Financial Data Science

Lecture 1
Introduction to Financial Data Science

Liu Peng
liupeng@smu.edu.sg

Pre-class, in-class, and after-class activities

Pre-class learning resources

Self-paced learning, covering essential contents to be covered in class

Group homework

Takeaway homework to be completed as a group and submitted before the next class

In-class quiz

Reinforce understanding via practice and discussion

Student discussions

Open forum for Q&A and discussions

Final project

Practice and deepen understanding of learned knowledge

Weekly lesson plan

Session	Topics	Assignments/Activities
1	Introduction to Financial Data Science	
2	Data Preprocessing and Feature Engineering	Group assignment
3	Foundational ML Models in Finance – Linear Regression	Project guideline release
4	Foundational ML Models in Finance – Logistic Regression	Group assignment
5	Portfolio Optimization with ML	
6	Tree-Based Models	Group assignment
7	Neural Networks	
8	Deep Reinforcement Learning	Project presentation
9	Model Evaluation and Explainability	Project presentation
10	Course Review and Final Team Presentation	Project presentation
11	FINAL EXAM (Closed book)	

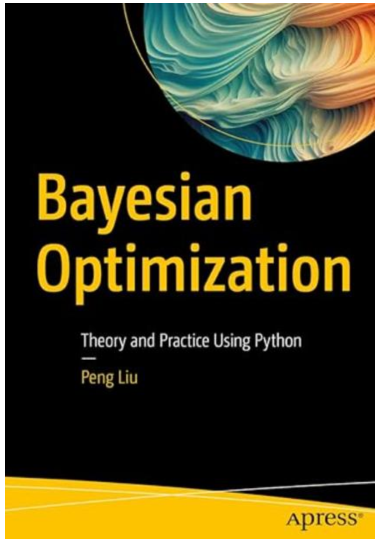
Course assessment rubics

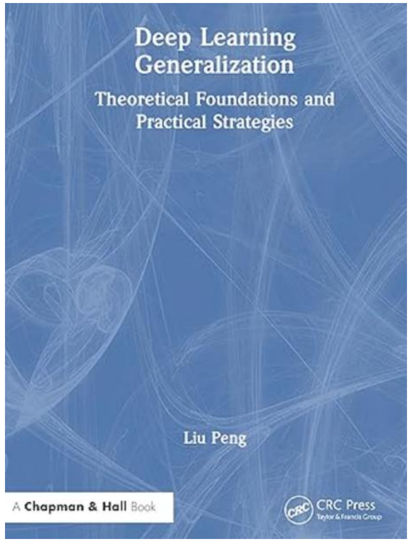
Class participation	Group assignment	Final exam
• 20% • Engagement in discussion, critical thinking	• 40% • Team work, problem solving, communication	• 40% • Problem solving, decision making

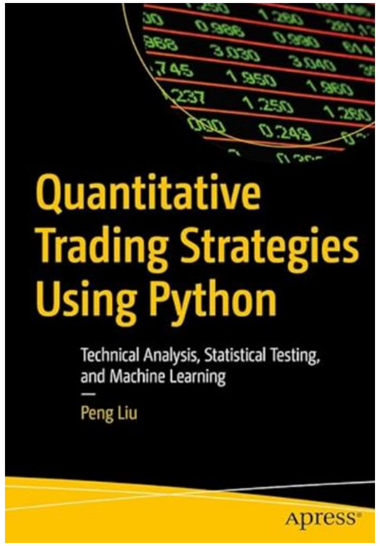
References and resources

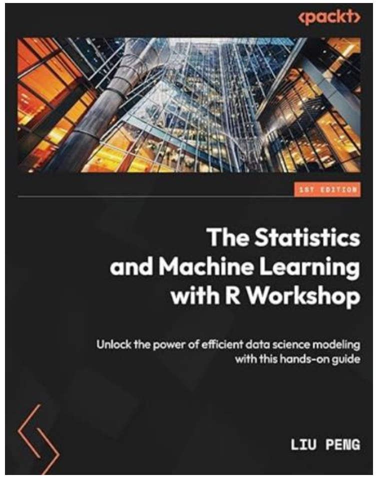
- Books:
  - Bayesian Optimization: Theory and Practice Using Python, Liu Peng, Apress
  - Quantitative Trading Strategies with Python, Liu Peng, Apress
  - The Statistics and Machine Learning with R Workshop, Liu Peng, Packt
  - Deep Reinforcement Learning in Portfolio Optimization, Liu Peng, CRC (upcoming)
  - Deep Learning Generalization, Liu Peng, CRC (upcoming)
  - Quantitative Risk Management with Python, Liu Peng, Apress (upcoming)

- Papers: See reference papers for each session





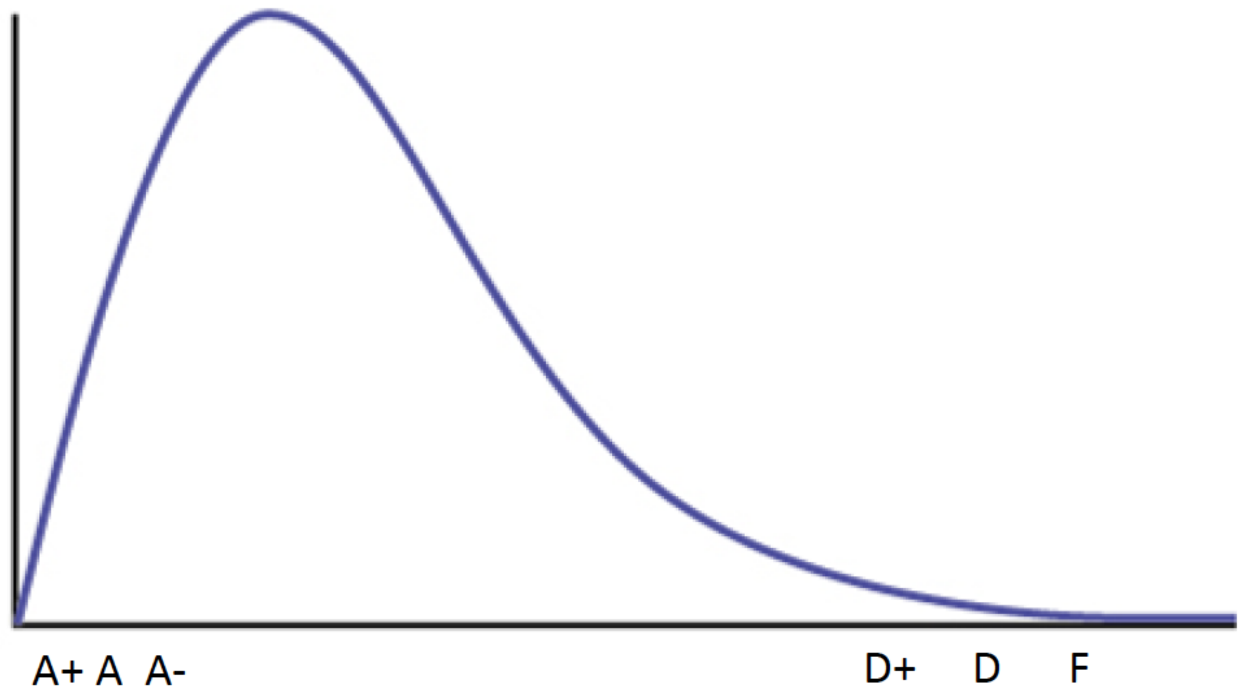




Grading curve

Not drawn to scale

Exact distribution is class-specific and confidential



Office consultation hours

10am-12pm, Fri

- Email me before you come

Room 5118, LKCSB

Alternatively, can send me an email to book other slots

Quick self-introduction

- Your name and hobby

- Form class groups

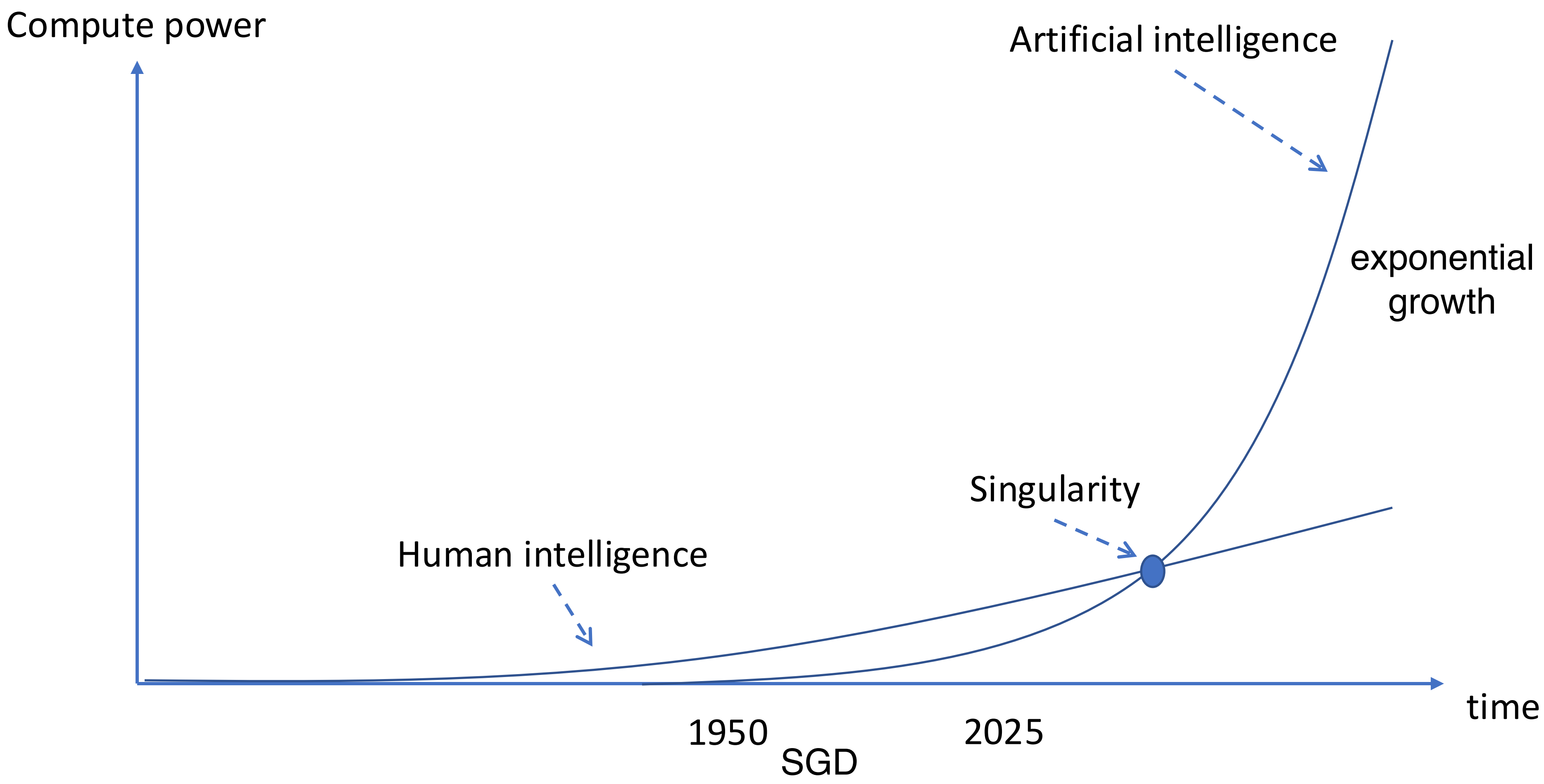
Learning outcomes

- Course outlook overview
- Ways to engage in learning and discussion
- Getting to know data science
- Understand major Financial ML applications
- Python basics

Tutorials and References

- Introduction to machine learning https://youtu.be/00t53nPpbnU
- Python programming basics https://youtu.be/u5mSDRCoaEo
- Downloading and visualizing stock prices with Python https://youtu.be/ngPjj93B5kE
- Chapter 1&2, The Statistics and Machine Learning with R Workshop

Artificial (General) Intelligence



risk

mkt

credit

ops

model

Financial Data Science

Finance

Data Science

Intersection

What data do we work with?

- Tabular structured data
- Image pixel [0, 255], color RGB
- Text input —> model —> output

  structured

Common Data Structure

- Scalar 3
    - Boolean 0 / 1
    - Numeric 2, 3
    - Categorical A / B
- Vector [1, 2, 3]
- List [1, A, 3]
- Matrix
- Dataframe
- Tensor

In-class quiz

Q1-2

Common Control Logic

- True or False
- If-else
- Loop
  - For
  - While
- Function

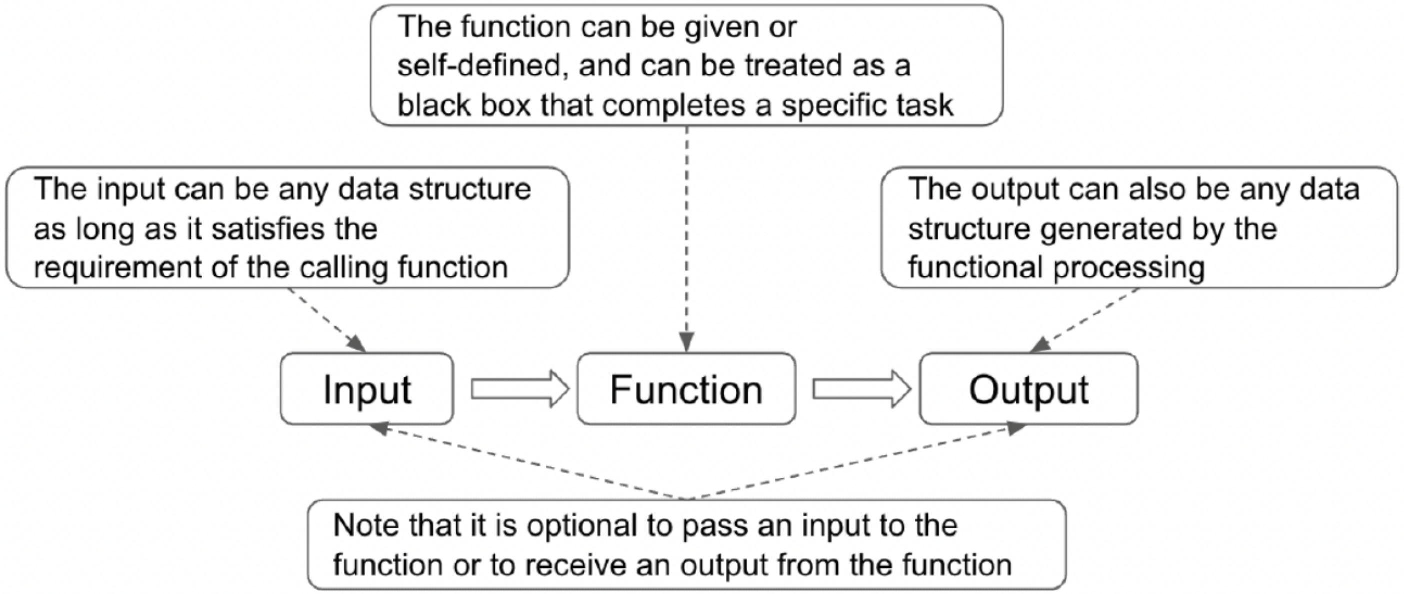


Figure 1.9 – Illustration of a function's workflow

Source: Chapter 1, The Statistics and Machine Learning with R Workshop

Different Data Representation

- Wide format
- Long format
- One-hot encoding

Merging Two Datasets

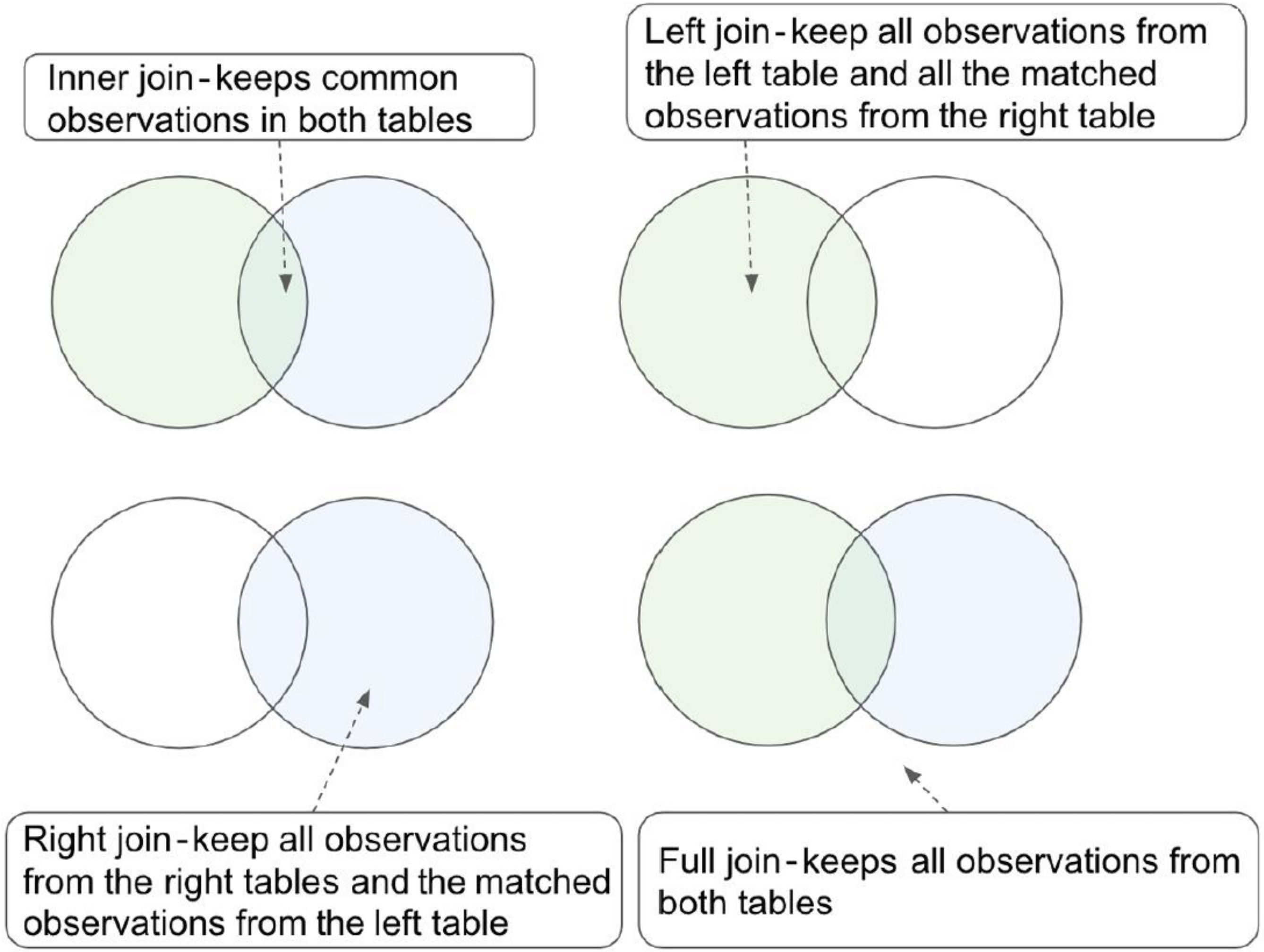


Figure 2.2 – Four different joins commonly used in practice

In-class quiz

Q3-7

Three Main Types of Models

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

How to build a good model?

https://playground.tensorflow.org/

Case study: large language models (LLM)

- **Definition:** LLMs are neural networks designed to generate and understand human-like text based on vast amounts of data.
- **Supervised Learning Paradigm:** LLMs are typically trained using supervised learning, where the model learns to map input sequences (e.g., text) to target sequences (e.g., next word or sentence).

- The training process involves minimizing cross-entropy loss using gradient-based optimization.
- Regularization techniques and evaluation metrics are crucial for model performance.

- Exploring more sophisticated models, optimization strategies, and data augmentation techniques to enhance LLM capabilities.

LLM: Data and Representation

- **Training Data:**
  - Consists of large corpora of text, where each text sequence $\mathbf{x}$ is paired with a target sequence $\mathbf{y}$.
  - Data pairs: $(\mathbf{x}_i, \mathbf{y}_i)$ for $i = 1, 2, ..., N$, where N is the number of samples.

Q: how many data pairs can we get from a sentence?

- **Tokenization:**
  - Text is tokenized into a sequence of tokens $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{im})$.
  - Embedding $E : \mathcal{V} \to \mathbb{R}^d$ maps each token x_{ij} from vocabulary $\mathcal{V}$ to a d-dimensional vector.
- **Input Representation:**
  - Input sequence: $\mathbf{X}_i = (E(x_{i1}), E(x_{i2}), ..., E(x_{im}))$, where $\mathbf{X}_i \in \mathbb{R}^{m \times d}$.

LLM: Model Architecture

- **Neural Network:**
  - Typically uses transformers with layers of self-attention and feed-forward networks.
  - Model parameters: θ (weights of the neural network).

    Q: how many parameters does a typical LLM have?
  - Output: Predicted sequence $\hat{\mathbf{y}}_i$.
- **Sequence Prediction:**
  - For each input $\mathbf{X}_i$, the model predicts the next token $\hat{y}_{i,j+1}$ conditioned on previous tokens.
  - Probability distribution: $P(\hat{y}_{i,j+1} | \mathbf{X}_i, \theta)$.

    Q: how does this influence LLM output?

LLM: Loss Function

- **Cross-Entropy Loss:**
  - Measures the difference between the predicted probability distribution and the true distribution.
  - Loss for a single sequence:

$$\mathcal{L}_i(\theta) = -\sum_{j=1}^{m} \log P(y_{ij}|\mathbf{X}_i, \theta)$$

Q: is this differentiable?

  - Total loss over all training samples:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i(\theta)$$

- **Goal:** Minimize the loss $\mathcal{L}(\theta)$ by adjusting model parameters θ.

LLM: Optimization

- **Gradient Descent:**
  - Update rule for model parameters:
$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t)$$
  - η: Learning rate.

Q: how to tune this hyperparameter?

- **Backpropagation:**
  - Computes gradients $\nabla_\theta \mathcal{L}(\theta)$ through the network layers.
  - Uses chain rule to propagate errors from the output to the input layer.
- **Stochastic Gradient Descent (SGD):**
  - Often used for large datasets, updating parameters using mini-batches:
$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}_{\text{mini-batch}}(\theta_t)$$

In-class quiz

Q8-11

Flavors of machine learning

Case Study: Unsupervised Learning

Hidden Markov Model (HMM) for Market Regime Detection

- **States** S_t:
  - Hidden market regimes (e.g., Bull, Bear, Neutral).
  - S_t represents the hidden state at time t.
- **Observations** O_t:
  - Observable market variables (e.g., returns).
  - O_t is the observed data at time t.
- **Transition Probabilities** $P(S_t|S_{t-1})$:
  - Probability of moving from one state to another.
  - Transition matrix A:

$$A = \{a_{ij}\}, \quad a_{ij} = P(S_t = j|S_{t-1} = i)$$

- **Emission Probabilities** $P(O_t|S_t)$:
  - Probability of observing O_t given state S_t.
  - Modeled as Gaussian:

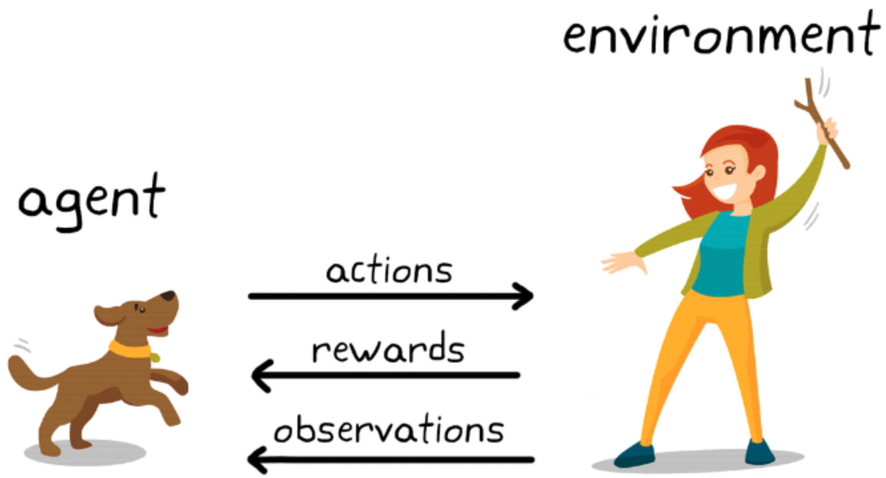
$$P(O_t|S_t = j) = \mathcal{N}(O_t|\mu_j, \sigma_j^2)$$

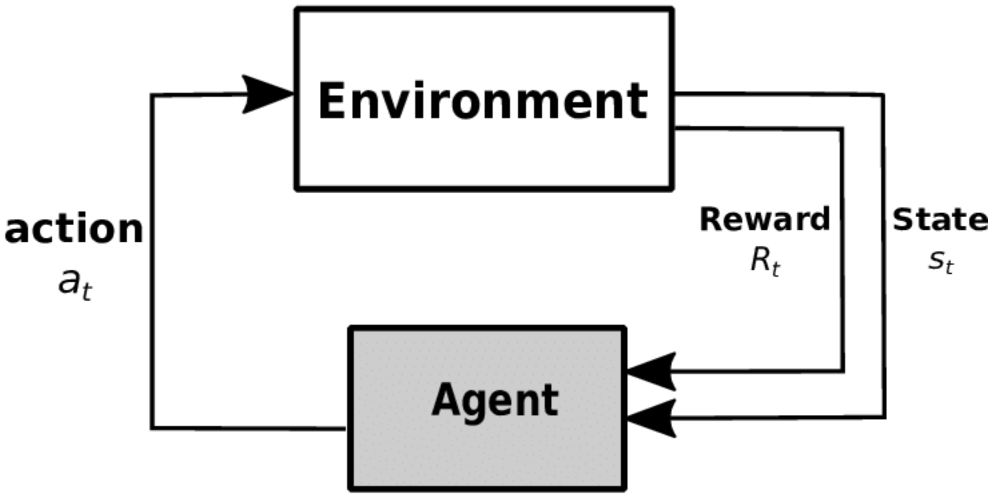
- **Initial State Distribution** π:
  - Probability of each state at $t = 0$:

$$\pi = P(S_0 = i)$$

Flavors of machine learning

Case Study: Reinforcement Learning





model f

Source: Mathworks

Q-Learning

Let’s play a game!

- **Context:**
  - Q-Learning is a model-free reinforcement learning algorithm used to learn optimal action-selection policies in environments modeled as Markov decision processes (MDPs).
- **Q-Value Update (Bellman Equation):**

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a')$$

- **Q-Learning Update Rule:**

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

- **Optimal Policy:**

$$\pi^*(s) = \arg\max_{a} Q(s,a)$$

Reading materials

- https://www.coursera.org/articles/machine-learning-in-finance
- Deep learning in finance and banking: A literature review and classification
- An Overview of Machine Learning, Deep Learning, and Reinforcement Learning-Based Techniques in Quantitative Finance: Recent Progress and Challenges
- Machine Learning for Quantitative Finance Applications: A Survey

Homework

- Watch/review video tutorials and class recording for week 1 lecture (if you have not done so)
- Post learning reflections and questions in the group chat if any
- Form groups, get to know your teammates and discuss ideas for final project