MATHEMATICAL TOOLS IN FINANCE: PROJECTS

GREGORY BERKOLAIKO

Introduction

This document contains several projects for the class "Mathematical Tools in Finance", which are specified with a varying degree of detail. Most involve some open exploration with particular aspect to explore left to the discretion of the student.

1. Poject: Seasonality in data

Consider the monthly average household electricity consumption in Austin, TX (courtesy of Austin Energy and available at `https://doi.org/10.26000/040.000001`).

(1) Extract 5 years worth of consecutive data points (check that there no gaps) from the kWh column. Perform Discrete Fourier Transform, for example by multiplying the vector of length 60 you extracted by the DFT matrix with entries

$$F_{jk} = \frac{1}{\sqrt{N}} e^{2\pi i \frac{jk}{N}},$$

where $N = 60$ in our case. Plot the resulting transformed vector (entry value versus entry coordinate). Discuss particular features of the data: meaning and location of the peaks, symmetries.

(2) Extract 10 years worth of data and repeat the process. How did the location of the peaks and the values there change and why?

(3) Formulate your findings as an informal rule-of-thumb of the form "if the data has period n, the DFT coefficients will exhibit a peak at location(s) L".

2. Project: Principal Component Analysis of the stock returns

Find and download daily close data for the following stocks: AAL, AMD, AVGO, BAC, BRK-A, CVX, DAL, EOG, INTC, JPM, LUV, NVDA, SLB, UAL, WFC, XOM. Repeat the analysis of Sections 2.1-2.2 of the paper by Avellaneda and Lee (posted). Explain how the signs of the second and third eigenvectors partition the stocks into groups. Does this partition correspond to the grouping of the stocks by industries?

3. Project: "Plus one, minus one" game

A casino offers the following game: the customer is given a well-shuffled deck of $N = 52$ cards. The cards are turned over one-by-one and the customer gets \$1 for each black card and -\$1 for each red card. The customer can stop at any time, collecting their winnings (if the customer is losing, there is no reason to stop: once the whole deck is used up, the customer is automatically back to \$0). Design the procedure and write code to

(1) Compute the risk-neutral value of the game.
(2) Determine the optimal stopping strategy.

The risk-neutral value is the expected winnings maximized over all valid strategies. A valid (deterministic) strategy is a function that decides stopping (0 or 1) based only on the information available at the time: number of cards left in the deck and the current winnings balance.

Optional: Now the casino changes the rules. At the start of the game, 10 random cards are removed from the deck without revealing their color. Adjust your code to compute the risk-neutral value of this game.

Hints/Remarks: The problem is similar to the American Put/Call optimal exercise problem that is handled so well by binary trees / dynamical programming. In this problem, the probabilities change depending on the tree node. It is best to vizualize your stopping condition as the least balance required for stopping, as a function of the remaining number of cards.

For the optional part, there is an easy way to implement the adjustment (as well as some very difficult ones).

4. Project: Ternary tree

The initial price of the stock is S_0. Each time period, the price goes up by 1$, stays the same, or goes down by 1$ with probabilities p_+, p_0 and p_- correspondingly. The task is to construct a dynamical self-financing replicating portfolio for a given payoff. The replicating portfolio will contain cash and the stock.

We can make the following assumptions:

(1) The payoff is a call option with strike $E = S_0$.
(2) The interest rate $r = 0$.
(3) There are $L = 3$ time periods. We can rebalance the hedge at every node.
(4) The probability is binomial $\mathrm{Bin}(2, p)$ — this way the probabilities on the final level of the tree is known explicitly.
(5) The replicating portfolio error is to be measured with probability-weighted L_2-norm (mean square).

Complete the following tasks.

(1) **Direct linear algebra.** For the non-recombining tree, compute the full basis of available assets: cash used to purchase stock at a given node of the tree. Determine the dimension of the space of payoffs. Determine the number of available assets. Can the market be complete? Find the least squares solution and determine the time 0 portfolio composition and price.
(2) **Locally optimal least squares.** Determine the time-0 portfolio price by preforming least squares at every node recursively starting from level L. Design and code a self-financing hedging procedure for an option sold at this price. Estimate the mean square error using Monte Carlo.
(3) **Dynamically optimal least squares.** Code the algorithm in [Černý section 12.4] to obtain the time-0 portfolio price. Note that our tree does not have IID returns. Compare your answer to the “Direct linear algebra” approach. Code the self-financing hedging procedure for an option sold at the price you determined. Estimate the mean square error using Monte Carlo.