MATH0062 (Mathematics and Statistics of Algorithmic Trading)

Year:	2024{2025
Code:	MATH0062
Level:	Masters
Value:	15 UCL credits (= 7.5 ECTS credits)
Term:	2
Structure:	On campus
Assessment:	100% examination. Students must achieve at least 50% to pass this course.
Normal Pre-requisites:	MATH0088 Quantitative and Computational Finance
Lecturer:	Dr A Tse

Course Description and Objectives

This module introduces the theories and practices of algorithmic trading, where we will see how statistical and mathematical models can be utilised to guide traders what strategies to be employed under di erent contexts. Some examples include:

- How to generate trading signals to recommend if a trader should long/short a particular asset?
- How to optimally construct a portfolio of di erent assets to achieve certain risk-return objectives?
- How to dynamically hedge a derivative product under a more complicated model beyond the Black-Scholes paradigm?
- A trader wants to sell a large quantity of stock. Submitting this large order to the market in a single trade will create market impact and move the price against the trader. Then how should this order be optimally broken down into smaller pieces and executed over time to minimise the cost associated with market impact?

This module consists of three parts:

<u>Part 1: neural network</u>

In the rst part of the module, we will focus on \neural network" { a class of machine learning models that generalise standard regression models and are capable of describing data with complex structure. This is the backbone of many cutting-edge arti cial intelligence technologies such as image classi cation and AlphaGo (the computer program that managed to defeat a world champion of Go).

Important concepts such as universal approximation theorem, activation function and loss function will be introduced. Then we illustrate how a neural network can be trained via the ideas of backpropagation, stochastic gradient descent, etc.

As case studies, we will demonstrate how neural networks can be constructed and trained to predict the movement of exchange rate, to approximate the weights on di erent assets within an optimal portfolio, and to recommend the delta-hedge of a derivative product. These will be showcased via Python Notebooks.

Part 2: stochastic control

The second part of the module is about \stochastic control" and its applications to algorithmic trading. Stochastic control is a subfield of mathematical optimisation where an agent wants to choose the best action over time to achieve a certain objective in a random environment characterised by some stochastic processes. There are many applications in finance that can be formulated as stochastic control problems. Examples include how a trader with a large quantity of stock to sell should set the number of shares to be sold at each point of time to maximise the sale proceed (the optimal execution problem), and how much capital investors should put in some risky stocks over time based on their risk-return requirements (the portfolio optimisation problem).

We will begin by introducing the essential theories of stochastic control such as dynamic programming principle and Hamilton-Jacobi-Bellman equation. It turns out that the solution of a stochastic control problem is closely tied to the solution of a partial differential equation. Then we will specialise to the optimal execution problem and describe how the problem under different types of market impact can be modelled and solved mathematically. We will also explore some other applications of stochastic control such as portfolio optimisation and pairs trading.

Part 3: reinforcement learning

The final part of this module provides a brief introduction to \reinforcement learning". Similar to stochastic control, reinforcement learning guides how an agent should behave optimally in a random environment. However, a key difference is that the probabilistic structure of the environment is usually unknown a priori. A distinctive challenge is to balance exploration (to gain information about the unknown environment) and exploitation (to act in the most reward-efficient way). We will introduce several model-free learning algorithms, and if time permits, we will cover advanced topics such as actor-critic methods and Deep-Q learning.