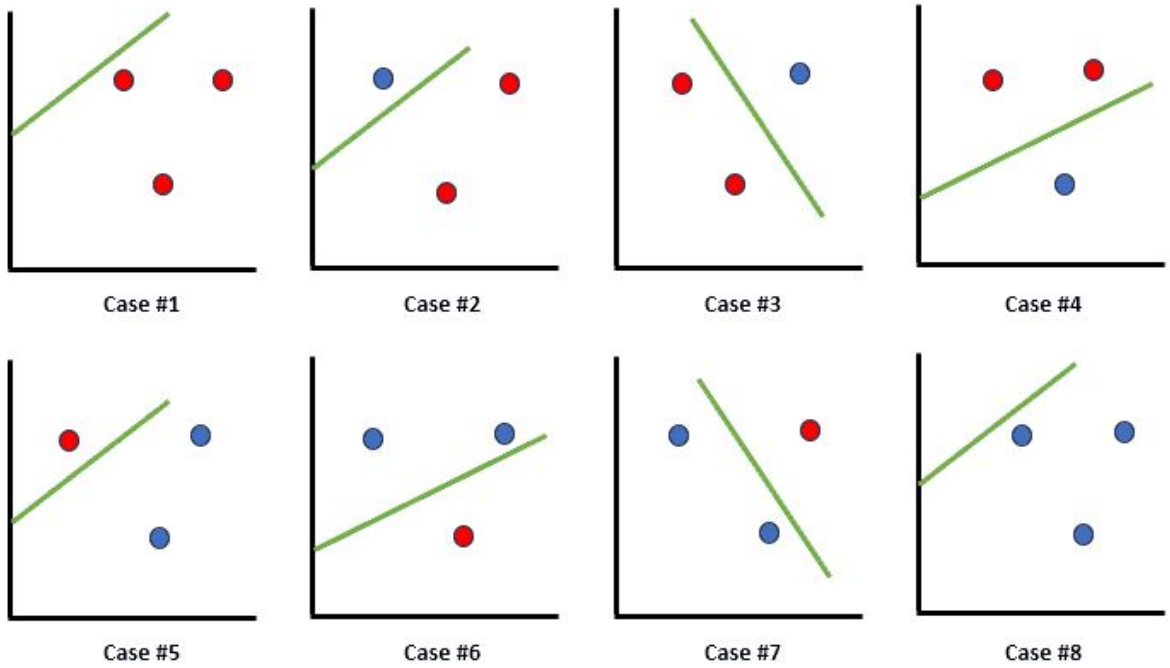
Example 1: Linear Classifiers in 2D

Imagine using straight lines to separate points into two classes (+1 or -1):

- You can **shatter any 3 non-collinear points** in 2D using a straight line. For all 8 labelings (since $2^3 = 8$), there exists a line that can separate them.
- But you **cannot always shatter 4 points**. For instance, if they are arranged in a convex quadrilateral (like a square), some labelings cannot be separated using a single straight line.

- **VC Dimension** = **3** for linear classifiers in 2D.

Examples



Examples -2

Example 2: Intervals on a Real Line

Let's consider a hypothesis class H where each hypothesis is an interval on the real line: $h(x) = 1$ if $a \leq x \leq b$, and 0 otherwise.

- You can shatter **2 points**. Any labeling of two points (on or off the interval) can be represented.
- For **3 points**, not all labelings can be achieved (e.g., 1-0-1 can't be represented by a single interval).

🔸 **VC Dimension** = **2** for this class.

Why Does VC Dimension Matter?

1. Generalization

A model with a **very high VC dimension** may fit the training data well (low training error) but fail to generalize (overfitting).

2. Sample Complexity

It tells us **how much data** we need to learn well. For a hypothesis class with VC dimension d, the number of training samples n required for good generalization grows roughly like:

$$n = O\left(\frac{d}{\epsilon}\log\frac{1}{\delta}\right)$$

where:

- ϵ is the error tolerance
- δ is the probability bound (e.g., 0.05)

What is PAC Learning?

- PAC (Probably Approximately Correct) learning is a formal framework that helps us understand when and how well a machine learning algorithm can learn a concept from data.

- The idea is:

  > “A concept is PAC-learnable if, with high probability, a learner can find a hypothesis that is approximately correct using a reasonable amount of data and computation.”

Breaking it Down: PAC Terminology

- **Let's define the terms in "Probably Approximately Correct":**
- **Probably (1 - δ): The learning algorithm should succeed with high probability (e.g., 95% of the time).**
- **Approximately (ε): The hypothesis should be close to the true concept, i.e., error $\leq \varepsilon$.**
- **Correct: Refers to how well the model generalizes on unseen data (not just training data).**

Formal Definition

A hypothesis class H is **PAC-learnable** if for every:

- target concept $c \in H$,
- distribution D over the input space,
- $0 < \epsilon < 1$ (error tolerance),
- $0 < \delta < 1$ (confidence),

there exists an algorithm that can, **with probability at least** $1 - \delta$, find a hypothesis $h \in H$ such that:

$$\Pr_{x \sim D}[h(x) \neq c(x)] \leq \epsilon$$

and the number of samples required is **polynomial in** $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, **and the size of the hypothesis class.**

Relationship with VC Dimension

VC dimension plays a central role in PAC learnability:

> **If a hypothesis class H has finite VC dimension, then:**

- It is **PAC-learnable**, and
- The number of examples needed to learn grows with the VC dimension.

📌 **The sample complexity bound:**

$$m \geq \frac{1}{\epsilon}\left(4 \cdot \mathrm{VC}(H) \cdot \log\left(\frac{2e}{\epsilon}\right) + \log\left(\frac{4}{\delta}\right)\right)$$

This tells us how many examples m are sufficient for PAC learning.

Examples of PAC Learnable Classes

Hypothesis Class	VC Dimension	PAC Learnable?
Thresholds in 1D	1	Yes
Intervals in 1D	2	Yes
Linear Classifiers in 2D	3	Yes
All subsets of $\mathbb{R}$	∞	❌ No (not PAC-learnable)

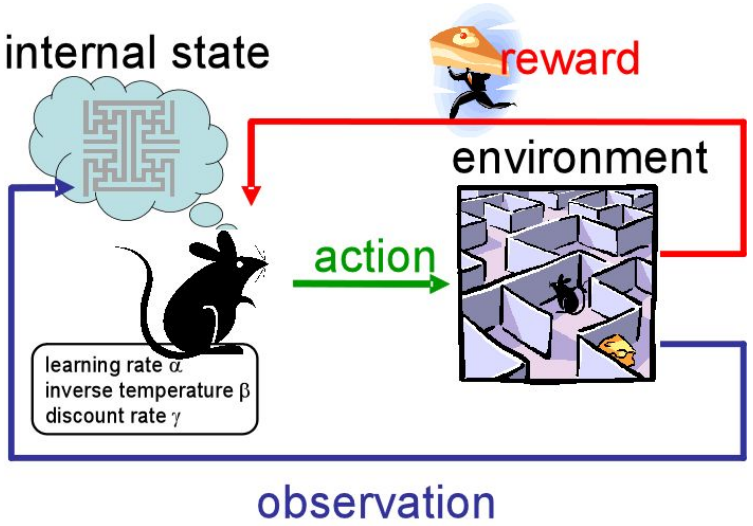
Intuition: What Makes a Class PAC-Learnable?

- **To be PAC-learnable, a class must have:**

- **Finite VC dimension (limited capacity to overfit)**
- **Efficient learning algorithm (computable in reasonable time)**
- **Sufficient data to learn from**

What is Reinforcement Learning (RL)?

- Reinforcement Learning is a type of machine learning where an agent learns by interacting with its environment, using trial and error, and receiving feedback in the form of rewards or punishments.
- It's neither supervised nor unsupervised learning.



Key Comparisons

Supervised Learning

You're given labeled data - inputs with corresponding outputs (e.g., image → "cat").

The model learns to map inputs to outputs.

Examples: Classification, Regression.

Unsupervised Learning

You're given unlabeled data, with no clear output.

The goal is to find structure or patterns (e.g., clustering, association rules).

Reinforcement Learning

You're not told the right answer.

You interact with an environment, try actions, and observe the outcomes (rewards/punishments).

Learning is driven by experience, not direct supervision.

Illustrative Example: Learning to Ride a Bicycle

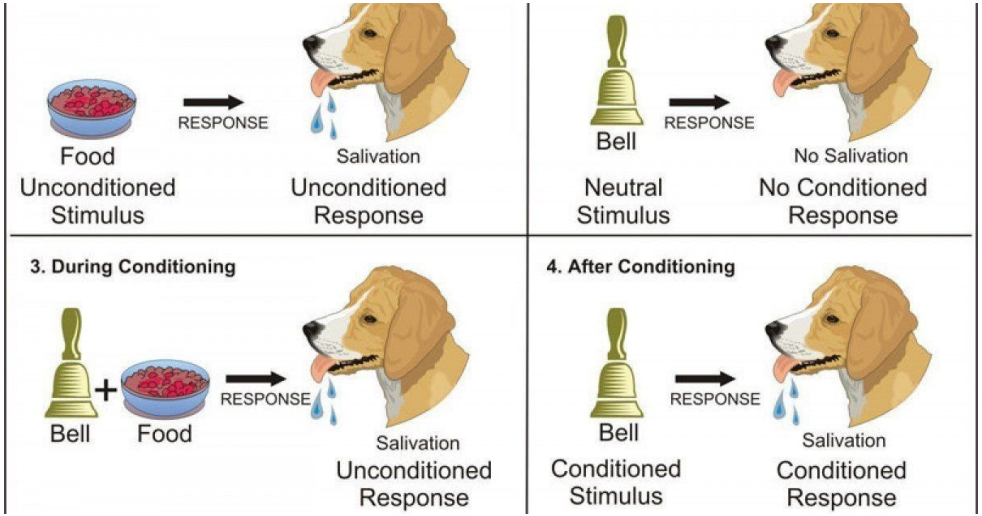
- **Not supervised:** No one gives exact instructions (like move your foot 3 pounds, tilt 2 degrees).
- **Not unsupervised:** You don't just watch people ride and figure it out.
- **It's reinforcement:** You try, fall, learn not to fall. You get minimal feedback — like pain (punishment) or clapping (reward) — and gradually figure it out.

Trial and Error is Key

- The RL agent tries different actions (explores).
- It learns which actions lead to better outcomes over time.
- The delayed nature of rewards adds complexity. You might get feedback well after the action that caused it.

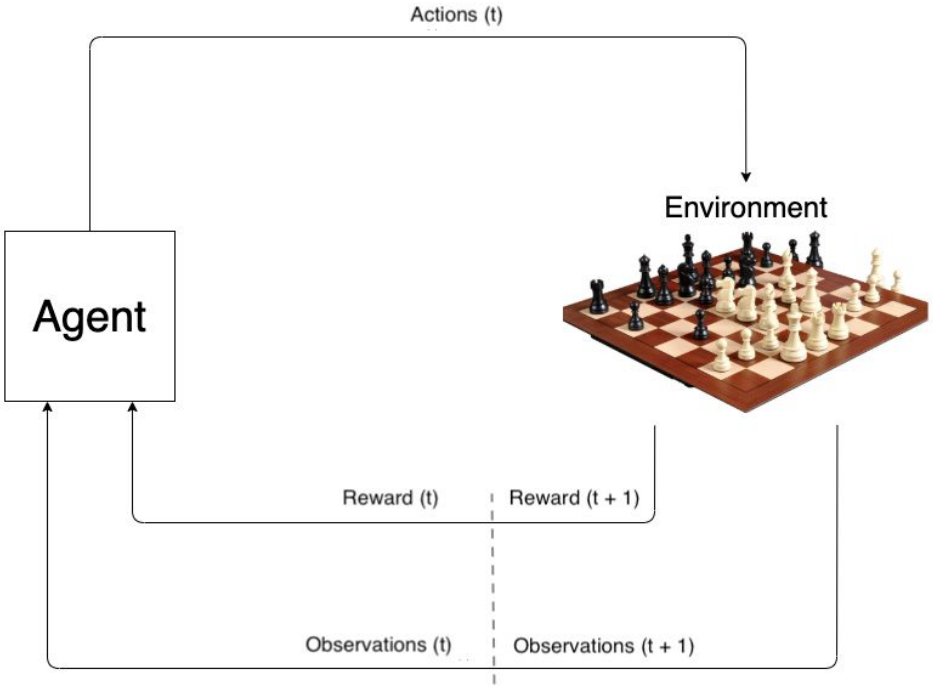
Pavlov’s Dog & RL Origins

- RL is inspired by behavioral psychology — like Pavlov’s classical conditioning.
- The modern field was kickstarted by Sutton & Barto, whose 1983 work laid the foundation for today's algorithms and techniques.



Games as a RL Metaphor

- In chess, if you're told what move to play in a given situation, that's supervised learning.
- If you just play, win or lose, and only get a reward at the end, that's reinforcement learning.
- You must figure out which moves led to winning, despite delayed rewards.



Reinforcement Learning (RL)

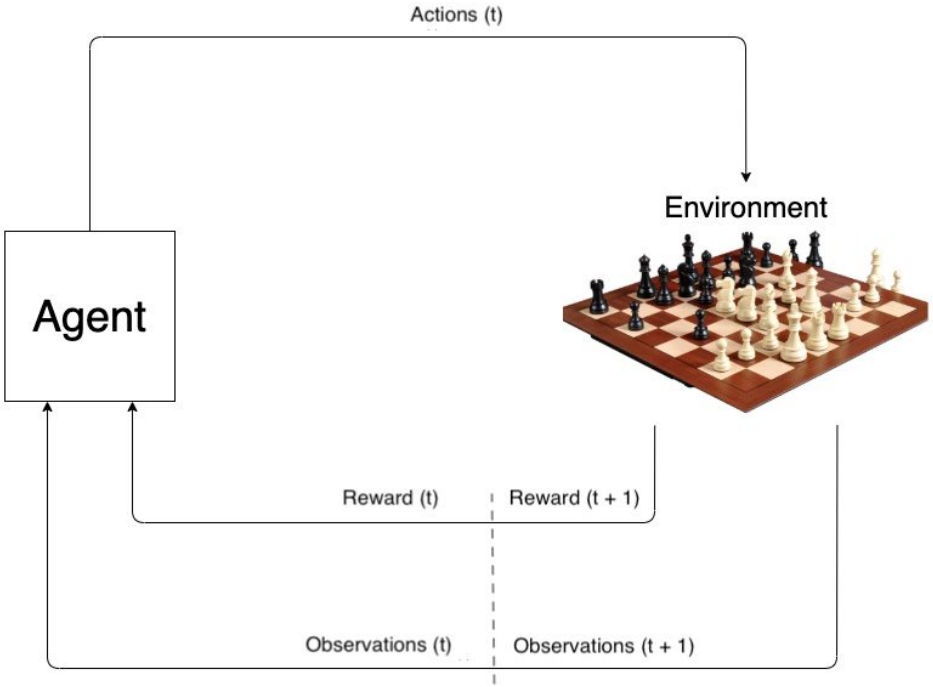
- RL involves an agent learning through interaction with an environment (e.g., helicopter, game board, opponent).
- The agent senses the state of the environment and takes actions to influence it.
- Key challenge: Actions must consider long-term benefits, not just immediate rewards (e.g., chess strategy).
- Reward signal: Scalar feedback from the environment (e.g., +1 for winning, -100 for crashing).
- Biological analogy: Rewards/punishments are interpreted from sensory inputs (e.g., pain as negative feedback).

Key Comparisons

Supervised Learning	Unsupervised Learning	Reinforcement Learning
Input → Output (with target labels). Error signal guides learning (e.g., gradient descent).	Input → Pattern detection (no explicit labels).	Input → Action → Scalar reward (no target labels). Trial-and-error learning (exploration required).

Temporal Difference (TD) Learning

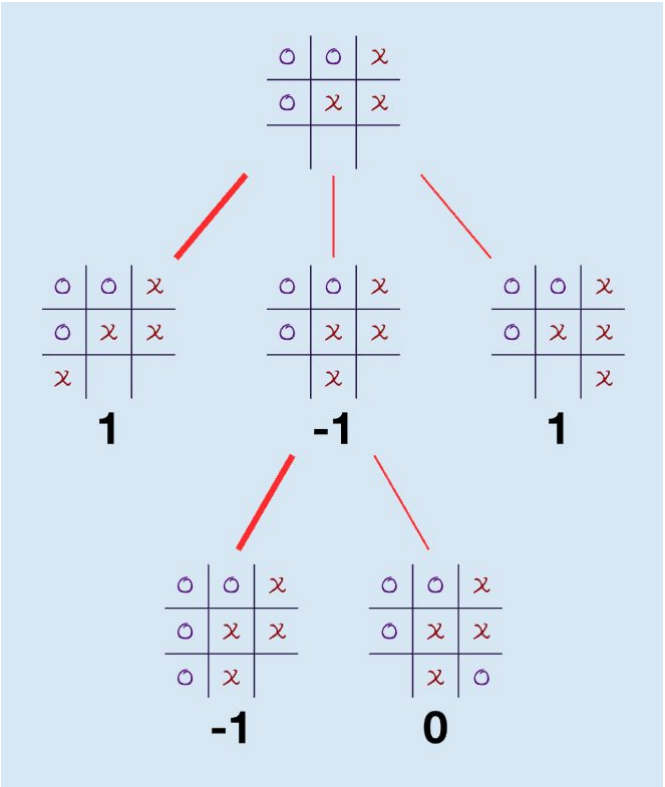
- **Core Idea: Predict future rewards by updating estimates based on later predictions.**
- **Prediction at time t+1 is more accurate than at t.**
- **Example: Chess—confidence in winning increases as the game progresses.**
- **Update Rule: Adjust earlier predictions using newer information (e.g., reduce winning probability from 0.6 to 0.55 if later evidence suggests lower odds).**
- **Biological basis: Similar to dopamine-driven learning in brains.**



Tic-Tac-Toe as an RL Problem

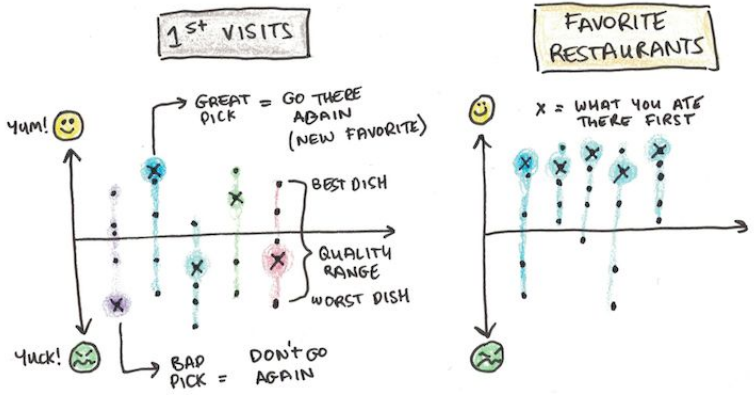
- States: Board configurations.
- Actions: Placing X or O in empty cells.
- Rewards:
  - +1 for winning, 0 otherwise (binary).
  - Alternative: +1 (win), -1 (lose), 0 (draw).

- Key Points:
- Learn a value function: Expected reward from each state (probability of winning).
- Update values via TD learning or end-game feedback.
- Imperfect opponent required for meaningful learning (perfect play leads to always draws).

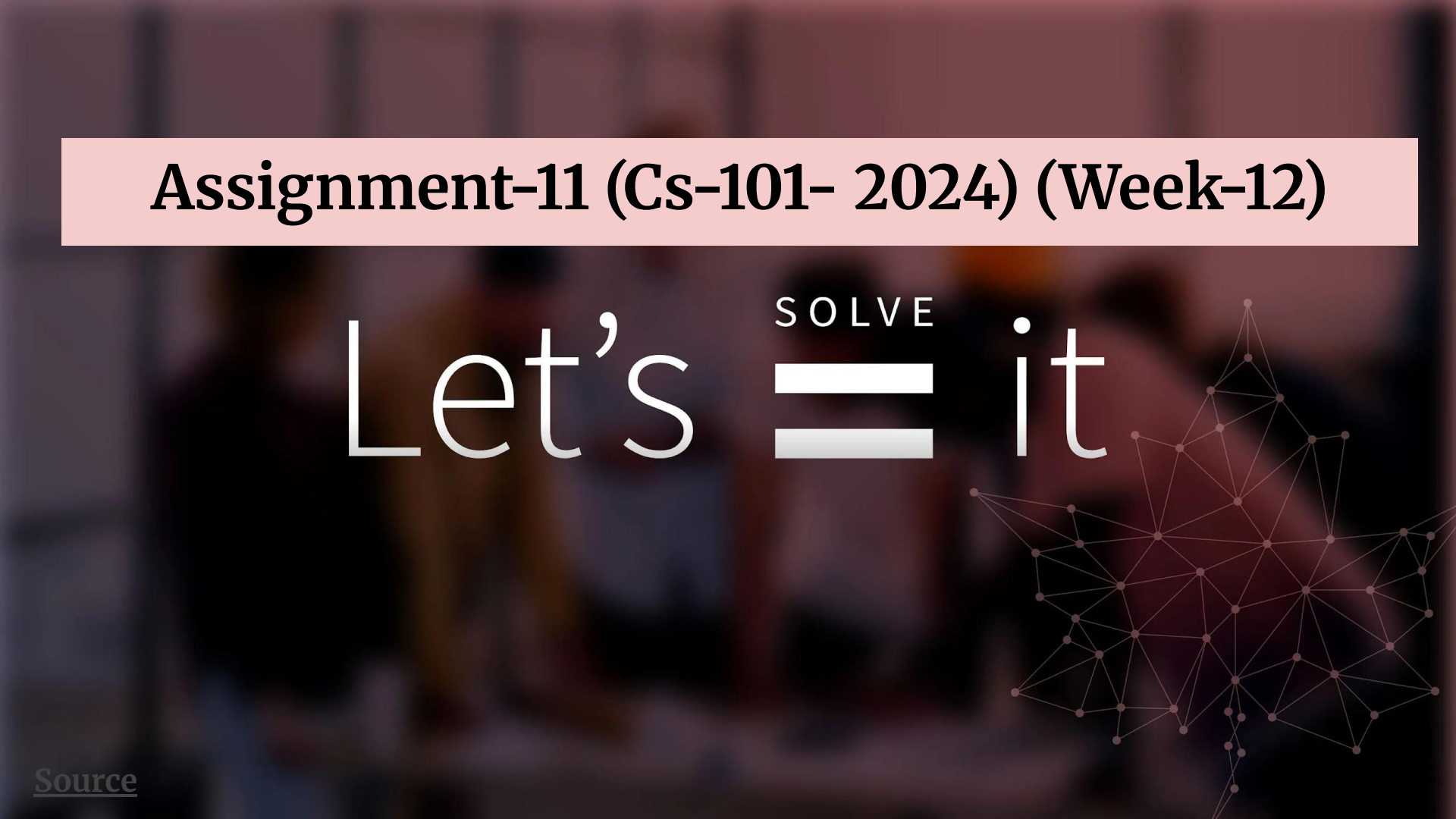


Exploration vs. Exploitation

- Exploitation: Choose actions with the highest known rewards.
- Exploration: Try suboptimal actions to discover better strategies.
- Challenge: Balancing exploration (learning) vs. exploitation (performance).
- Bandit Problems: Simplified RL focusing on this trade-off (no sequential states).
- Example: In tic-tac-toe, occasionally pick random moves to avoid repeating suboptimal paths.



Assignment-11 (Cs-101- 2024) (Week-12)



Source

Question-1

01:00

What is the VC dimension of the class of linear classifiers in 2D space?

a) 2
b) 3
c) 4
d) None of the above

Question-1- Correct answer

What is the VC dimension of the class of linear classifiers in 2D space?

a) 2

b) 3

c) 4

d) None of the above

Correct options: (b) Any 3 points can be classified using a linear decision boundary

Question-2

01:00

Which of the following learning algorithms does NOT typically perform empirical risk minimization?

a) Linear regression
b) Logistic regression
c) Decision trees
d) Support Vector Machines

Question-2 - Explanation

01:00

Which of the following learning algorithms does NOT typically perform empirical risk minimization?

SVM, is called structural risk minimization because they have an additional constraint that is there apart from the empirical they also try to minimize the solution size.

They try to minimize the norm of the weight factor so that actually gives rest to a different kind of minimization.

So it does not do empirical, they called structural risk minimization.

Question-2- Correct answer

Which of the following learning algorithms does NOT typically perform empirical risk minimization?

a) Linear regression

b) Logistic regression

c) Decision trees

d) Support Vector Machines

Correct options: (d).

Question-3

03:00

Statement 1: As the size of the hypothesis class increases, the sample complexity for PAC learning always increases.
Statement 2: A larger hypothesis class has a higher VC dimension. Choose the correct option:

a) Statement 1 is true. Statement 2 is true. Statement 2 is the correct reason for statement 1
b) Statement 1 is true. Statement 2 is true. Statement 2 is not the correct reason for statement 1
c) Statement 1 is true. Statement 2 is false
d) Both statements are false

Question-3

03:00

Statement 1: As the size of the hypothesis class increases, the sample complexity for PAC learning always increases.
Statement 2: A larger hypothesis class has a higher VC dimension. Choose the correct option:

Statement 1: As the size of the hypothesis class increases, the sample complexity for PAC learning always increases.

✅ True:

In PAC learning, the **sample complexity** (i.e., the number of training samples needed for probably approximately correct learning) generally **increases with the capacity** of the hypothesis class. Larger hypothesis classes can express more functions, so more data is needed to rule out bad hypotheses.

However, note:

- It's **not always strictly increasing** with **just the count of hypotheses**, especially in infinite hypothesis classes.
- But **in general**, bigger capacity → higher sample complexity.

Question-3

03:00

Statement 1: As the size of the hypothesis class increases, the sample complexity for PAC learning always increases.
Statement 2: A larger hypothesis class has a higher VC dimension. Choose the correct option:

Statement 2: A larger hypothesis class has a higher VC dimension.

✅ **True** in general:

A larger hypothesis class **can shatter more points**, so its **VC dimension tends to be higher**. This is a good measure of the class's expressiveness.

But this is not **always** a direct correlation — there are **counterexamples** where two hypothesis classes can have the same VC dimension but different sizes.

Question-3

03:00

Statement 1: As the size of the hypothesis class increases, the sample complexity for PAC learning always increases.
Statement 2: A larger hypothesis class has a higher VC dimension. Choose the correct option:

Why **Statement 2 is NOT the correct reason** for Statement 1?

- **VC dimension** is the key quantity that determines **sample complexity** — not just **size of the hypothesis class**.
- So the correct reason for Statement 1 is **VC dimension**, not the **size** itself.
- Hence, **Statement 2 is true**, but **not the correct reason** for Statement 1.

Question-3 - Correct answer

Statement 1: As the size of the hypothesis class increases, the sample complexity for PAC learning always increases.
Statement 2: A larger hypothesis class has a higher VC dimension. Choose the correct option:

a) Statement 1 is true. Statement 2 is true. Statement 2 is the correct reason for statement 1
b) Statement 1 is true. Statement 2 is true. Statement 2 is not the correct reason for statement 1
c) Statement 1 is true. Statement 2 is false
d) Both statements are false

Correct options: (b)

Question-4

03:00

When a model's hypothesis class is too small, how does this affect the model's performance in terms of bias and variance

a) High bias, low variance

b) Low bias, high variance

c) High bias, high variance

d) Low bias, low variance

Question-4 - Correct answer

When a model's hypothesis class is too small, how does this affect the model's performance in terms of bias and variance

a) High bias, low variance

b) Low bias, high variance

c) High bias, high variance

d) Low bias, low variance

Correct options: (a)

Question-5

01:00

Imagine you're designing a robot that needs to navigate through a maze to reach a target. Which reward scheme would be most effective in teaching the robot to find the shortest path

a) +5 for reaching the target,-1 for hitting a wall
b) +5 for reaching the target,-0.1 for every second that passes before the robot reaches the target.
c) +5 for reaching the target,-0.1 for every second that passes before the robot reaches the target, +1 for hitting a wall.
d) -5 for reaching the target, +0.1 for every second that passes before the robot reaches the target.

Question-5 - Correct answer

Imagine you're designing a robot that needs to navigate through a maze to reach a target. Which reward scheme would be most effective in teaching the robot to find the shortest path

a) +5 for reaching the target,-1 for hitting a wall

b) +5 for reaching the target,-0.1 for every second that passes before the robot reaches the target.

c) +5 for reaching the target,-0.1 for every second that passes before the robot reaches the target, +1 for hitting a wall.

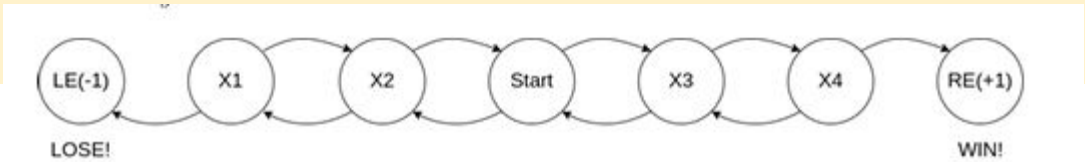
d) -5 for reaching the target, +0.1 for every second that passes before the robot reaches the target.

Correct options: (b) The +5 reward for reaching the target encourages goal achievement, while the-0.1 penalty for each second promotes finding the shortest path. Omitting rewards for hitting walls as question has nothing in this regard

Question-6-8

03:00

For the rest of the questions, we will follow a simplistic game and see how a Reinforcement Learning agent can learn to behave optimally in it.
This is our game:



a) At the start of the game, the agent is on the Start state and can choose to move left or right at each turn.
b) If it reaches the right end(RE), it wins and if it reaches the left end(LE), it loses.
c) Because we love maths so much, instead of saying the agent wins or loses, we will say that the agent gets a reward of +1 at RE and a reward of-1 at LE. Then the objective of the agent is simply to maximum the reward it obtains!

Question-6

03:00

For each state, we define a variable that will store its value. The value of the state will help the agent determine how to behave later. First we will learn this value. Let V be the mapping from state to its value.

Initially,

V(LE) =-1

V(X1) = V(X2) = V(X3) = V(X4) = V(Start) = 0

V(RE) = +1

For each state $S \in \{X1,X2,X3,X4,Start\}$, with SL being the state to its immediate left and SR being the state to its immediate right, repeat: $V(S) = 0.9 \times \max(V(SL),V(SR))$

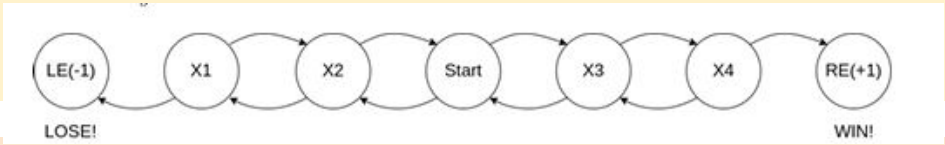
Till V converges (does not change for any state).

What is V(X4) after one application of the given formula?

a) 1
b) 0.9
c) 0.81
d) 0

Question-6 - Correct answer

What is V(X4) after one application of the given formula?



a) 1
b) 0.9
c) 0.81
d) 0

$$V(X4) = 0.9 \times \max(V(X3), V(RE))$$

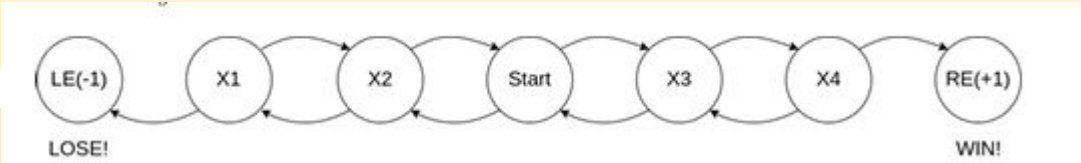
$$V(S) = 0.9 \times \max(0, +1) = 0.9$$

Correct options: (b)

Question-7

03:00

What is V(X1) after one application of given formula?



(a)-1

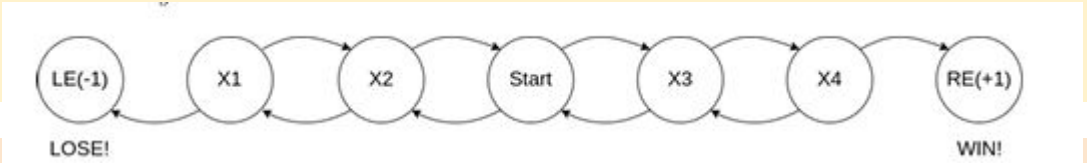
(b)-0.9

(c)-0.81

(d) 0

Question-7 - Correct answer

What is V(X1) after one application of given formula?



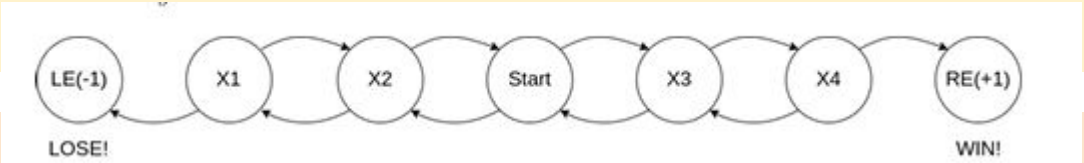
a) -1
b) -0.9
c) -0.81
d) 0 - $V(X1) = 0.9 \times \max(V(LE), V(X2))$ $V(S) = 0.9 \times \max(-1, 0) = 0$

Correct options: (d)

Question-8

03:00

What is V(X1) after V converges?



a) 0.59

b) -0.9

c) 0.63

d) 0

Question-8 - Correct answer

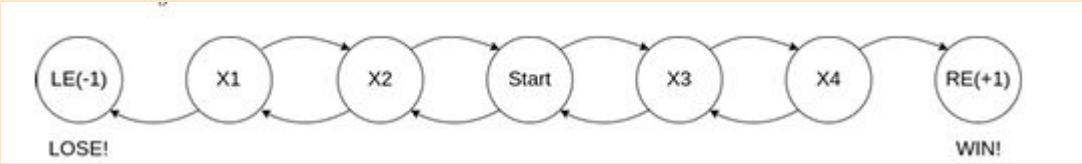
What is V(X1) after V converges?

a) 0.59 - V(X4) = 0.9 → V(X3) =0.81 → V(Start) = 0.729 → V(X2) = 0.656 → V(X1) = 0.59 Final value for X1 is 0.59.

b) -0.9

c) 0.63

d) 0



Correct options: (a)


THANK YOU

Suggestions and Feedback

Next Session:

Wednesday:
16-Apr-2025
6:00 - 8:00 PM