Computational Finance – Optimal Trading

1 Measures of Performance

Imagine investing \$1 in a trading strategy, and monitoring the status of your investment at regularly spaced times $t_1, t_2, \ldots t_n$. For convenience we set $t_0 = 0$ and assume that the spacing between the times is τ , so that $t_i = i\tau$. For daily trading strategies, $\tau = 1$ day $= \frac{1}{250}$ years where there are approximately 250 trading days in a year. Let V_i be the value of your investment at time t_i ($V_0 = 1$). The sequence of V_i is known as the profit and loss curve (P&L curve) of the trading strategy.

The return sequence is given by $r_i = \log(V_i/V_{i-1}) \approx (V_i - V_{i-1})/V_{i-1}$ which is the percentage return. The cumulative return curve is given by $R_i = \sum_{\tau=1}^{i} r_{\tau}$. An example P&L curve and its corresponding cumulative return curve are shown below.



The mean τ -period return, μ_{τ} is the average of the returns, $\mu_{\tau} = \frac{1}{n} \sum_{i=1}^{n} r_i$. It is conventional to compute the mean annualized return which scales up this average return to 1 year, so that we define the average annualized return as

$$\mu = \frac{\mu_{\tau}}{\tau} = \frac{1}{n\tau} \sum_{i=1}^{n} r_i = \frac{1}{T} \sum_{i=1}^{n} r_i,$$

where $T = n\tau$ is the entire period of observation. Similarly, we can compute the variance of the returns, $\sigma_{\tau}^2 = \frac{1}{n} \sum_{i=1}^{n} (r_i - \mu_{\tau})^2$. Assuming that each time period is independent, we can compute the annualized variance by scaling up the τ -period variance to 1 year,

$$\sigma^2 = \frac{\sigma_\tau^2}{\tau} = \frac{1}{n\tau} \sum_{i=1}^n (r_i - \mu_\tau)^2 = \frac{1}{T} \sum_{i=1}^n (r_i - \mu_\tau)^2.$$

The annualized volatility is σ . The Sharpe ratio is a risk adjusted measure of performance defined by the ratio of the annualized return and the annualized volatility,

$$Sharpe = \frac{\mu}{\sigma} = \frac{\mu_{\tau}}{\sigma_{\tau}\sqrt{\tau}}.$$

It is often the case that these measures will be defined not with respect to the raw returns, but the excess returns, where the excess is with respect to the risk free rate, rf_i . Thus we define the excess returns by $\bar{r}_i = r_i - rf_i$.

Exercise 1.1

Give linear time algorithms for computing the annualized average return, the anualized volatility and the Sharpe ratio.

The Sharpe ratio is an example of a risk adjusted measure of return because it scales the return by a normalizing factor which is how variable the return is, which is a measure of the risk in the trading strategy. For two trading strategies with the same return, the one with lower volatility (or risk) will have a higher Sharpe ratio. Generally, 3 years is an acceptable track record for a trading strategy, and a Sharpe ratio of 2 or higher over a period of more than 3 years is considered very good in the industry.

There are two properties of the Sharpe ratio that make it a little undesirable. The first is that it penalizes the downside and upside risk equally. So a trading strategy which makes only positive but variable returns can have a low Sharpe ratio.

Exercise 1.2

Given any $\epsilon > 0$, construct a return sequence (for appropriately defined τ) consisting only of positive returns for which the Sharpe ratio is less than ϵ .

One way to alleviate this problem is to only consider the negative returns in defining the risk. Thus we compute the the root mean square of the negative returns, sometimes called the downside deviation. The ratio or the average return to the downside deviation, usually denoted the downside deviation ratio, is another risk adjusted measure of performance which now does not penalize variability in positive returns. This is not often used in practice because there is yet another flaw in the Sharpe ratio which also affects the downside deviation.

Exercise 1.3

Show that any permutation in the return sequence results in the same Sharpe ratio.





In the first one, all the negative returns occur first. In the second one the negative and positive returns alternate. These are two clearly different looking cumulative return curves, yet they will have the same Sharpe ratio.

The maximum drawdown risk measure takes these considerations into account. The maximum drawdown (MDD) of a cumulative return curve is the largest possible loss, assuming you entered and exited the trading strategy at the worst possible time. For the first curve in the above example, the MDD is 5%, but for the second curve, it is approximately 0. For the P&L curve shown at the begining of this chapter, the part of the cumulative return curve realizing the MDD is highlighted in red. Notice that the MDD as a measure of risk does not penalize positive returns, no matter how variable they are, and further, the MDD is sensitive to permutations of the return sequence.

Formally, the maximum drawdown can be defined by viewing the return sequence as a string of numbers. Then, the MDD is the minimum possible substring sum,

$$MDD = \min_{i \le j} \left\{ \sum_{k=i}^{j} r_k \right\}.$$

Thus, a straightforward algorithm to compute the MDD is to consider all possible substrings, and compute the substring sum, taking the minimum.

Exercise 1.4

Show that this algorithm is cubic.

If the minimum of the cumulative return curve occurs after the maximum, then the MDD is the maximum minus the minimum. Otherwise this is not the case, and in general, there is no relationship between the maximum, the minimum and the MDD. A cubic algorithm is not acceptable in practice, and in this case it turns out that one can compute the MDD in linear time.

Exercise 1.5

Let ${\cal R}_i$ denote the cumulative return curve of a trading strategy. Define the drawdown at time i by

$$DD_i = \max_{1 \le k \le i} R_k - R_i$$

which is the previous maximum minus the current value.

- (a) Show that $MDD = \max_i DD_i$.
- (b) Give a linear time algorithm to compute the MDD given as input the return sequence r_1, \ldots, r_n . The algorithm should have a run time which is linear in n.

The MDD is itself a very useful measure of risk. In fact, most hedge funds would like to have small MDD, because large drawdowns lead to fund redemptions. In addition, the Sterling ratio is a very common risk adjusted return measure obtained by dividing the return by the MDD,

$$Sterling = \frac{\mu}{MDD}.$$

Typically, the MDD is calculated over a period of 3 years, and the return is also scaled to 3 years. The choice of 3 years is a conventiional practice that has arisen because until recently, the scaling law for the MDD has not been known, and so the notion of an annualized MDD was not possible. Recently, [?] the behavior of the MDD with time has been computed, and so the notion of an annualized MDD does make sense, and can now be used to obtain standardized risk adjusted measures for funds which have been around for more than 3 years or less than 3 years.

2 Optimal Trading Strategies

A trader has in mind the task of developing a trading system that optimizes some profit criterion, the simplest being the total return. A more conservative approach is to optimize a risk adjusted return. In an environment where markets exhibit frequent crashes and portfolios encounter sustained periods of losses, it should be no surprise that the Sterling ratio and the Sharpe ratio have emerged as the leading performance measures used in the industry.

Given a set of instruments, a trading strategy is a switching function that transfers the wealth from one instrument to another. In this paper, we consider the problem of finding optimal trading strategies, i.e., trading strategies that maximize a given performance metric, on *historical data*. We focus on the total return as the measure of performance, but one can also construct optimal strategies efficiently for variants of the Sharpe and Sterling ratio [?]. Finding the optimal trading strategy for non-zero transactions cost is a path dependent optimization problem even when the price time series is known. A brute force approach to solving this problem would search through the space of all possible trading strategies, keeping only the one satisfying the optimality criterion. Since the number of possible trading strategies grows exponentially with time, the brute force approach leads to an exponential time algorithm¹, which for all practical purposes is infeasible – even given the pace at which computing power grows.

- (i) Knowing what the optimal trades are, one can take an inductive approach to real trading: on historical data, one can construct the optimal trades; one can then correlate various market and/or technical indicators with the optimal trades. These indicators can then be used to identify future trading opportunities. In a sense, one can try to *learn* to predict good trading opportunities based on indicators by emulating the optimal trading strategy. A host of such activity within the inductive framework, goes under the name of *financial engineering*.
- (ii) The optimal trading performance under certain trading constraints can be used as a benchmark for real trading systems. For example, how good is a trading system that makes ten trades with a Sterling ratio of 4 over a given time period? One natural comparison is to benchmark this trading strategy against a Sterling-optimal trading strategy that makes at most ten trades over the same time period.
- (iii) Optimal trading strategies (with or without constraints) can be used to quantitatively rank various markets (and time scales) with respect to their profitability according to a given criterion. So for example, one could determine the optimal time scale on which to trade a particular market, or given a set of markets, which is the most profit-friendly.

¹The asymptotic running time of an algorithm is measured in terms of the input size n. If the input is a time sequence of n price data points, then polynomial time algorithms have run time that is bounded by some polynomial in n. Exponential time algorithms have running time greater than some exponentially growing function in n [?].

(iv) Given a stochastic model for the behavior of a pair of instruments, one can use the efficient algorithms presented here to construct *ex-ante* optimal strategies using simulation. To be more specific, note that the optimal strategy constructed by our algorithms requires full knowledge of the future price paths. The stochastic model can be used to generate sample paths for the instruments. These sample paths can be used to compute the optimal trading strategy given the current history and information set. One then has a sample set of future paths and corresponding optimal trading strategies on which to base the current action. Note that such a stochastic model for future prices would have to take into account correlations (including auto-correlations) among the instruments.

It is beyond the scope of the current discussion to develop these applications. Our main goal here is to present the algorithms for obtaining optimal trading strategies, *given* a price time series.

2.1 Trading Model

We now make the preceding discussion more precise. We consider optimal trading strategies on two instruments, for concreteness, a stock S and a bond B with price histories $\{S_0, \ldots, S_n\}$ and $\{B_0, \ldots, B_n\}$ over n consecutive time periods, $\{[t_0, t_1], [t_1, t_2], \ldots, [t_{n-1}, t_n]\}$. The prices B_i, S_i correspond to the times $t_i, i \in \{0, \ldots, n\}$. We can assume that $t_0 = 0$. Thus, for example, over time period $[t_{i-1}, t_i]$, the price of stock moved from S_{i-1} to S_i .

We denote the return sequence for the two instruments by $\{s_1, \ldots, s_n\}$ and $\{b_1, \ldots, b_n\}$ respectively: $s_i = \log \frac{S_i}{S_{i-1}}$, and correspondingly, $b_i = \log \frac{B_i}{B_{i-1}}$. We assume that one of the instruments is the benchmark instrument, and that all the equity is held in the benchmark instrument at the begining and end of trading. The bond is usually considered the benchmark instrument, and for illustration, we will follow this convention. The trivial trading strategy is to simply hold onto bond for the entire duration of the trading period. It is useful to define the excess return sequence for the stock, $\hat{s}_i = s_i - b_i$. When the benchmark instrument is the bond, the excess return as we defined it is the conventionally used one. However, one may want to measure performances of a trading strategy with respect to the S&P 500 as benchmark instrument, in which case the excess return would be determined relative to the S&P 500 return sequence. The excess return sequence for the bond is just the sequence of zeros, $\hat{b}_i = 0$. Conventionally, the performance of a strategy is measured relative to some trivial strategy, so the excess return sequence will be the basis of most of our performance measures. We make the following assumptions regarding the trading:

A1 [All or Nothing]: The position at all times is either entirely bond or entirely stock.

- A2 [No Market Impact]: Trades can be placed without affecting the quoted price.
- A3 [Fractional Market]: Arbitrary fractions of stock or bond can be bought or sold.
- A4 [Long Strategies]: One can only hold long positions in stock or bond.

Assumption A1 is in fact not the case in many trading funds, for it does not allow legging into a trade, or holding positions in both instruments simultaneously. While this is technicaly a restriction, for many optimality criteria (for example return optimal strategies), one can show that there always exists an all-or-nothing optimal strategy. Thus, we maintain this simplifying assumption for our discussion. Further, such assumptions are typically made in the literature on optimal trading (see for example [?]). Assumptions A2–A4 are rather mild and quite accurate in most liquid markets, for example foreign exchange. Assumption A3 is needed for A1, since if all the money should be transfered to a stock position, this may necessitate the purchase of a fractional number of shares. Note that if $\mathcal{T}[i-1] \neq \mathcal{T}[i]$, then at the begining of time period $[t_{i-1}]$, the position was transfered from one instrument to another. Such a transfer will incur an instantaneous per unit transaction cost equal to the bid-ask spread of the instrument being transfered into. We assume that the bid-ask spread is some fraction (f_b for bond and f_s for stock) of the bid price.

With these constraints in mind, we define a trading strategy \mathcal{T} as a boolean n + 1dimensional vector indicating where the money is at time t_i :

$$\mathcal{T}[i] = \begin{cases} 1 & \text{if money is in stock at time } t_i, \\ 0 & \text{if money is in bond at time } t_i. \end{cases}$$

Exercise 2.1

How many possible trading strategies are there?

We assume that $\mathcal{T}[0] = \mathcal{T}[n] = 0$, i.e., all the money begins and ends in bond. If $\mathcal{T}[i] = 0$ and $\mathcal{T}[i+1] = 1$ then infinitessimally after time t_i , the money is moved from bond to stock. We say that a trade is entered at time t_i . A trade is exited at time t_i if $\mathcal{T}[i] = 1$ and $\mathcal{T}[i+1] = 0$. The number of trades made by a trading strategy is equal to the number of trades that are entered. The return (or excess return) of the trading strategy over time period $[t_i, t_{i+1}]$ depends on the values of $\mathcal{T}[i]$ and $\mathcal{T}[i+1]$. We let $r_{\mathcal{T}}[i]$ for $i \in \{1, 2, \ldots, n\}$ be the vector which contains the returns of the trading strategy over the time period $[t_{i-1}, t_i]$. Then,

$$r_{\mathcal{T}}[i] = \begin{cases} b_i & \text{if } \mathcal{T}[i-1] = 0, \mathcal{T}[i] = 0; \\ b_i - f_b & \text{if } \mathcal{T}[i-1] = 1, \mathcal{T}[i] = 0; \\ s_i & \text{if } \mathcal{T}[i-1] = 1, \mathcal{T}[i] = 1; \\ s_i - f_s & \text{if } \mathcal{T}[i-1] = 0, \mathcal{T}[i] = 1. \end{cases}$$
(1)

where f_b is the transactions cost incurred in terms of return for switching positions from stock to bond, and f_s is the transactions cost incurred for switching positions from bond to stock. We assume that these transactions costs are constants. In words, the return over time

period $[t_{i-1}, t_i]$ is the return for the instrument you end the period in minus a transactions cost if you started the period in the other instrument.

Exercise 2.2

The equity curve for a trading strategy \mathcal{T} is the vector $\mathcal{E}_{\mathcal{T}}$, where $\mathcal{E}_{\mathcal{T}}[i]$ is the value at time t_i , with $\mathcal{E}_{\mathcal{T}}[0] = 1$. The return sequence $r_{\mathcal{T}}$ is then $r_{\mathcal{T}}[i] = \log \frac{\mathcal{E}_{\mathcal{T}}[i]}{\mathcal{E}_{\mathcal{T}}[i-1]}$, for $i \geq 1$. Suppose that the bid-ask spread for bond is a fraction \hat{f}_b of the bid price, and for the stock is a fraction \hat{f}_s of the bid price.

Show that $f_s = \log(1 + \hat{f}_s)$ and $f_b = \log(1 + \hat{f}_b)$.

Note that when the bid ask spread is a constant, not a fraction of the bid price, then it is more convenient to work in the value (as opposed to the returns) space. The total return for a strategy is

$$\mu(\mathcal{T}) = \sum_{i=1}^{n} r_{\mathcal{T}}[i].$$

We will typically suppress the dependence on \mathcal{T} when it is clear what trading strategy we are referring to. We will focus on maximizing the total return, and refer the reader to the literature for the more complex problems of maximizing the Sharpe and Sterling ratios [?].

Exercise 2.3

Consider the 6 times [0, 1, 2, 3, 4, 5], over which the return sequence for bond was [1, 1, 1, 1, 1] and the return sequence for stock was [1, -2, 3, 2, 1]. Assume that $f_s = f_b = 1$ and compute the total return μ for the trading strategy $\mathcal{T} = [0, 0, 1, 0, 1, 1]$.

We now consider efficient algorithms for computing total return optimal trading strategies, with and without constraints on the number of trades ². In particular, it is possible to construct return optimal trading strategies in *linear time*:

i. Unconstrained Trading. A trading strategy \mathcal{T}^* can be computed in O(n) such that for any other strategy $\mathcal{T}, \mu(\mathcal{T}^*) \geq \mu(\mathcal{T})$.

²We will use standard O() notation in stating our results: let n be the length of the returns sequences; we say that the run time of an algorithm is O(f(n)) if, for some constant C, the runtime is $\leq Cf(n)$ for any possible return sequences. If f(n) is linear (quadratic), we say that the runtime is linear (quadratic).

ii. Constrained Trading. A trading strategy \mathcal{T}_K^* making at most K trades can be computed in $O(K \cdot n)$ time such that for any other strategy \mathcal{T}_K making at most K trades, $\mu(\mathcal{T}_K^*) \geq \mu(\mathcal{T}_K)$.

Exercise 2.4

For return optimal trading strategies, show that the all-or-nothing assumption can be made without loss of generality. In particular, show that there always exists a return optimal strategy which is all-or-nothing.

[Hint: You may want to use induction on the number of time steps n.]

In order to compute the return optimal strategies, we will use a dynamic programming approach to solve a more general problem. Specifically, we will construct the return optimal strategies for every prefix of the returns sequence. First we consider the case when there is no restriction on the number of trades, and then the case when the number of trades is constrained to be at most K.

2.2 Overview of the Algorithm

The basic idea of the algorithm is to consider the optimal strategy to time t_i . This strategy must end in either stock or bond. Suppose that it ends in stock, then it must arrive at the final position in stock at t_i by either passing through stock or bond at time t_{i-1} . Thus, the optimal strategy which ends in stock at time t_i must be either the optimal strategy which passes through stock at time t_{i-1} followed by holding the stock for one more time period, or the optimal strategy which passes through bond at time t_{i-1} and then makes a trade into the stock for the next time period. Whichever is better among these two options yields the optimal strategy to time period t_i that ends in stock. A similar argument applies to the optimal strategy to time t_i that ends in bond. Thus, having computed the optimal strategies which end in stock and bond to time t_{i-1} , we can compute the optimal strategies which end in stock and bond to time t_i . This induction can be propagated to obtain the final result.

We will illustrate this idea with the example in Exercise 2.1. First we consider the optimal strategies up to time 1, ending in stock and bond. In fact there is only one such strategy which ends in bond (namely to hold bond) and one such strategy to end in stock (namely to switch to stock), hence these are the optimal ones. We can compute the returns of these two strategies, summarized in the following picture,



We now consider the optimal strategies to time 2, ending in stock. There are two options, either you came from bond or from stock, in either case, gettinig to the previous point optimally,



Since both of these options have the same return, we may pick either as the optimal strategy to time 2 ending in stock. Similarly, we can consider the two options for the optimal strategy to time 2 ending in bond,



Since the option passing through bond at time 1 has higher return, this is the optimal strategy to time 2 ending in bond,



Continuing, we consider the two options for the optimal strategy to time 3, ending in stock,



Since the option coming through the optimal strategy to time 2 ending in bond has higher return we have the optimal strategy ending in stock at time 3 is,



Exercise 2.5

Continue the analysis of the example to obtain the optimal strategies ending in stock and bond at time 5, pictorially representing the solutions as above. Give a return optimal strategy ending in bond, and what is its return?

[Answer:



When there are constraints on the number of trades, we only need to slightly modify the above argument. We would like to compute the optimal strategy which ends (say) in stock and makes at most K trades. Any such strategy has to be one of two possibilities: it makes at most K trades ending in stock at time t_{i-1} , or it makes at most K - 1 trades, ending in bond at time t_{i-1} . If it ended in bond, it can only make at most K - 1 trades because one additional trade will be required to convert from bond at t_{i-1} to stock at t_i . Thus the inductive construction will start with K = 0 which is to hold bond. Assuming we have computed all the optimal strategies for K = k to all times $\{t_i\}$, we can then compute all the optimal strategies for K = k + 1 to all times.

2.3 Unconstrained Return-Optimal Trading Strategies

First we give the main definitions that we will need in the dynamic programming algorithm to compute the optimal strategy. Consider a return-optimal strategy for the first m + 1times $\{t_0, \ldots, t_m\}$. Define $\mathcal{S}(m, 0)$ (resp. $\mathcal{S}(m, 1)$) to be a return-optimal strategy ending in bond (resp. stock) at time t_m . Up to time t_1 , there is only one strategy ending in bond, and one strategy ending in stock, so $\mathcal{S}(1,0) = [0,0]$ and $\mathcal{S}(1,1) = [0,1]$ For $\ell \in \{0,1\}$, let $\mu(m,\ell)$ denote the return of $\mathcal{S}(m,\ell)$, i.e., $\mu(m,\ell) = \mu(\mathcal{S}(m,\ell))$. Let $\text{PREV}(m,\ell)$ denote the penultimate position of the optimal strategy $\mathcal{S}(m,\ell)$ at time t_{m-1} . Note that PREV(1,1) =PREV(1,0) = 0, since both optimal strategies to time t_1 started in bond.

We are after S(n, 0), the optimal strategy to time t_n ending in bond. Denote this strategy by \mathcal{T}^* . If we know $\operatorname{PREV}(m, \ell)$ for $m \geq 1$ and $\ell \in \{0, 1\}$, then we can construct S(n, 0)in linear time as follows. First, we have the obvious fact that $\mathcal{T}^*[n] = S(n, 0)[n] = 0$. The previous position is given by $\operatorname{PREV}(n, 0)$. Suppose $\operatorname{PREV}(n, 0) = \mathcal{T}^*[n] = 0$, i.e., the previous position was 0. Then the position previous to that is exactly the previous position for the strategy S(n-1,0) which is $\operatorname{PREV}(n-1,0)$. If on the other hand, $\operatorname{PREV}(n,0) = \mathcal{T}^*[n] = 0$, i.e., the previous position was 1. Then the position previous to that is exactly the previous position for the strategy S(n-1,1) which is $\operatorname{PREV}(n-1,1)$. More generally, suppose the optimal strategy at time m is $\mathcal{T}^*[m]$. Then the previous position is exactly $\operatorname{PREV}(m, \mathcal{T}^*[m])$. We thus have the following backward recursion for \mathcal{T}^*

$$\mathcal{T}^*[n] = 0,$$

$$\mathcal{T}^*[m-1] = \operatorname{PREV}(m, \mathcal{T}^*[m]), \text{ for } 1 \le m \le n.$$

Thus, a single backward scan is all that is required to compute all the elements in \mathcal{T}^* . This backward scan is typically called the backtracking step in a dynamic programming algorithm which is typically the step that is used in constructing the solution in a dynamic programming approach. Note that storing the entire PREV array requires memory that is linear in n. The remainder of the discussion focusses on the computation of the array PREV (m, ℓ) for $m \ge 1$ and $\ell \in \{0, 1\}$.

The optimal strategy $\mathcal{S}(m, \ell)$ must pass through either bond or stock at time t_{m-1} . Thus, $\mathcal{S}(m, \ell)$ must be the extension of one of the optimal strategies $\{\mathcal{S}(m-1, 0), \mathcal{S}(m-1, 1)\}$ by adding the position ℓ at time period t_m . More specifically,

$$S(m, \ell) = \begin{cases} [S(m-1, 0), \ell] & \text{or,} \\ [S(m-1, 1), \ell]. \end{cases}$$

In particular, $\mathcal{S}(m,\ell)$ will be the extension that yields the greatest total return. Thus,

$$\mu(m,\ell) = \max\{\mu([S(m-1,0),\ell]), \mu([S(m-1,1),\ell])\}$$

Using (1), we have that

$$\mu([\mathcal{S}(m-1,0),\ell]) = \begin{cases} \mu(m-1,0) + b_m & \ell = 0, \\ \mu(m-1,0) + s_m - f_s & \ell = 1; \end{cases}$$
$$\mu([\mathcal{S}(m-1,1),\ell]) = \begin{cases} \mu(m-1,1) + b_m - f_b & \ell = 0, \\ \mu(m-1,1) + s_m & \ell = 1. \end{cases}$$

Using these expressions, we can compute $\mu(m, \ell)$ for $m \ge 1$ and $\ell \in \{0, 1\}$ using the following recursion,

$$\mu(m,\ell) = \begin{cases} \max\{\mu(m-1,0) + b_m, \mu(m-1,1) + b_m - f_b\} & \ell = 0, \\ \max\{\mu(m-1,0) + s_m - f_s, \mu(m-1,1) + s_m\} & \ell = 1. \end{cases}$$

Simultaneously, as we compute $\mu(m, \ell)$, we can also compute $\text{PREV}(m, \ell)$ as follows,

$$PREV(m,0) = \begin{cases} 0 & \text{if } \mu(m-1,0) + b_m \ge \mu(m-1,1) + b_m - f_b, \\ 1 & \text{otherwise;} \end{cases}$$
$$PREV(m,1) = \begin{cases} 0 & \text{if } \mu(m-1,0) + s_m - f_s \ge \mu(m-1,1) + s_m, \\ 1 & \text{otherwise.} \end{cases}$$

It should be evident that if we already know $\mu(m-1,0)$ and $\mu(m-1,1)$, then we can compute $\mu(m,\ell)$ and $\text{PREV}(m,\ell)$ for $\ell \in \{0,1\}$ in constant time. Further, we have that $\mu(1,0) = b_1$ and $\mu(1,1) = s_1 - f_s$, and so, by a straight forward induction, we can compute $\mu(m,\ell)$ and $\text{PREV}(m,\ell)$ in linear time.

Exercise 2.6

Implement this dynamics programming algorithm as a function which takes as input two return series and the corresponding transactions costs, and outputs an optimal strategy, together with the return of the optimal strategy.

Is it possible to have more than one optimal strategy? If so, what can you say about the returns of the optimal strategies?

The generalization of this algorithm to N > 2 instruments is straightforward by suitably generalizing a trading strategy. $S(m, \ell)$ retains its definition, except now $\ell \in \{0, \ldots, N-1\}$. To compute $\mu(m, \ell)$ will need to take a maximum over N terms depending on $\mu(m-1, \ell')$, and so the algirithm will have runtime O(Nn).

One of the assumptions we maintained was the all or nothing assumption. The next exercise shows that we did not lose any generality in doing so.

Exercise 2.7

Show that there always exists an all or nothing trading strategy which is return optimal. in particular, show that for any trading strategy which makes K trades, there is a trading strategy which makes at most K trades with at least as much return. (This also shows that the all or nothing assumption is also not a serious restriction to constrained return optimal trading.)

[Hint: You may want to use induction on n.]

One concern with the unconstrained optimal strategy is that it may make too many trades. It is thus useful to compute the optimal strategy that makes at most a given number of trades. We discuss this next.

2.4 Constrained Return-Optimal Strategies

We now suppose that the number of trades is constrained to be at most K. The general approach is similar to the unconstrained case. It is more convenient to consider the number

of position switches a strategy makes, which we define as the number of times the position switches. For a valid trading strategy, the number of trades entered equals the number of trades exited, so k = 2K. Analogous to $\mathcal{S}(m, \ell)$ in the previous section, we define $\mathcal{S}(m, k, \ell)$ to be the optimal trading strategy to time t_m that makes at most k position switches ending with position ℓ . Let $\mu(m, k, \ell)$ be the return of strategy $\mathcal{S}(m, k, \ell)$, and let PREV(m, k, l)store the pair (k', ℓ') , where ℓ' is the penultimate position of $\mathcal{S}(m, k, \ell)$ at t_{m-1} that leads to the end position ℓ , and k' is the number of position switches made by the optimal strategy to time period t_{m-1} that was extended to $\mathcal{S}(m, k, \ell)$.

Exercise 2.8

How many possible trading strategies are there with k position switches?

The algorithm once again follows from the observation that the the optimal strategy $S(m, k, \ell)$ must pass through either bond or stock at t_{m-1} . A complication is that if the penultimate position is bond and $\ell = 0$, then at most k position switches can be used to get to thhe penultimate position, however, if $\ell = 1$, then only at most k - 1 position switches may be used. Similar reasoning applies if the penultimate position is stock. We thus get the following recursion,

$$\mu(m, k, 0) = \max \{ \mu(m-1, k, 0), \ \mu(m-1, k-1, 1) - f_b \}, \mu(m, k, 1) = \max \{ \mu(m-1, k-1, 0) + s_m - f_s, \ \mu(m-1, k, 1) + s_m \}.$$

This recursion is initialized with $\mu(m, 0, 0) = \sum_{i=1}^{m} b_i$ and $\mu(m, 0, 1) = -\infty$ for $1 \le m \le n$. Once $\mu(m, k, \ell)$ is computed for all m, ℓ , then the above recursion allows us to compute $\mu(m, k+1, \ell)$ for all m, ℓ . Thus, the computation of $\mu(m, k, \ell)$ for $1 \le m \le n, 0 \le k \le 2K$ and $\ell \in \{0, 1\}$ can be accomplished in O(nK) time. As in the unconstrained case, the strategy that was extended gives $\text{PREV}(m, k, \ell)$,

$$PREV(m, k, 0) = \begin{cases} (k, 0) & \text{if } \mu(m - 1, k, 0) > \mu(m - 1, k - 1, 1) - f_b, \\ (k - 1, 1) & \text{otherwise.} \end{cases}$$

$$PREV(m, k, 1) = \begin{cases} (k - 1, 0) & \text{if } \mu(m - 1, k - 1, 0) + s_m - f_s > \mu(m - 1, k, 1) + s_m, \\ (k, 1) & \text{otherwise.} \end{cases}$$

Thus, $\operatorname{PREV}(m, k, \ell)$ can be computed as we compute $\mu(m, k, \ell)$ in O(nK) time.

The optimal trading strategy T_K^* making at most K trades is then given by $\mathcal{S}(n, 2K, 0)$, and the full strategy can be reconstructed in a single backward scan using the following backward recursion (we introduce an auxilliary vector κ),

$$\begin{aligned} \mathcal{T}_{K}^{*}[n] &= 0, \\ \kappa[n] &= 2K \\ (\kappa[m-1], \mathcal{T}_{K}^{*}[m-1]) &= \operatorname{PREV}(m, \kappa[m], \mathcal{T}_{K}^{*}[m]), \text{ for } 1 \leq m \leq n. \end{aligned}$$

Since the algorithm needs to store $PREV(m, k, \ell)$ for all m, k, the memory requirement is O(nK). Once again, it is not hard to generalize this algorithm to work with N instruments, and the resulting run time will be O(nNK).

Exercise 2.9

Implement the dynamic programming algorithm as a function which takes as input the return sequences, the transactions costs and the maximum number of trades and returns the optimal trading strategy together with its return.

2.5 Other Work on Optimal Trading

The body of literature on optimal trading is so enormous that we only highlight here some representative papers. The reasearch on optimal trading falls into two broad categories. The first group is on the more theoretical side where researchers assume that instrument prices satisfy some particular model, for example the prices are driven by a stochastic process of known form; the goal is to derive closed-form solutions for the optimal trading strategy, or a set of equations that the optimal strategy must follow. The main drawbacks of such theoretical approaches is that their prescriptions can only be useful to the extent that the assumed models are correct. Our work does not make any assumptions about the price dynamics to construct ex-post optimal trading strategies.

The second group of research which is more on the practical side is focused on exploring data driven / learning methods for the prediction of future stock prices moves and trading opportunities. Intelligent agents are designed by training on past data and their performance is compared with some benchmark strategies. Our results furnish (i) optimal strategies on which to train intelligent agents and (ii) benchmarks with which to compare their performance.

Theoretical Approaches Boyd et al. in [?] consider the problem of single-period portfolio optimization. They consider the maximization of the expected return subject to different types of constraints on the portfolio (margin, diversification, budget constraints and limits

on variance or shortfall risk). Under certain assumptions on the returns distribution, they reduce the problem to numerical convex optimization. Similarly, Thompson in [?] considered the problem of maximizing the (expected) total cumulative return of a trading strategy under the assumption that the asset price satisfies a stochastic differential equation of the form $dS_t = dB_t + h(X_t)dt$, where B_t is a Brownian motion, h is a known function and X_i is a Markov Chain independent of the Brownian motion. In this work, he assumes fixed transaction costs and imposes assumptions A1, A2, A4 on the trading. He also imposes a stricter version of our assumption A3: at any time, the trader can have only 0 or 1 unit of stock. He proves that the optimal trading strategy is the solution of a free-boundary problem, gives explicit solutions for several functions h and provides bounds on the transaction cost above which it is optimal never to buy the asset at all.

Pliska et al. in [?] and Bielecki et al. in [?] considered the problems of optimal investment with stochastic interest rates in simple economies of bonds and a single stock. They characterize the optimal trading strategy in terms of a nonlinear quasi-variational inequality and develop a numerical approaches to solving these equations.

Some work has been done within risk-return frameworks. Berkelaar and Kouwenberg in [?] considered asset allocation in a return versus downside-risk framework, with closedform solutions for asset prices following geometric Brownian motions and constant interest rates. Liu in [?] consider the optimal investment policy of a constant absolute risk aversion (CARA) investor who faces fixed and proportional transaction costs when trading multiple uncorrelated risky assets.

Zakamouline in [?] studies the optimal portfolio selection problem using Markov chain approximation for a constant relative risk averse investor who faces fixed and proportional transaction costs and maximizes expected utility of the investor's end-of-period wealth. He identifies three disjoint regions (Buy, Sell and No-Transaction) to describe the optimal strategy.

Choi and Liu in [?] considered trading tasks faced by an autonomous trading agent. An autonomous trading agent works as follows. First, it observes the state of the environment. According to the environment state, the agent responds with an action, which in turn influences the current environment state. In the next time step, the agent receives a feedback (reward or penalty) from the environment and then perceives the next environment state. The optimal trading strategy for the agent was constructed in terms of the agent's expected utility (expected accumulated reward).

Cuoco et al. in [?] considered Value at Risk as a tool to measure and control the risk of the trading portfolio. The problem of a dynamically consistent optimal porfolio choice subject to the Value at Risk limits was formulated and they proved that the risk exposure of a trader subject to a Value at Risk limit is always lower than that of an unconstrained trader and that the probability of extreme losses is also decreased.

Mihatsch and Neuneier in [?] considered problem of optimization of a risk-sensitive expected return of a Markov Decision Problem. Based on an extended set of optimality equations, risk-sensitive versions of various well-known reinforcement learning algorithms were formulated and they showed that these algorithms converge with probability one under reasonable conditions.

Data Driven Approaches Moody and Saffell in [?] presented methods for optimizing portfolios, asset allocations and trading systems based on a direct reinforcement approach, which views optimal trading as a stochastic control problem. They developed reccurent reinforcement learning to optimize risk-adjusted investment returns like the Sterling Ratio or Sharpe Ratio, while accounting for the effects of transaction costs.

Liu et al. in [?] proposed a learning-based trading strategy for portfolio management, which aims at maximizing the Sharpe Ratio by actively reallocating wealth among assets. The trading decision is formulated as a non-linear function of the latest realized asset returns, and the function cam be approximated by a neural network. In order to train the neural network, one requires a Sharpe-Optimal trading strategy to provide the supervised learning method with target values. In this work they used heuristic methods to obtain a locally Sharp-optimal trading strategy. The transaction cost was not taken into consideration. Our methods can be considerably useful in the determination of target trading strategies for such approaches.

3 Optimal Trade Entry - The Deterministic Case

We will now consider another important application of dynamic programming in constructing the optimal way to enter a trade (short or long). We will focus on selling shares in a stock (for example) but the same general approach applies equally well to buying.

The general formulation of the problem is that you have a (usually large) number of shares, K which you would like to sell and the entire trade must be executed over the next n time steps t = 1, 2, ..., n. Two things complicate the process. The first is that typically you would like to sell because you have some market view that the price will be dropping, and so you would like to sell as fast as possible. The second is that since you are selling a large amount of stock, you will likely have *market impact*, which means that as you sell the stock, the price of the stock will change, and in general you will also affect the future market view. Typically as you sell larger quantities, you have a larger impact. Since you are selling, the impact will be negative, i.e., you will lower the future price, and the more you sell, the more the price will be lowered. This market impact encourages spreading out your trade. These two competing effects created by the market view and the market impact leads to significant profit saving if you optimally enter the trade versus not. We formalize the problem in a general way before considering simplifications which we can efficiently solve using dynamic programming approaches.

The number of shares to be sold is K over the times t = 1, 2, ..., n. Suppose that we sell k_i shares at time i, so the exit strategy can be represented by the *n*-dimensional vector $\mathbf{k} = [k_1, k_2, \cdots, k_n]$, with $\sum_i k_i = K$.

Exercise 3.1

Given K and n, how many possible exit strategies are there?

Before any shares are sold, you have some view as to how the market will behave. Specifically, the no-market impact price p_i at time *i* is known, which can be summarized in the no-market impact price vector $\mathbf{p} = [p_1, p_2, \dots, p_n]$. If the trade is to sell *K* shares, then typically the p_i 's are decreasing (one sells if one believes that the market is going down). If you sell according to the strategy \mathbf{k} , the prices will change, in particular drop, both as you sell and in the future. We will make some simplifying assumptions as to how this happens.

At time *i*, suppose that the price is p_i . If you sell *k* shares at time *i*, assume that you will execute your *k* shares at an average price of $p_i - g(k)$, where g(k) is the *execution impact* of selling the *k* shares. There will also be a *future price impact* due to this sale. In particular, all your future realizations of the price will drop by an amount f(k). Since the price is typically dropping during the execution, the average price for the execution will be higher than the final price after the execution, thus in a practical setting, one usually has that $f(k) \ge g(k)$.

At time *i* for exit strategy \mathbf{k} , let q_i be the amount by which the price has already dropped,

$$q_i = \sum_{j=1}^{i-1} f(k_j).$$

The average price for the sale of k_i shares at time *i* is then $p_i - q_i - g(k_i)$. Let c_i be the proceeds from this sale. Then,

$$c_i = k_i(p_i - q_i - g(k_i)).$$

We can thus compute the proceeds from the entire sale,

$$C(\mathbf{k}) = \sum_{i=1}^{n} c_i,$$

= $\sum_{i=1}^{n} k_i (p_i - q_i - g(k_i)),$
= $\sum_{i=1}^{n} k_i (p_i - g(k_i)) - \sum_{i=1}^{n} \sum_{j=1}^{i-1} k_i f(k_j)$

The functions g, f are specified as vectors: $\mathbf{g} = [g_0, g_1, \dots, g_K]$ and $\mathbf{f} = [f_0, f_1, \dots, f_K]$. Note that $g_0 = f_0 = 0$. Given $\mathbf{p}, \mathbf{g}, \mathbf{f}$ and K, the task is to maximize C over all strategies $\mathbf{k} \ge 0$ such that $\sum_i k_i = K$. We can assume that \mathbf{k} is a non-negative integer vector, because one can only execute an integral number of shares at a time.

Exercise 3.2

Suppose that the price vector is a constant vector equal to 100. Assume that K = 10 and that f(k) = g(k) = 0 if $0 \le k \le 1$ and 1 otherwise.

- (a) Compute C, q_i for the strategy $\mathbf{k} = [4, 3, 2, 1, 0, 0, 0, 0, 0, 0]$.
- (b) What is the optimal exit strategy and corresponding to it, what is C.

We now develop a dynamic programming solution for obtaining the optimal exit strategy. Suppose that you are at time i and the price has already dropped by an amount q and you have k shares remaining to sell. Let $C^*(k, q, i)$ denote the maximum possible proceeds from optimally executing the remainder of the trade (k shares) starting at time i.

We would like to know for starters, what is $C^*(K, 0, 1)$, the maximum possible proceeds from the sale of the K shares, in addition to the exit strategy to obtain that maximum. We begin by observing that at time n, there is nothing to be done but sell all the remaining k shares no matter what the price drop has been, so,

$$C^*(k, q, n) = k(p_n - q - g(k)).$$

Now consider $C^*(k, q, i)$ for a time i < n. Of the k shares remaining to be sold, there are only k + 1 possibilities, corresponding to selling $0, 1, \ldots, k$ shares at time i. After selling $0 \le \ell \le k$ shares at time i, the maximum amount of money which can be made is

$$\ell(p_i - q - g(\ell)) + C^*(k - \ell, q + f(\ell), i + 1).$$

To obtain $C^*(k, q, i)$, which is the *maximum* amount of money that can be made at time *i*, we should take the maximum over all possible choices of ℓ to obtain,

$$C^*(k,q,i) = \max_{0 \le \ell \le k} \left\{ \ell(p_i - q - g(\ell)) + C^*(k - \ell, q + f(\ell), i + 1) \right\}.$$
 (2)

The value of ℓ which attains this maximum will also be useful for us in reconstructing the optimal strategy through the usual process of backtracking in a dynamic program. Let $\ell^*(k, q, i)$ be this value of ℓ ,

$$\ell^*(k,q,i) = \operatorname*{argmax}_{0 \le \ell \le k} \left\{ \ell(p_i - q - g(\ell)) + C^*(k - \ell, q + f(\ell), i + 1) \right\}.$$
 (3)

This backward induction allows us to compute C^* , ℓ^* at time *i* for all k, q if we have already computed C^* , ℓ^* at time i + 1 for all k, q. Since we know C^* at time *n* for all k, q, we can initiate the process at time *n* and continue all the way back to time 1, where we need $C^*(K, 0, 1)$. Note that $\ell^*(k, q, n) = k$ since there is nothing more to do than sell off all the remaining shares.

To construct an optimal strategy, we have to use a forward induction with ℓ^* . Clearly $k_1 = \ell^*(K, 0, 1)$. Let κ_i be the number of shares remaining to execute at time i, $\kappa_1 = K$. Note that the price drop at time 1 is 0, $q_1 = 0$. In general, if there are κ_i shares to execute at time i and the price has dropped by q_i , then the optimal strategy sells $k_i = \ell^*(\kappa_i, q_i, i)$ shares, and we update $\kappa_{i+1} = \kappa_i - k_i$ and $q_{i+1} = q_i + f(k_i)$. Summarizing, we have that $\kappa_1 = K, q_1 = 0$ and for $i \ge 1$,

$$k_i = \ell^*(\kappa_i, q_i, i),$$

$$\kappa_{i+1} = \kappa_i - k_i,$$

$$q_{i+1} = q_i + f(k_i).$$

This forward induction allows us to compute k_i for $i \ge 1$.

Exercise 3.3

We will explore the maximum possible market impact that one can have when executing the trade. The total market impact is $q = \sum_i f(k_i)$, the total amount by which the market was moved. We would like to compute the maximum possible value of q under the restriction that $\sum_i k_i = K$. Thus define

$$q^*(K) = \max_{\sum_i k_i = K} \sum_i f(k_i).$$

Give a dynamic programming algorithm to compute $q^*(K)$.

[Hint: Show that $q^*(K) = \max_{1 \le k \le K} \{f(k) + q^*(K-k), \text{ with } q^*(0) = 0.\}$

3.1 Computational Considerations

It turns out that everything we have said is correct, but for the particular model we are considering, we can get a very efficient algorithm for computing the optimal strategy by looking a little more closely at C^* .

Exercise 3.4

Show that

$$C^*(k,q,i) = C^*(k,0,i) - kq$$

[Hint: You may want to consider proof by induction.]

Exercise 3.5

Using the previous exercise or otherwise show that $\ell^*(k,q,i)$ is independent of q, i.e. $\ell^*(k,q,i) = \ell^*(k,0,i)$.

The previous two exercises show that we can rewrite (2) and (3) because $C^*(k,q,i) = C^*(k,i) - kq$, where $C^*(k,i) = C^*(k,0,i)$:

$$C^{*}(k,q,i) = C^{*}(k,i) - kq,$$

$$C^{*}(k,i) = \max_{0 \le \ell \le k} \{ \ell p_{i} - \ell g(\ell) - (k-\ell)f(\ell) + C^{*}(k-\ell,i+1) \},$$

$$\ell^{*}(k,i) = \operatorname{argmax}_{0 \le \ell \le k} \{ \ell p_{i} - \ell g(\ell) - (k-\ell)f(\ell) + C^{*}(k-\ell,i+1) \}.$$

The boundary condition for \mathcal{C}^* is $\mathcal{C}^*(k,n) = k(p_n - g(k))$. We reconstruct the optimal strategy from ℓ^* . In particular, $\kappa_1 = K$ and for $i \geq 1$,

$$k_i = \ell^*(\kappa_i, i),$$

$$\kappa_{i+1} = \kappa_i - k_i,$$

The algorithm needs to store $C^*(k, i)$ for $1 \le k \le K$ and $1 \le i \le n$, which is O(nK)memory, similarly for ℓ^* . In terms of computation, at each time step i, K numbers need to be computed, $C^*(k, i)$ for $1 \le k \le K$, and each computation requires taking a maximum over $k \le K$ numbers which is $O(K^2)$ computation at each time step, resulting in a total computation of $O(nK^2)$ computation.

Exercise 3.6

Consider the following exit scenario. You wish to sell 10 shares (K=10) of a stock by time T = 10. Assume the stock price is decreasing linearly, $P_t = 100 - \alpha t$, where α is a parameter we will play with. Assume that the impact function is linear, $f(x) = \beta x$ where β is also a parameter to we will play with. You can only sell an integral number of shares at a time.

K

- (a) If you were to do brute force search for the optimal exit strategy, how many possible exit strategies are there?
- (b) Implement efficiently the dynamic programming algorithm to compute the optimal exit and determine the optimal exit strategy together with the maximum proceeds from the sale when $\alpha = \{0, 1, 2\}$, with $\beta = 1$. Repeat with $\beta = 2$.
- (c) Explain intuitively what is going on.

3.2 Market Impact, Execution Impact and the Order Book

The discussion so far has been for arbitrary price and execution impact functions. One choice is for the price and execution impact functions to be the same, $\mathbf{f} = \mathbf{g}$. This is not completely realistic though reasonable. The price and market impact functions are related through the order book and its dynamics. In particular, since we have been focussing on the sale of K shares, we should look at the bid side order book.

Lets postulate a very simple model for the order book dynamics. In particular, suppose that the zero impact market view \mathbf{p} gives the top level of the bid side order book, i.e., the highest price at which someone is willing to by.

Lets assume that the order book has an equilibrium state which it can restore over the course of one time step. The order book state is the precise description of the orders which have been placed on the bid stack, which is a function F that specifies the number of orders placed at a particular prices at or below the bid price. In particular, let p be the bid price (top level of the order book). Let δ be the tick size, the minimum possible difference between prices (for example $\delta = 1$ cent). The function F(i) (for $\delta \ge 0$) which specifies the bid stack state is the number of bid orders with price at $p - i\delta$. Thus,

$$F(i)$$
 = number of bid orders with price $p - i\delta$.

So, for example, F(0) is the number of orders placed at the bid. Typically (in the equilibrium state) the number of orders placed gets smaller (the bid stack gets thinner) as you move further away from the bid stack.

The market impact is of placing an order of size k is the determined by removing the k orders with highest prices and the resulting price is at the top level of the bid stack is the price after market impact. Thus, for example, if k = F(0), then $f(k) = \delta$. Define the cumulative sequence $G(i) = \sum_{j=0}^{i} F(j)$. Then we obtain the market impact function as

$$f(k) = \begin{cases} 0 & 0 \le k < G(0), \\ \delta & G(0) \le k < G(1), \\ 2\delta & G(1) \le k < G(2), \\ \vdots \\ i\delta & G(i-1) \le k < G(i) \\ \vdots \end{cases}$$

Note that if $k > \sum_{i=0}^{\infty} F(i)$, then $f(k) = \infty$. We can now compute the average execution price for an order of size k, and hence the execution impact function g(k) using the following logic. The first F(0) shares will be sold at price p. The next F(1) shares will be sold at price $p - \delta$. The next F(2) shares will be sold at price $p - 2\delta$, and so on.

Exercise 3.7

Show that the execution impact function is given by

$$g(k) = \begin{cases} 0 & 0 \le k \le G(0), \\ \frac{\delta}{k} \left[\sum_{i=0}^{i-1} iF(i) + i(k - \sum_{i=0}^{i-1} F(i)) \right] & G(i-1) < k \le G(i). \end{cases}$$

Typically, F(i) is a non-increasing function. Some useful examples are the uniform order book, where $F(i) = \beta$; the linear order book, $F(i) = \lceil \max(1, \beta - \gamma i) \rceil$; polynomial decay, $F(i) = \lceil \max(1, \beta/(1+i)^{\rho}) \rceil$; exponential decay, $F(i) = \lceil \max(1, \beta e^{-\rho i}) \rceil$;

Exercise 3.8

For the four types of order book state,

$$F(i) = \beta,$$

$$F(i) = \lceil \max(1, \beta - \gamma i) \rceil,$$

$$F(i) = \lceil \max(1, \frac{\beta}{(1+i)^{\gamma}}) \rceil,$$

$$F(i) = \lceil \max(1, \beta e^{-\gamma i}) \rceil,$$

compute f(k) and g(k), giving plots. In all cases, $F(0) = \beta$. [Answer: For example, when $F(i) = \beta$,

$$f(k) = \delta \left\lfloor \frac{k}{\beta} \right\rfloor,$$

$$g(k) = \delta \left[\left\lceil \frac{k}{\beta} \right\rceil - 1 - \frac{\beta}{2k} \left\lceil \frac{k}{\beta} \right\rceil \left(\left\lceil \frac{k}{\beta} \right\rceil - 1 \right) \right].$$

The other cases are more complicated.



4 Optimal Trade Entry with Stochastic Inference

So far, everything we have considered is deterministic in the sense that the market view \mathbf{p} is known and deterministic, the impact functions are known and deterministic, and hence the problem was solved as a deterministic dynamic program. The two main issues are that the entire setup was *deterministic* and *known*. What happens when we try to relax some of these restrictions. In particular, when the setup is not deterministic, we have to deal with uncertainties in the outcomes, and so we should maximize the *expected* wealth returned by the exit strategy. The exit strategy may also be dynamically changing in the sense that if for example prices are not independently distributed, after a realization of one price, it may alter your market view for the remaining prices, and hence alter your exit strategy. When things are not known, one has to then make inferences based on the incremental realization of the process of exit. The first things we need to examine are where we need to relax the assumptions.

First consider the sale of k_1 shares at time 1. Certainly, by looking at the order book we can more or less deterministically compute q(k) if we assume there is no change in the order book between the time you place your sale and the time it gets executed. So it is reasonable to assume that the function q(k) is known deterministically. The assumption that the equilibrium state of the order book is more or less stable (after the order book refills) is also not such a drastic assumption, and so we can assume that q(k) is a known, deterministic, time invariant function. A similar argument can be made for the instantaneous price impact after the sale is made, i.e. the instanteneous price impact will be deterministically f(k). How this instantaneous price impact will affect the distribution of future prices however may not be the deterministic model of lowering all of them by the same constant f(k). For example, if the market view had originally prices rising, then according to the deterministic uniform impact of lowering all prices in the future, the prices will still be rising. On the other hand, it is an observed phenomenon in the real markets that a large trade may infact shift the sentiment of the market from one of rising prices to one of dropping prices, in which case the impact of the sale is in some sense much more drastic. However, it is reasonable to assume that how the market is affected is known, though the impact may not be deterministic.

As for the price time series, however, this is clearly not deterministically known. Certainly we may have distributions for the prices in the future, but this is also a far reaching assumption. In general, we do not even know the distribution for the prices in the future. More specifically, we may conjecture that the prices follow some parameterized stochastic process, and the parameters of this stochastic process are not known. One may have some prior beliefs about these parameters, but essentially, these parameters must be *inferred* simultaneously with trying to optimally execute the trade.

4.1 A Stochastic Process for Trade Execution

The first thing we will do is see what happens when we introduce a known stochastic process for the price time series. Thus, instead of specifying the price vector \mathbf{p} deterministically

ahead of time lets specify instead a stochastic process \mathbf{S} by which prices are generated as a trade is executed. We will assume that the execution impact function and the price impact function are given. In particular, we will assume that the execution impact is a percentage of the price, q = q(k)p and similarly for the price impact, f = f(k)p, where $0 \le f(k), q(k), \le 1$. This assumption, in addition to be technically more convenient also guarantees that all prices (execution or final) are positive. Suppose that the price at time t_1 is p_1 , known deterministically, and suppose that we have an execution strategy $\mathbf{k} = [k_1, \ldots, k_n]$. The stochastic process for trade execution will be based on a Geometric Brownian Motion for the instrument price. Specifically, the price at the next time step has a log-normally distributed value around the price at the current time step. Specifically, let's define the random vector of future prices $\tilde{\mathbf{p}} = [\tilde{p}_1, \dots, \tilde{p}_n]$, where $\tilde{p}_1 = p_1$. The instantaneous price after execution of k_i shares at time i is denoted by \tilde{q}_i , again a random variable³. Via the market impact, $\tilde{q}_i = \tilde{p}_i - f(k_i)$. The proceeds of the sale at time *i* are $\tilde{c}_i = k_i(\tilde{p}_i - g(k_i))$. The final piece of the stochastic process is to specify how the price at the next time step, i + 1 is determined from \tilde{q}_i , the price after market impact at time *i*. As already mentioned, we will use a Geometric Brownian motion, and so we have that

$$\log \tilde{p}_{i+1} = \log \tilde{q}_i + \eta,$$

where η is a normal random variable with mean μ and variance σ^2 . Thus, the entire stochastic process which ensues during the execution of the exit strategy **k** is as follows. $\tilde{p}_1 = p_1$, and for $i \geq 1$,

$$\tilde{q}_i = \tilde{p}_i(1 - f(k_i)),
\tilde{c}_i = k_i \tilde{p}_i(1 - g(k_i)),
\tilde{p}_{i+1} = \tilde{q}_i e^{\eta}.$$

The wealth achieved from the sale is $\tilde{C} = \sum_{i \ge 1} \tilde{c}_i$. We will assume that k(1 - g(k)) is an increasing function of k for $k \le K$, to avoid the complication that it may be better not to liquidate all shares. Note that we assume that μ and σ^2 are known and are constants. The assumption of known μ, σ^2 is a large one. The assumption that they are constant is not so bad, it basically amounts to the fact that the person selling the K shares has instantaneous market impact via the order book, but does not change the sentiment in the market place as to the general trend and volatility. Later we will see what happens if we relax both of these assumptions. This model, with linear choice for $g(\ell)$ was studied in [?].

4.1.1 Optimal Trade Exit with Known μ, σ^2

The task is to maximize the expected wealth $E[\tilde{C}]$ after the sale of the entire k shares. The expectation is with respect to the stochastic process defined above, and ofcourse the complication is that the stochastic process itself depends on the exit strategy **k**.

 $^{^{3}}$ We will generally use the tilde to indicate a random variable, and the same variable without a tilde for a realization.

Once again, we begin by considering time step n. If there are k shares to be sold and the price is p_n , then the maximum wealth to be gained is

$$C^*(k, n, p_n) = k p_n (1 - g(k))$$

Let's now consider time n-1, with k shares to be sold and the price being p_{n-1} . At time n-1, one can sell $0, 1, 2, \ldots, k$ shares, and sell the remaining at time n. The expected return after selling ℓ shares at time n-1 is

$$\ell p_{n-1}(1-g(\ell)) + E[C^*(k-\ell, n, \tilde{p}_n)]$$

Since $\tilde{p}_n = p_{n-1}(1 - f(\ell))e^{\eta}$, we have

$$\ell p_{n-1}(1-g(\ell)) + E[(k-\ell)p_{n-1}(1-f(\ell))(1-g(k-\ell))e^{\eta}],$$

= $p_{n-1}(\ell(1-g(\ell)) + (k-\ell)(1-f(\ell))(1-g(k-\ell))e^{\mu+\frac{1}{2}\sigma^2}).$

The important thing to note is that this function is linear and homogeneous in p_{n-1} , and the coefficient, which depends on the number of shares sold is

$$\ell(1 - g(\ell)) + (k - \ell)(1 - f(\ell))(1 - g(k - \ell))e^{\mu + \frac{1}{2}\sigma^2}$$

The work in [?] focuses on the case when f(k) = 0 and $g(k) = \theta k$, i.e., no price impact and linear execution impact. Let $\ell^*(k, n-1)$ be the value of $\ell \in \{0, 1, \ldots, k\}$ which maximizes this coefficient, which is *independent of* p_{n-1} . Then,

$$\ell^*(k, n-1; \mu, \sigma^2) = \operatorname*{argmax}_{\ell \in \{0, 1, \dots, k\}} \{\ell(1 - g(\ell)) + (k - \ell)(1 - f(\ell))(1 - g(k - \ell))e^{\mu + \frac{1}{2}\sigma^2}\},\$$

where we have explicitly shown the dependence on μ, σ^2 . Corresponding to this optimal value ℓ^* we have that

$$C^*(k, n-1, p_{n-1}) = p_{n-1}w^*(k; \mu, \sigma^2),$$

where

$$w^*(k;\mu,\sigma^2) = \ell^*(1-g(\ell^*)) + (k-\ell^*)(1-f(\ell^*))(1-g(k-\ell^*))e^{\mu+\frac{1}{2}\sigma^2},$$

and $\ell^* = \ell^*(k, n-1; \mu, \sigma^2)$. Note that the maximization problem to obtain ℓ^* is an O(k) operation. So far, we see that $C^*(k, i, p_i) = p_i w^*(k, i)$ holds for i = n, n-1, with

$$w^*(k,n) = k(1-g(k)),$$

$$w^*(k,n-1) = \ell^*(1-g(\ell^*)) + (k-\ell^*)(1-f(\ell^*))(1-g(k-\ell^*))e^{\mu+\frac{1}{2}\sigma^2}.$$

Suppose that for t = i + 1, i + 2, ..., n we have that

$$C^*(k, t, p_i) = p_t w^*(k, t).$$

We now analyze the general case at time *i* and consider $C^*(k, i, p_i)$. We will suppress the dependence on μ, σ^2 . Assuming we sell ℓ shares at time *i*, the expected wealth is

$$\ell p_i(1 - g(\ell)) + E[C^*(k - \ell, i + 1, \tilde{p}_{i+1})].$$

By the induction hypothesis,

$$C^*(k-\ell, i+1, \tilde{p}_{i+1}) = \tilde{p}_{i+1}w^*(k-\ell, i+1),$$

= $p_i(1-f(\ell))e^{\eta}w^*(k-\ell, i+1).$

We can now compute the expected wealth as

$$p_i(\ell(1-g(\ell)) + w^*(k-\ell, i+1)(1-f(\ell))e^{\mu + \frac{1}{2}\sigma^2}),$$

which is linear in p_i for every possible choice of ℓ , in particular the optimal choice of ℓ . The maximum expected wealth is obtained by maximizing with respect to ℓ ,

$$\ell^*(k,i) = \operatorname*{argmax}_{\ell \in \{0,1,\dots,k\}} \{\ell(1-g(\ell)) + w^*(k-\ell,i+1)(1-f(\ell))e^{\mu + \frac{1}{2}\sigma^2}.\}$$

one of the important things to note is that the optimal trading strategy ℓ^* is independent of p_i , it only depends on k, and hence it is independent of the entire price dynamics. As a result, the optimal trading strategy can be computed *ahead of time*. This is not necessarily the case for other possible stochastic price dynamics. We can write the maximum expected wealth as

$$C^*(k, i, p_i) = p_i w^*(k, i),$$

where

$$w^*(k,i) = \ell^*(1 - g(\ell^*)) + w^*(k - \ell^*, i + 1)(1 - f(\ell^*))e^{\mu + \frac{1}{2}\sigma^2},$$

and $\ell^* = \ell^*(k, i; \mu, \sigma^2)$. We are now ready to summarize this entire discussion into the full dynamic programming algorithm to compute the optimal exit strategy. First, note that the optimal exit strategy can be computed from $\ell_i^*(k)$ as follows. Let $\kappa_1 = K$ and for $i \ge 1$,

$$k_i = \ell^*(\kappa_i, i),$$

$$\kappa_{i+1} = \kappa_i - k_i,$$

The optimal exercise functions, $\ell^*(k, i)$ and the maximum expected wealth functions $C^*(k, i, p)$ are computed simultaneously by the following backward process. We start at i = n,

$$\ell^{*}(k,n) = k, w^{*}(k,n) = k(1-g(k)), C^{*}(k,n,p) = p \cdot w^{*}(k,n).$$

Now comes the backward induction. Assume that for time i+1 we have computed $\ell^*(k, i+1)$ and $w^*(k, i+1)$ for all $k \in \{0, 1, \ldots, K\}$, where i < n. This is certainly true for i = n - 1. Then we can compute $\ell^*(k, i)$ and $w^*(k, i)$ for all $k \in \{0, 1, \ldots, K\}$ as follows,

$$\ell^{*}(k,i) = \underset{\ell \in \{0,1,\dots,k\}}{\operatorname{argmax}} \{\ell(1-g(\ell)) + w^{*}(k-\ell,i+1)(1-f(\ell))e^{\mu+\frac{1}{2}\sigma^{2}}\},\$$

$$w^{*}(k,i) = \ell^{*}(1-g(\ell^{*})) + w^{*}(k-\ell^{*},i+1)(1-f(\ell^{*}))e^{\mu+\frac{1}{2}\sigma^{2}},\$$

$$C^{*}(k,i,p) = p \cdot w^{*}(k,i),$$

where $\ell^* = \ell^*(k, i)$. The first step here, in the computation of $\ell^*(k, i)$ requires an optimization, which has to be performed for each $k \in \{0, 1, \ldots, K\}$. The maximum expected return is given by $C^*(K, 1, p_1)$, and the optimal strategy is computed starting with $\ell^*(K, 1)$.

Computational Considerations. We see that we have to store $\ell^*(k, i)$ and $w^*(k, i)$ for $k \in \{0, 1, \ldots, K\}$ and $i \in \{1, \ldots, n\}$, which is an O(nK) memory requirement. At each time step *i*, to compute $\ell^*(k, i)$ we need to take a maximum of O(k) numbers which can be computed in constant time, which has to be done for the *K* possible values of *k* at each time step, for a total computation of $O(nK^2)$. Thus, we see that this more complex case with stochastic price dynamics can be solved with similar efficiency to the deterministic case we considered earlier.

Exercise 4.1

In the discussion, to simplify the problem, we assumed the execution and price impact were proportional, i.e., the execution price was p(1 - g(k)) and the price after impact was p(1 - f(k)). We will investigate what would happen had we remained with the additive model, p - g(k) and p - f(k) for the execution price and final price respectively. Define the optimal execution function

4.1.2 Optimal Trade Exit with Uncertain μ, σ

So far, we have assumed that μ , σ^2 are known. Suppose instead that μ , σ^2 are not known, but we have some belief about what they are, i.e., we have some prior distribution for the possible values of μ , σ . The entire derivation of the previous section follows through unaffected except for the calculation of $E[e^{\eta}]$. This computation must now must now take into account the fact that μ and σ come from some distribution. Suppose that this distribution is parameterized by a parameter $\boldsymbol{\theta}$,

$$p(\mu, \sigma^2 | \boldsymbol{\theta}).$$

Then,

$$E[e^{\eta}] = E[e^{\mu + \frac{1}{2}\sigma^2}],$$

= $\int d\mu \, d\sigma^2 \, e^{\mu + \frac{1}{2}\sigma^2} p(\mu, \sigma^2 | \boldsymbol{\theta}),$
= $Q(\boldsymbol{\theta}),$

where $Q(\boldsymbol{\theta})$ is a function of the parameters which determine the joint distribution of (μ, σ^2) . A simple case is when σ , the volatility in the market place is known and μ is not known,

having a Normal distribution with mean θ and variance ρ^2 , $\mu \sim N(\theta, \rho^2)$. For this case, $Q(\theta)$ is computed in the following exercise.

Exercise 4.2

Suppose that σ^2 is known and $\mu \sim N(\theta, \rho^2)$. Show that

- (a) $\eta \sim N(\theta, \rho^2 + \sigma^2)$.
- (b) Using the previous part or otherwise, show that

$$Q(\theta, \rho^2, \sigma^2) = e^{\theta + \frac{1}{2}(\rho^2 + \sigma^2)}.$$

Analogously to the previous section, The optimal exercise functions, $\ell^*(k, i)$ and the maximum expected wealth functions $C^*(k, i, p)$ are computed simultaneously by the following backward process. We start at i = n,

$$\ell^*(k, n; \boldsymbol{\theta}) = k,$$

$$w^*(k, n; \boldsymbol{\theta}) = k(1 - g(k)),$$

$$C^*(k, n, p; \boldsymbol{\theta}) = p \cdot w^*(k, n).$$

Now comes the backward induction. Assume that for time i + 1 we have computed $\ell^*(k, i + 1; \theta)$ and $w^*(k, i + 1; \theta)$ for all $k \in \{0, 1, ..., K\}$, where i < n. This is certainly true for i = n - 1. Then we can compute $\ell^*(k, i; \theta)$ and $w^*(k, i; \theta)$ for all $k \in \{0, 1, ..., K\}$ as follows,

$$\begin{aligned} \ell^*(k,i;\theta) &= \arg \max_{\ell \in \{0,1,\dots,k\}} \{ \ell(1-g(\ell)) + w^*(k-\ell,i+1;\theta)(1-f(\ell))Q(\theta) \}, \\ w^*(k,i;\theta) &= \ell^*(1-g(\ell^*)) + w^*(k-\ell^*,i+1;\theta;\theta)(1-f(\ell^*))Q(\theta), \\ C^*(k,i,p;\theta) &= p \cdot w^*(k,i;\theta), \end{aligned}$$

where $\ell^* = \ell^*(k, i; \boldsymbol{\theta})$. The first step here, in the computation of $\ell^*(k, i; \boldsymbol{\theta})$ requires an optimization, which has to be performed for each $k \in \{0, 1, \dots, K\}$. The maximum expected return is given by $C^*(K, 1, p_1; \boldsymbol{\theta})$, and the optimal strategy is computed starting with $\ell^*(K, 1; \boldsymbol{\theta})$ as follows: Let $\kappa_1 = K$ and for $i \geq 1$,

$$k_i = \ell^*(\kappa_i, i; \boldsymbol{\theta}),$$

$$\kappa_{i+1} = \kappa_i - k_i.$$

4.2 A Stochastic Process for Trade Execution with Changing μ

We now consider what happens if the market impact is so large that it can change the market sentiment, i.e., affect the trend. We will assume that volatility remains constant.

Thus, $\mu_i = \mu_{i-1} - h(k_i)$, which indicates that the drift shifts down (which is the correct direction for selling shares), by an amount which depends on the quantity of shares sold. In most liquid markets, h(k) is close to 0 for small k and only once k gets really large does g(k) become significant. A simple case is linear response, with $h(k) = \delta k$. The full dynamics are therefore, $\tilde{p}_1 = p_1, \mu_1 = \mu$, and for $i \geq 1$,

where $N(\cdot)$ is the Normal distribution. In this model, there is an instantaneous price impact by a factor $(1 - f(k_i))$ which affects all future prices by this factor, as well as an impact on the price process itself (in this case only on the drift). This can be viewed as the sale of a big number of shares changing the sentiment in the market.

4.3 Stochastic Inference

So far we have assumed that μ is known, and constant. Suppose that μ is not known, but rather one has some prior over the possible values of μ . In this case one is faced with a tradeoff between two choices: execute now, based on the current information about μ ; or, delay the execution a little in order to get a better estimate of μ and then execute optimally later for more profit. Lets assume that at time 1, we have a Normal prior on μ , with mean θ , and variance ρ^2 . If we make a trade at time 1 of size k_1 and observe the price at the next time step, \tilde{p}_2 , then we know that

$$\log(\tilde{p}_2) - \log(q_1) \sim N(\mu, \sigma^2),$$

where $N(\cdot)$ is the Normal distribution.

Exercise 4.3

Suppose that the prior distribution of μ is a normal distribution with mean θ and variance $\rho^2,$

$$P(\mu) \sim N(\theta, \rho^2).$$

Suppose that a random variate x is drawn from a Normal distribution with mean μ and variance σ^2 ,

$$x \sim N(\mu, \sigma^2).$$

Show that the posterior distribution for μ after observing x is Normal,

$$P(\mu|x) \sim N(\theta', {\rho'}^2),$$

where

$$\theta' = \theta + \frac{\rho^2}{\rho^2 + \sigma^2} (x - \theta), \qquad {\rho'}^2 = \rho^2 \cdot \frac{1}{1 + \rho^2 / \sigma^2}$$

Notice in the previous exercise that the variance (uncertainty) in μ as measured by ρ^2 always drops by the same amount, independent of x and the expected value θ shifts in the direction of x by an amount which depends on the current uncertainty.

The dynamics for the trade exit are as follows. Initially, the price is p_1 , and we have some prior for μ , the drift of the price in the Geometric Brownian Motion, $\mu \sim \mathbb{N}(\theta_1, \rho_1^2)$. If we make a trade of k shares, we will have a temporary effect on μ , the drift, given by $\mu \to \mu - h(k)$ which affects the price at the next time step, and we will have a permanent effect on $\mu \to \mu - \delta h(k)$, where the permanent effect is generally small, $0 \leq \delta \ll 1$. The full dynamics are given by the following equations. Assume that we have an exit strategy **k**. At time 1, we have $\tilde{p}_1 = p_1, \mu_1 \sim N(\theta_1, \rho_1^2)$, where $\theta_1 = \theta$ and $\rho_1^2 = \rho^2$ where p_1, θ, ρ are inputs to the system. For $i \geq 1$, we have

$$\mu \sim N(\theta_i, \rho_i^2),$$

$$\eta_i \sim N(\mu, \sigma^2) - h(k_i), \tag{4}$$

$$\tilde{p}_{i+1} = \tilde{p}_i (1 - f(k_i)) e^{\eta_i},$$
(5)

$$\tilde{c}_{i} = k_{i}\tilde{p}_{i}(1 - g(k_{i}))e^{\eta_{i}}, \tag{6}$$

$$\theta_{i+1} = \theta_i - \delta h(k_i) + \frac{\rho_i}{\rho_i^2 + \sigma^2} \left(\log \frac{p_{i+1}}{\tilde{p}_i(1 - f(k_i))} + h(k_i) - \theta_i \right),$$
(7)

$$\rho_{i+1}^2 = \rho_i^2 \cdot \frac{1}{1 + \rho_i^2 / \sigma^2}.$$

In (4) we model the instant, temporary impact of the sale of k_i shares on the drift μ by postulating that it decreases the η_i by $h(k_i)$ for the current random variate η_i . The price evolution (5) includes both the price impact from $f(k_i)$ and the geometric evolution according to e^{η_i} . The execution price (??) reflects the instant execution impact $g(k_i)$ as well as the evolution. From the dynamics, $\log \tilde{p}_{i+1} - \log \tilde{p}_i(1 - f(k_i))$ equals η_i and so $x = \log \tilde{p}_{i+1} - \log \tilde{p}_i(1 - f(k_i)) + h(k_i)$ is a random variate from $N(\mu, \sigma^2)$, and hence this x should be used in conjunction with the result in Exercise 4.3 to update the prior on μ , which is given in (7). The mean update for the prior is as given by Exercise 4.3 with the provision for a permanent affect on the mean of lowering it by an additional $\delta h(k_i)$. Notice we have introduced a parameter δ here to account for the fact that the permanent impact may be different from the temporary impact, in general we would suppose that $0 < \delta \ll 1$. Thus, the state at any given time step i, can be summarized by the number of shares k which remain to be sold, the price p_i and the belief about μ as encoded in θ_i . Technically, ρ_i is also part of the state, but since it deterministically changes, we do not need to worry about it.

As always, lets begin the analysis with the final time step n. In this case, there is nothing

to do but sell all the shares, and so

$$\tilde{C}_n^*(k, p_n, \theta_n) = p_n k (1 - g(k)) e^{\eta_n}.$$

Taking the expectation, the maximum wealth attainable at time n is

$$C_n^*(k, p_n, \theta_n) = p_n k (1 - g(k)) E[e^{\eta_n}].$$

Exercise 4.4

Show that the distribution of η_i is a normal distribution, specifically,

$$\eta_i \sim N(\theta_i - h(k_i), \rho_i^2 + \sigma^2).$$

Hence, show that

$$E[e^{\eta_i}] = e^{\theta_i - h(k_i) + \frac{1}{2}(\rho_i^2 + \sigma^2)}.$$

Using the result of the previous exercise, we have that

$$C_n^*(k, p_n, \theta_n) = p_n e^{\theta_n} k(1 - g(k)) e^{-h(k) + \frac{1}{2}(\rho_n^2 + \sigma^2)},$$

= $p_n e^{z_n \theta_n} w_n(k),$

where $w_n(k) = k(1 - g(k))e^{-h(k) + \frac{1}{2}(\rho_n^2 + \sigma^2)}$ and $z_n = 1$.

Let's now consider time n-1. Assuming that ℓ shares are sold at time n-1, the expected maximum wealth attainable is

$$C_{n-1}^{*}(k,\ell,p_{n-1},\theta_{n-1}) = E[p_{n-1}\ell(1-g(\ell))e^{\eta_{n-1}} + C_n(k-\ell,\tilde{p}_n,\tilde{\theta}_n)],$$

and we have that

$$C_{n-1}^*(k, p_{n-1}, \theta_{n-1}) = \max_{\ell \in \{0, 1, 2, \dots, K\}} C_{n-1}^*(k, \ell, p_{n-1}, \theta_{n-1}).$$

Since $\tilde{p}_n = p_{n-1}(1 - f(\ell))e^{\eta_{n-1}}$ and $\tilde{\theta}_n = \frac{\sigma^2}{\rho_{n-1}^2 + \sigma^2}\theta_{n-1} - \delta h(\ell) + \frac{\rho_{n-1}^2}{\sigma^2 + \rho_{n-1}^2}(\eta_{n-1} + h(\ell))$, and using the expression for C_n^* , we have

$$\begin{aligned} C_{n-1}^*(k,\ell,p_{n-1},\theta_{n-1}) &= p_{n-1}\ell(1-g(\ell)) + E[p_{n-1}(1-f(\ell))(k-\ell)(1-g(k-\ell))e^{\eta_{n-1}}], \\ &= p_{n-1}\left[\ell(1-g(\ell)) + (1-f(\ell))(k-\ell)(1-g(k-\ell))e^{\frac{1}{2}(\sigma^2+\rho_{n-1}^2)-h(\ell)}e^{\theta_{n-1}}\right], \\ &= p_{n-1}\left[x_{n-1}(\ell) + y_{n-1}(k,\ell)e^{z_{n-1}\theta_{n-1}}\right], \end{aligned}$$

where $z_{n-1} = 1$, $x_{n-1}(\ell) = \ell(1 - g(\ell))$ is an increasing function of ℓ for $\ell \in \{0, \ldots, K\}$, and $y_{n-1}(k,\ell) = (1 - f(\ell))(k-\ell)(1 - g(k-\ell))e^{\frac{1}{2}(\sigma^2 + \rho_{n-1}^2) - h(\ell)}$ is increasing in k and decreasing in ℓ .