TRAJECTORY BASED MARKET MODELS WITH OPERATIONAL ASSUMPTIONS

by

Andrew W.L. Fleck

B.Sc., Carleton University, 2014

A thesis

presented to Ryerson University

in partial fulfilment

of the requirements for the Degree of

Master of Science

in the program of

Applied Mathematics

Toronto, Ontario, Canada, 2016

DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public

ABSTRACT

TRAJECTORY BASED MARKET MODELS WITH OPERATIONAL ASSUMPTIONS

Master of Science 2016

Andrew Fleck

Applied Mathematics

Ryerson University

Mathematical finance makes use of stochastic processes to model sources of uncertainty in market prices. Such models have helped in the assessment of many financial situations. These approaches impose the stochastic process *a priori* which is then fitted to data. Hence, unchecked hypotheses can creep into the formalism and observable phenomena plays little role in building the model fundamentals.

We attempt to reverse the procedure in order to include presumably more realistic price movements. Operational assumptions are used to construct a trajectory set relating discrete chart properties with investors' portfolio re-balancing preferences. By identifying features of these trajectories we can construct models that capture different sources of risk and use a geometric procedure to produce replication bounds for a contingent claim.

Why a future unfolding chart fails to belong to the proposed trajectory set is testable. A preliminary risk-reward analysis based on this is also developed.

ACKNOWLEDGEMENTS

Many thanks to my two supervisors Dr. Sebastian Ferrando and Dr. Alexey Rubtsov for their discerning editing and guidance. Also, thanks to my officemate and colleague Nolan Nichols for his critical but honest take on my modelling ideas.

TABLE OF CONTENTS

LIST OF FIGURES

List of Appendices

Chapter 1

INTRODUCTION

Modern mathematical finance relies on stochastic processes to account for the sources of uncertainty which are present in market prices. Relying on such models, the existing literature has shown how to successfully encode a rich and versatile array of financial situations ranging from risk assessment, modelling of defaults and market bubbles as well as credit risk, just to name a few financial phenomena that have been modelled.

Despite such happy situation, it is reasonable to speculate that the above approach may not capture all possible sources of uncertainties or it may introduce unnecessary theoretical baggage. Notice that a main input to the construction of a stochastic process is its probability distribution. This construction assumes, in one form or another, some kind of stationarity, in particular a multitude of 0-measure events are part of the formalism. That is, impossible events, according to the stochastic model, make their way into the theory without financial justification and purely as a result of formal mathematical constructions. The ensuing risk implicit in these assumptions could be called (stochastic process) model based risk.

This model based risk can take different forms. For example the form of the underlying probability distribution is assumed to be "well behaved" in the sense that it's ultimately normal or some modification therein. However some econometric studies ([3],[4]) have found it easy to reject the Gaussian random walk hypothesis. Others have proposed that the discrepancy is large enough that the standard models do not serve as a good approximation of the real world even with modifications ([5]).

Most problematic of these assertions of model risk is the idea that the real world underlying distributions have infinite second moment making the use of standard stochastic calculus tools difficult if not impossible ([5], [4]). Attempts to remedy this appear to be unable to do so without requiring exiting derivatives prices as an exogenous input ([6]). Naively this seems to undermine the point of developing models in the first place.

There is a large amount of research dealing with the challenges of model based risk, e.g. the stream of literature labeled ”robust modeling” (for recent contributions see [10] and [7]). Such an approach is close to the spirit of our thesis in that the probabilistic assumptions are weakened (or non existing). A goal is to incorporate uncertainty about the probability distribution as well as its support (e.g. by taking into consideration non equivalent measures).

The main thrust of the present thesis illustrates, by example, a formalism that is centered on a given set of trajectories and which does not rely on a stochastic framework. The idea is that, trajectories besides being directly observable, carry useful financial information and hence it is reasonable to model them directly (and not as a by-product of an stochastic process construction). The approach is described in [8] to where we refer for any required details. The approach in [8] had before now not been implemented in a practical way. Here we attempt to balance the accuracy of the models produced by said approach over different time scales and implementation concerns.

The approach in this thesis additionally seeks to remedy a modelling issue separate from model based risk alone. When discussing challenges to the standard approaches used by Mathematical finance it is worth considering why hypothetical unchecked assumptions would have become so pervasive. Alternative and presumably more realistic distributions may be difficult to simulate. Additionally said distributions may not be well behaved enough to use the tools of stochastic calculus.

A deeper issue may in fact be a philosophical one. The standard modelling philosophy of Finance and Economics has been to discard the realism of assumptions in favour of the accuracy of model results compared to the real world [1]. While this approach is certainly a natural response to the limitations of the subject (lack of controlled experimentation etc...), some ([2]) have challenged that this leads to the creation of ”chameleon models”, toy models that are taken to be ontologically true based on the accuracy of their results which are most likely over fit. In this thesis we add to this discussion by asserting that an additional point to consider is the lack of variables in mathematical finance that are operational; variables that can be defined through a measurement process independent of the model.

The thesis is organized as follows. The next section provides a short description to the general framework, any related material that we may need will be introduced at due time in the thesis. The section following this will outline the methodology of modelling with operational variables, this will be expanded on in the next chapter. Details and issues related to implementation will be handled briefly in Chapter 3. Finally we will develop specific models and look at a preliminary risk-reward framework before concluding.

1.0.1 Trajectory Based Models

The framework of the thesis is a discrete market model $\mathcal{M} = \mathcal{M}(s_0) = \mathcal{S} \times \mathcal{H}$, $S = \{S_i\} \in \mathcal{S}$ is a sequence of real numbers called a trajectory with $S_0 = s_0$ for all $S \in \mathcal{S}$. $H = \{H_i\} \in \mathcal{H}$ a sequence of functions acting on $\mathcal{S}$ representing the portfolio holdings $H(S) = \{H_i(S)\}$ along S; we assume $H_i(S) = H_i(S_0, \ldots, S_i)$. There is also an integer $N_H(S)$ representing the last portfolio rebalance. Portfolio values for self financing strategies are given by

$$V_H(i, S) = V_H(0, S_0, w_0) + \sum_{K=0}^{i-1} H_k(S)\ (S_{k+1} - S_k). \tag{1.1}$$

The models are *discrete* in the sense that we index potential portfolio rebalances, $H_i(S) \rightarrow H_{i+1}(S)$, by integer numbers. Otherwise, stock charts and investment amounts can take values in general subsets of the real numbers, data could flow in a time continuous manner and portfolio rebalances could be triggered by arbitrary events without the need to be associated to a time variable.

The following conditional spaces play a key role. Given $\mathcal{M}$, $S \in \mathcal{S}$ and $k \geq 0$ fixed, set:

$$\mathcal{S}_{(S,k)} \equiv \{\tilde{S} \in \mathcal{S} : \tilde{S}_i = S_i, 0 \leq i \leq k\}.$$

The cardinality of these sets indicate the incomplete nature of the markets that we are introducing. The analogue to the sets $\mathcal{S}_{(S,k)}$ in stochastic models are, in general, sets of measure zero. Unless specified otherwise, Z denotes a general function $Z : \mathcal{S} \to \mathbb{R}$, pairs (S, k) or triples (S, k, H), $S \in \mathcal{S},\ k \geq 0, H \in \mathcal{H}$ will be referred as *nodes*.

The quantities $\overline{V}(S_0, Z, \mathcal{M})$ and $\underline{V}(S_0, Z, \mathcal{M})$, introduced below, are the usual super and sub hedging prices defined in robust frameworks (e.g. [**?**]), respectively, for a contingent claim (function) $Z : \mathcal{S} \to \mathbb{R}$.

Definition 1.0.1 (Conditional Minmax Bounds)**.** Given a discrete market $\mathcal{M} = \mathcal{S} \times \mathcal{H}$ and a node (S, k) define

$$\overline{V}_k(S, Z, \mathcal{M}) \equiv \inf_{H \in \mathcal{H}} \sup_{\tilde{S} \in \mathcal{S}_{(S,k)}} [Z(\tilde{S}) - \sum_{i=k}^{N_H(S)-1} H_i(\tilde{S})(\tilde{S}_{i+1} - \tilde{S}_i)]. \tag{1.2}$$

Also $\underline{V}_k(S, Z, \mathcal{M}) \equiv -\overline{V}_k(S, -Z, \mathcal{M})$. Set $\overline{V}(S_0, Z, \mathcal{M}) \equiv \overline{V}_0(S, Z, \mathcal{M})$ and $\underline{V}(S_0, Z, \mathcal{M}) \equiv \underline{V}_0(S, Z, \mathcal{M})$ as well.

Definition 1.0.2 (Conditionally 0-Neutral)**.** We say that a discrete market $\mathcal{M}$ is *conditionally 0-neutral* at node (S, k) if

$$\overline{V}_k(S, Z = 0, \mathcal{M}) = 0.$$

For $k = 0$, the conditional 0-neutral property, which depends on S only through S_0, will be referred to as 0-neutral.

Properties relative to a given node will be referred as *local*; the following local definitions are fundamental for the approach.

Definition 1.0.3 (Local Conditions: 0-Neutral & Arbitrage-Free Nodes)**.** Given a trajectory space $\mathcal{S}$ and a node (S, j):

- (S, j) is called a 0-*neutral node* if

$$\sup_{\tilde{S} \in \mathcal{S}(S,j)} (\tilde{S}_{j+1} - S_j) \geq 0 \quad \text{and} \quad \inf_{\tilde{S} \in \mathcal{S}(S,j)} (\tilde{S}_{j+1} - S_j) \leq 0. \tag{1.3}$$

- (S, j) is called an *arbitrage-free node* if

$$\sup_{\tilde{S} \in \mathcal{S}(S,j)} (\tilde{S}_{j+1} - S_j) > 0 \quad \text{and} \quad \inf_{\tilde{S} \in \mathcal{S}(S,j)} (\tilde{S}_{j+1} - S_j) < 0 \tag{1.4}$$

  or

$$\sup_{\tilde{S} \in \mathcal{S}(S,j)} (\tilde{S}_{j+1} - S_j) = 0 = \inf_{\tilde{S} \in \mathcal{S}(S,j)} (\tilde{S}_{j+1} - S_j). \tag{1.5}$$

$\mathcal{S}$ is called *locally* 0-*neutral* if (1.3) holds at each node (S, j). $\mathcal{S}$ is said to be *locally arbitrage-free* if either (1.4) or (1.5) hold at each node (S, j). A node that satisfies (1.4) will be called an up-down node, and a node satisfying (1.5) will be called a *flat node.* A node that is 0-neutral but that is not an arbitrage-free node, will be called an *arbitrage node.*

The next definition is the natural notion of arbitrage.

Definition 1.0.4 (Arbitrage-Free Market)**.** Given a discrete market $\mathcal{M}$, $H \in \mathcal{H}$ is an arbitrage strategy if:

- $\forall S \in \mathcal{S}$, $V_H(N_H(S), S) \geq V_H(0, S_0)$.
- $\exists S^* \in \mathcal{S}$ satisfying $V_H(N_H(S^*), S^*)) > V_H(0, S_0)$.

$\mathcal{M}$ is said to be arbitrage-free if $\mathcal{H}$ contains no arbitrage strategies.

Here are some main points to have in mind as background for our thesis. Under general conditions on $\mathcal{H}$, the following holds (proofs and details are in [8]):

- If $\mathcal{M}$ is conditionally 0-neutral, then $\underline{V}_k(S, Z, \mathcal{M}) \leq \overline{V}_k(S, Z, \mathcal{M})$ (Theorem 1.1 below). The resulting interval can be interpreted as a price interval, for a claim Z, even though $\mathcal{M}$ may contain arbitrage opportunities.

- If $\mathcal{S}$ is locally 0-neutral, then $\mathcal{M}$ is conditionally 0-neutral.

- The notion of 0-neutral market is a weakening of no-arbitrage that allows for rational prices without any logical contradictions and with a basic financial supporting argument.

- A trajectory set $\mathcal{S}$ which is locally arbitrage-free guarantees $\mathcal{M}$ to be arbitrage free. Such set $\mathcal{S}$ is the trajectory analogue of a martingale process in a stochastic framework.

- Under 0-neutrality the sequence of projections $\Pi_k : \mathcal{S} \rightarrow \mathbb{R}$ satisfy $\overline{V}_k(\mathcal{S}, \Pi_{k+1}, \mathcal{M}) = \Pi_k$. Other martingale-like properties hold as well, e.g. an optional sampling theorem, in particular $\underline{V}(S, S_\tau, \mathcal{M}) = \overline{V}(S, S_\tau, \mathcal{M}) = S_0$, for a trajectory based stopping time τ.

Theorem 1.1. Consider a discrete market $\mathcal{M} = \mathcal{S} \times \mathcal{H}$, a function Z defined on $\mathcal{S}$, $S \in \mathcal{S}$ and $k \geq 0$ fixed. Assume that for N_H all $H \in \mathcal{H}$ are liquidated (i.e. $N_k(S) = 0 \ \ \forall \ S$ and $k \geq N_H(S)$). If $\mathcal{S} \times (\mathcal{H} + \mathcal{H})$ is conditionally 0-neutral at node (S, k), then

$$\underline{V}_k(S, Z, \mathcal{M}) \leq \overline{V}_k(S, Z, \mathcal{M}), \tag{1.6}$$

in particular,

$$\underline{V}(S_0, Z, \mathcal{M}) \leq \overline{V}(S_0, Z, \mathcal{M}).$$

For the present thesis, the above setting needs to be extended by incorporating other sources of uncertainty besides the variable S_i. This extra source of uncertainty will be denoted by $W = \{W_i\}$ which, in financial terms, will be considered to be an observable quantity. This is analogous to moving from the natural filtration to an augmented filtration in the stochastic setting. The sequence elements W_i are assumed to belong to abstract sets Ω_i from which we only require to have defined an equality relationship. We provide some of the required notation and formal definitions. For (s_0, w_0) fixed as well as given sets $\sum_i, \Omega_i$, we change the notation of the trajectory set from $\mathcal{S}$ to $\mathcal{S}^{\mathcal{W}}$, here are some details:

$$\mathcal{S}^{\mathcal{W}}(s_0, w_0) \subseteq \mathcal{S}^{\mathcal{W}}_{\infty}(s_0, w_0) \equiv \{\mathbf{S} = \{\mathbf{S}_i \equiv (S_i, W_i)\}_{i \geq 0} : S_i \in \Sigma_i, W_i \in \Omega_i,\ (S_0, W_0) = (s_0, w_0)\}.$$

$$H_i(\mathbf{S}) = H_i(\mathbf{S}_0, \dots, \mathbf{S}_i). \tag{1.7}$$

$$\mathcal{S}^{\mathcal{W}}_{(\mathbf{S},k)}(s_0, w_0) \equiv \{\tilde{\mathbf{S}} \in \mathcal{S}^{\mathcal{W}}(s_0, w_0),\ \tilde{\mathbf{S}}_i = \mathbf{S}_i, \quad 0 \leq i \leq k\}. \tag{1.8}$$

$$V_H(k, \mathbf{S}) = V_H(0, (S_0, w_0)) + \sum_{i=0}^{k-1} H_i(\mathbf{S}) \Delta_i S.$$

In some of the models to be introduced in this thesis we will have $W_i = T_i$, where T_i will be associated to transaction time t_i. Such a case will give a two dimensional model with trajectory components (S_i, T_i). A three dimensional model will also be considered with trajectory components (S_i, W_i, T_i) with W_i associated to sampled quadratic variation. So, we will be abusing the notation as follows $W_i = (W_i, T_i)$ where the W_i in the left hand side is our abstract trajectory component introduced in the general formalism and the W_i in the right hand side will be sampled quadratic variation.

1.0.2 Operational Models Methodology

Placing emphasis on a trajectory set is to be contrasted to a stochastic process approach where the support of the process is a by-product of specifying its law in the first place. As already indicated, the expectation is that the trajectory set, being a direct input to modeling, will reflect useful features of financial prices. Therefore, construction of trajectory sets is a major undertaking from this new point of view.

The present thesis puts forward a general approach to trajectory set construction with the intent to model a single stock chart (*chart*: observable numerical value of a given financial variable). We denote chart values by x(t). The approach followed is general but it is here developed by example, the remaining of this introduction describes in general terms the motivations and general features of the approach.

From our point of view, the *unjustified* use of a stochastic process as mentioned above is questionable. A main problem we see, is that unobservable variables are linked to chart values by *imposing* an evolution equation. As a side remark we mention that this procedure is in widespread use in modern stochastic modeling via stochastic ordinary or partial differential equations. The reliance on such equations is not something that we dispute, our point is that the model introduces variables that do not have a well defined empirical meaning. An example of such issue is given by “volatility”; in practice there are several ways of measuring volatility providing non comparable values. Despite this, such a notion gets related to a specific feature of a stochastic process model. It is reasonable to conclude that ambiguities and confusion spreads due to the fact that the model dependent notion of volatility may not reflect entirely, or consistently, properties implied by the different ways of measuring volatility ([9]). The analogous situation in physics is the measuring of the diffusion coefficients for a Brownian particle where well defined physical situations give consistent observational results backed up by detailed physical knowledge of the phenomena. More generally,one could compare the situation with how definitions and a minimal choice of fundamental variables is done in physical theories, in particular basic concepts and measurable quantities in thermodynamics see [13], [14] and [11].

Of course, the fact that volatility has not established itself as a directly observable quantity (in which case it would have a meaning with lesser dependence of a specific stochastic process model), gets patched by using statistical theory. This theory provides results indicating how to estimate various parameters of the stochastic model, in particular how to evaluate volatility. Without getting into a debate of this delicate topic (the proper use of statistics in science), we just mention that the statistical results may be of dubious applicability and may also complicate the analysis of the validity of the models put forward.

The above arguments are meant just to serve as preamble to *operational models* for charts. We provide some of the fundamental ideas as a list of methodological steps:

(1) We prescribe how investors interact with charts and other observable variables collected into a vector valued variable denoted generically by w. So, implicitly, one concentrates in a specific class of investors. The process leads to the availability of chart samples $x(r_l)$ as well as $w(r_l)$ samples (along a sequence of x-dependent times $\{r_l\}$). As the values $w(r_l)$ are obtained for a given x we could use the notation $w(x, r)$ but w can depend on other financial variables as well.

(2) Those samples are used by the investor to monitor the chart changes and decide when to rebalance her portfolio. The rebalancing times are denoted by t_i with $t_i \in \{r_l\}$ and are not set apriori (i.e. $t_i = t_i(x, w)$), namely their values depend on the values of the samples.

(3) We call the above setup *operational* as the chart samples are obtained by an investor operating on a set of financial variables. The main implication is that all variables and parameters used have an empirical/operational meaning.

(4) The discreteness of chart samples as well as constraints on portfolio rebalances on the part of the investor introduces relationships between the sampled variables at rebalancing times t_i.

(5) One can also make assumptions in the form of relationships between the samples $x(r_l)$ and $w(r_l)$ with the goal of further constraining the trajectory set.

(6) The full collection of relationships are used to define a trajectory set $\mathcal{S}^{\mathcal{W}}$, with elements $\mathbf{S} = \{(S_i, W_i)\}$, in a combinatorial way, i.e. all trajectories obeying the said relationships are in $\mathcal{S}^{\mathcal{W}}$ by definition (as previously anticipated W_i could be a vector variable).

One then associates:

$$x(t_i) \to S_i \text{ and } \ w(x, t_i) \to W_i.$$

A market model is then defined by $\mathcal{M} = \mathcal{S}^{\mathcal{W}} \times \mathcal{H}$, where $\mathcal{H}$ is a set of portfolios.

The ensuing risk associated to model $\mathcal{M}$ is given by the possibility that the samples $x(t_i),\ w(x, t_i)$ imply that the associated trajectory variables S_i, W_i satisfy $\{(S_i, W_i)\} \notin \mathcal{S}^{\mathcal{W}}$.

Parameters appearing in operational models are estimated in a worst case way. This is a naive approach but as operational models reflect a real setup, convergence of values as more data is aggregated, is expected.

To compare the above described modeling approach to a conventional one we provide the following explanation: assume one has a model for $x(t)$, continuous time, and has a sequence of stopping times τ_n one then obtains samples $x(\tau_n(x))$. One can think that we are modeling directly the values $x(\tau_n(x))$ (i.e. without stipulating the functions $x(t)$ and the stopping times τ_n separately) by prescribing objective/operational conditions leading to relationships among the sampled variables. The fact that the conditions for obtaining the samples are objective is used for estimation and having a well defined risk-reward interpretation of the resulting model.

Let us address a natural potential objection; it is customary that the "real" model should be independent of investors preferences while we intent to define chart values S_i that result from investors interaction with sampled data. The implication is that the resulting chart model has a subjective component. The answer to this objection is that the model is objective as the values (S_i, W_i) are a conjunction of the chart moving in a certain way and the investor reacting in a specified manner to such a change. Both of these events are objective as the former can be observed (or not) and the second is operational (i.e. it is the performance of an action). Hence, it follows that the resulting model does relate to a class of investors and it has an objective character as well. One may also think in analogous terms to physical experiments, the investor, with its monitoring and portfolio rebalancing is setting up conditions for experiments, the trajectory set being the set of possible outcomes.

Chapter 2

OPERATIONAL MODELS AND A GENERAL TRAJECTORY BASED FRAMEWORK

Consider the following philosophical preliminaries. Assume for a moment that the sciences of thermodynamics and statistical mechanics does not in fact exist. How would one go about developing it? The subject matter concerns the bulk properties of matter that we observe at the macroscopic scale that do not depend on the specific atomic make-up of materials, e.g things that can be generalized across matter entirely. That rules out discussions of say hardness, stiffness, conductivity etc... What remains are general properties of assemblages of atoms: temperature, pressure, volume and mass.

All these things are easily understood on an intuitive level. But again, assuming no knowledge of thermodynamics and it becomes clear that these concepts can be very poorly defined. A stove top is more hot to the touch than a counter, an air balloon deforms more easily than a water filled one, but is this how you would define temperature or pressure? How could you guarantee a consistency of observation across different settings and experimenters? How would you perform calculations with your observations? In order to avoid these complications one can resort to the use of *operational definitions*

An operational definition is simply a way to define a quantity through the act of measurement. For example, temperature can be defined by the way it is measured with a thermometer or mass with a scale. This avoids the issue of consistency. Additionally by making these quantities treatable mathematically, it allows us to operationally define more phenomenon that may not be directly measurable, but whose existence is indicated by other effects (like heat in our thermodynamics example).

Once we have operational definitions we can perform experiments and develop mathematical laws involving assemblages of operational variables in order to infer deeper truths about the universe. While this may seem exceedingly trivial consider an example of how this process can be useful. Einstein observed that two operational definitions, the inertial definition of mass (mass measured by measuring acceleration under a force) and the gravitational one (mass measured on a scale) consistently gave the same value. By proposing a theory whereby the two were equivalent he was able to derive the General Theory of Relativity.

While conventional probabilistic models of finance are not non-operational *per se* (as they easily can be made to be). They are not developed with this view in mind and as a result we suggest there is significant confusion when applying these models and attempting to extend them from general principles. Our goal is then provide, by example, a way to define market models that minimizes the use of theoretical probabilistic quantities and relies on operational definitions and assumptions to build a trajectory set suitable for chart modelling.

Here we will proceed in the following way. First we will define an assemblage of operational variables relating to the movement of a underlying tradable security and propose a general unspecified model of their dynamics. Then we will consider a function on these variables modelling a contingent claim. Finally we will propose a theory of no arbitrage on these variables and use convex optimization theory to relate this to our price bounds.

2.1 Operationalization

Mathematical Finance usually proceeds under a standard collection of methodologies related to applications from Stochastic Calculus and optimization. Certain abstract principles like no-arbitrage and risk-neutrality are used to justify the creation of stochastic models describing the evolution through time of a set of securities that serve as the underlying assets in a derivative. These models are then fit to market data using various tools from probability to estimate the model parameters such as the volatility and more rarely the drift.

This approach suffers from some key limitations. The underlying movement of prices is driven by a complex and highly dynamic set of interactions between market participants who are themselves complicated to describe. The models then describing the underlying will then at best be approximations. Related to this issue is the idea that the parameters of the models are unobservable in a direct sense. Assuming a degree of accuracy in the models, one can still at best only hope to estimate in a probabilistic sense model parameters from data. This latter issue is more problematic for philosophical reasons.

In the current framework it is difficult to create operational models. As discussed earlier this is unlike in say physics where one can define quantities operationally through measurement and abstractly create theory describing relationships between state variables(temperature, speed, current, weight etc...). Put another way using an example, how does one define operationally volatility? Volatility can be thought of as the change in quadratic variation through time. Quadratic variation may be a good idealization in some markets, but it is not known when this will be the case. Furthermore true quadratic variation as defined by a limit with respect to a time partition is itself impossible to measure given that markets are discrete. Maximum likelihood estimation methods are highly dependent on the choice of underlying model and have their own problems as a result. It would be desirable then to find a way of defining operational variables that are observable with which to construct models.

The rest of the chapter describes how to do this. First we will make realistic assumptions on charts and investors and define an operational ”observed chart”. Secondly, using variables from this setting we will construct the general model of a ”trajectory set”. Lastly we will show that under certain easily verifiable assumptions the observed chart will lie in the trajectory set.

2.2 The Operational Setting

2.2.1 Assumptions on Charts and Investors

Financial market data is inherently discrete. Stock quotes are often separated by seconds or minutes as order books take in new orders and list completed trades in cycles and stock ticks usually restrict listed price movements to a minimum. Furthermore most investors can't continuously trade due to losses from transaction costs and will only act on the market over a specific time horizon with the idea that only moves of a certain magnitude can be captured then.

We seek to capture all this in the following setting. First we point out that there will be an inherent granularity to the way investors view the discrete stream of data as it comes to them. We assume that given a stock chart $x(t)$ over some period of time $[0, T]$ it is a subset of all functions $f : [0, T] \to \{k\delta_0 : k \in \mathbb{N}\}$.

Moreover given a specific investors preference, small movements will largely be ignored below a certain threshold, δ. Realistically this will be much larger than the smallest possible price movement δ_o. We can interpret this as the investor specific scale at which one views the market and samples on accordingly. That is, any other price movements will be considered too small to be worth taking into consideration and beyond that the investor will sample the market to gain information.

Sampling Times Therefore, we assume given a stock chart $x(t)$ over some period of time $[0, T]$ there is a sequence of dynamically sampled times $\{r_l = r_l(x)\}$ such that:

$$\delta \leq |x(r_{l+1}) - x(r_l)|, 0 \leq l + 1 < L, r_0 = 0, r_L = T \tag{2.1}$$

Additionally we assume there is some minimum amount of time Δ between sampling times corresponding to the time resolution of $x(t)$ and a maximum amount of time before sampling again τ. Specifically, $\tau \geq r_{l+1} - r_l \geq \Delta$ and $\exists K$ such that $K\Delta = T$ and $r_l \in \{n\Delta : n \in \mathbb{N}\}$.

We will assume that there is a range of times an investor will sample before rebalancing his hedging portfolio given by a parameter $s_i = s(x, t_i) \in [s^{min}, s^{max}]$. This last assumption of an s parameter comes from the fact that our investor will be acting over a finite period of time T and would like to control the minimum and maximum number of transactions over that period.

Rebalancing Times Given a sequence $\{r_l\}$ satisfying (2.1) let there be a subsequence $\{t_i\} \subseteq \{r_l\}$ representing the possible rebalancing times where an investor will act on the market. We only require that $t_0 = r_0$. We will obtain the following:

$$\Delta_i t = t_{i+1} - t_i = \sum_{j=0}^{s_i-1} (r_{l_i+j+1} - r_{l_i+j}) \tag{2.2}$$

Chart Vales at rebalancing Times Given that the trajectory moves an arbitrary amount $\Delta_i x = x(t_{i+1}) - x(t_i)$

$$\Delta_i x = x(r_{l_{i+1}}) - x(r_{l_i}) = \sum_{j=0}^{s_i-1} x(r_{l_i+j+1}) - x(r_{l_i+j}) \tag{2.3}$$

Realized Quadratic Variation Defining sampling quadratic variation as:

$$\Delta_i w = w(x, t_{i+1}) - w(x, t_i) = w(x, r_{l_{i+1}}) - w(x, r_{l_i}) = \sum_{j=0}^{s_i-1} (x(r_{l_i+j+1}) - x(r_{l_i+j}))^2 \tag{2.4}$$

Note that s_i is the same value in (2.2),(2.3) and (2.4)

2.3 Main Trajectory Model

Having defined our variables operationally we wish to construct models describing possible observed charts. In order to do this, we will need to mathematically define an abstract setting corresponding to the operational one.

Defining model parameters Given that we assume $x(t)$ has the range $\{k\delta_0 : k \in \mathbb{N}\}$ and $t_i \in \{n\Delta : n \in \mathbb{N}\}$ we can model the rebalancing price levels as $S_i = k_i\delta_o$ and $T_i = n_i\Delta$ where k_i and n_i are integers. In the same way, w(x,t) will have the range $\{j^2\delta_0^2 : j \in \mathbb{N}\}$ and we can write the realized quadratic variation at rebalancing times as $W_i = j_i\delta_o^2$.

Similarly we can model our investor assumptions as:

$$\boxed{\begin{aligned} |\Delta_i S| &\geq \delta && (2.5a) \\ \tau \geq |\Delta_i T| &\geq \Delta && (2.5b) \end{aligned}}$$

Furthermore we will need some way of modelling an investor sampling the market as well as rebalancing.

Ultimately we will want the correspondence $S_i = x(t_i)$. So then momentarily ignoring the distinction between sampling and rebalancing we get that $x(r_{l_i+j+1}) - x(r_{l_i+j}) = (k_{i+j+1} - k_{i+j})\delta_o = m_j\delta_o$ for some $m_j \in \mathbb{Z}$ and similarly for (2) you could define some q_j such that $r_{l_i+j+1} - r_{l_i+j} = (n_{i+1} - n_i)\Delta = q_j\Delta$. By definition we have that $|x(r_{l_i+j+1}) - x(r_{l_i+j})|$ must satisfy (1). As a result we get that $|k_{i+j+1} - k_{j+1}|\delta_o = |m_j|\delta_o \geq \delta$. Say that $z = \delta/\delta_o$.

We can then define the relationships between our model parameters:

$$\boxed{\begin{aligned} \Delta_i S = (k_{i+1} - k_i)\delta_o &= \sum_{j=0}^{s_i-1} m_j\delta_o && (2.6a) \\ \Delta_i W = (j_{i+1} - j_i)\delta_o^2 &= \sum_{j=0}^{s_i-1} m_j^2\delta_o^2 && (2.6b) \\ \Delta_i T = (n_{i+1} - n_i)\Delta &= \sum_{j=0}^{s_i-1} q_j\Delta && (2.6c) \\ |m_j| \geq \frac{\delta}{\delta_o} &= z && (2.6d) \end{aligned}}$$

Model Hypothesis At this point, it is good to identify the source of the different hypothesis on the models. We made some assumptions on a chart. Namely that it was a function that has a range divisible by δ_o and a time domain divisible by Δ. We then made some assumptions on an investor acting on these charts. Specifically we asserted that they will only sample information from the chart after price movements of at least δ, before $\tau\Delta$ time has passed. Finally we assumed that there was a number of times the investor will sample before rebalancing his portfolio, s.

Clearly the new parameters m_j and q_j will be the key to determining the relationships between rebalancing times. The hypotheses that govern them will identically specify models of this kind. In later chapters we will discuss specific possible model hypotheses and their effects. For the time being it is sufficient to describe the following set of integer sequences that is identically determined by a set of model hypotheses involving m_j and q_j given by $\mathcal{A}$ and an investor specified range for s.

Conditional Set The conditional set,

$$\mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i) = \{(m_j, q_j)_{j=1}^{s_i} | \mathcal{A} \text{ is satisfied}, s_i \in [s^{min}, s^{max}]\} \tag{2.7}$$

i.e $\mathcal{N}_o^{\mathcal{A}}$ is a set of sequences of pairs of the form $(m_j, q_j)_{j=1}^{s_i}$ that satisfy the hypotheses of a specific model. This is the set specified by a model describing the possible relative distribution of points an investor will be able to rebalance at.

N.B given the assumptions on our sampling at rebalancing times we must minimally assume that $m_j \in \mathbb{Z} \setminus (-z, z)$.

For example, $\mathcal{A}$ may consist of the hypothesis:

$$\mathcal{A} = \{s_i = 1, |m_j| = q_j = 1, \quad \forall j \quad m_j^2 \delta_o^2 = \sigma q_j \Delta\}$$

corresponding to a standard binomial model.

Trajectory Sets Now that we have discussed the setting and defined the parameters and their relationships we wish to unify these ideas into a discrete object with which we can build models. Given the rules laid out by equations (2.5) and (2.6) we can imagine starting at an initial point (S_o, W_o, t_o) and moving to a new point (S_1, W_1, T_1) such that (2.6) are satisfied. Furthermore we could do this again at our new point and so on until we would have a long sequence of points. This "trajectory" could be bundled together into a "trajectory set" and would map out all possible futures our investor could move along.

Defining the trajectory set Given a set of hypotheses $\mathcal{A}$, Set $(S_0, W_0, T_0) = (s_0 = k_0\ \delta_0, 0\ \delta_0^2, 0\ \Delta)$ and define:

$$\boxed{\mathcal{S}^{\mathcal{W}} \equiv \{\mathbf{S} = \{(S_i, W_i, T_i)\} : (S_{i+1}, W_{i+1}, T_{i+1}) \in \mathcal{N}_{\mathcal{A}}((S_i, W_i, T_i))\}} \tag{2.8}$$

where, for $S_i = k_i\ \delta_0, W_i = j_i\ \delta_0^2, T_i = n_i\ \Delta$,

$$\begin{gathered}\mathcal{N}_{\mathcal{A}}((S_i, W_i, T_i)) \equiv \{(k_{i+1}\delta_0, j_{i+1}\ \delta_0^2, n_{i+1}\Delta)| \\ (k_{i+1}\delta_0, j_{i+1}\ \delta_0^2, n_{i+1}\Delta) = (k_i\delta_0 + \sum_{j=0}^{s_i-1} m_j\delta_o, j_i\delta_0^2 + \sum_{j=0}^{s_i-1} m_j^2\delta_o^2, n_i\Delta + \sum_{j=0}^{s_i-1} q_j\Delta) \\ (m_j, q_j)_{j=1}^{s_i} \in \mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i)\}\end{gathered} \tag{2.9}$$

Paths in $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$ will be called trajectories. All subsequent models presented here will simply further specify this trajectory set, by specifying the conditional set $\mathcal{N}_o^{\mathcal{A}}$ through $\mathcal{A}$.

A major point to be mindful of is that the $\{(S_i, W_i, T_i)\}$ trajectories we are modelling do not in fact represent some possible unfolding stock chart, but rather the points in the unfolding future where the investor will rebalance his holdings. One should think of the points in the trajectory set as belonging to some common points in the future between which many possible unfolding charts are possible. Furthermore this model accounts for a certain element of choice. By not specifying the number of transactions *a priori* the possible trajectories from this model account for a wide range of investor preferences.

In order to think about the underlying process one should not look at the trajectories themselves but rather the parameters m_j and q_j and their relative magnitudes and interpretation. Once a δ has been specified and given Δ the smallest amount of time before an investor will have new information, the range of parameters m_j and q_j will define the conditional set and the next "sampling times" of interest. That is, a set of possible states where the underlying has moved by at least δ in value in at least Δ time. While simplistic compared to say a continuous time stochastic process the main advantage of using these parameters to specify the movements of the underlying is that they are observable in nature and have very clear operational interpretations for an investor.

A final point regarding this approach is that given a specific δ and τ we are modelling an equivalent *class* of investors in the market that is acting on the market in a predetermined operational way. We will see that the models will be affected by this in a deep way even as far as parameter estimation is concerned. This approach has the advantage of eliminating from the actions and trajectory assumptions that are out of our investors reach while still incorporating this information into the replication price bounds.

2.4 Model Agreement

The motivation behind the framework described so far is the idea that the operational setting will provide a clear set of hypothesis that need to be fulfilled by an unfolding observed chart in order to to match a particular model. Once a δ, τ and Δ have been specified by an investor we will have the necessary conditions to specify our sampling times $\{r_l\}$. That is, if we are at $(x(t_i), w(x, t_i), t_i)$ we have the necessary conditions to identify $(x(t_{i+1}), w(x, t_{i+1}), t_{i+1})$. Additionally once we have specified a set of model hypothesis and corresponding $\mathcal{N}_o^{\mathcal{A}}$ we get our trajectory set $\mathcal{S}^{\mathcal{W}}$.

We wish to identify the sufficient conditions our unfolding observed chart needs to satisfy in order to have and agreement with the model.

Proposition 2.1. Assume a sequence of samples from an unfolding chart $(x(r_l), w(x, r_l), r_l)_{l=1}^{L}$. and an investor specified $[s^{min}, s^{max}]$ such that $s \in [s^{min}, s^{max}]$ i.e the range of possible sampling before rebalances. Given a trajectory set $\mathcal{S}^{\mathcal{W}}$ with model assumptions $\mathcal{A}$ where $(x(t_0), w(x, t_0), t_0) = (S_0, W_0, T_0)$ if the following condition holds:

$$(\tfrac{x(r_{l_i+j+1})-x(r_{l_i+j})}{\delta_o}, \tfrac{r_{l_i+j+1}-r_{l_i+j}}{\Delta})_{j=1}^{s} \in \mathcal{N}_o^{\mathcal{A}}(x(t_i), w(x, t_i), t_i)$$

Then there is a $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$ and a sequence of rebalancing times $\{t_i\}$ such that $(x(t_i), w(x, t_i), t_i) = (S_i, W_i, T_i)$

Proof. We will construct the sequence of rebalancing times and the trajectory iteratively using the following procedure. Say we are at $(x(t_i), w(x, t_i), t_i)$.

$$(\frac{x(r_{l_i+j+1}) - x(r_{l_i+j})}{\delta_o}, \frac{r_{l_i+j+1} - r_{l_i+j}}{\Delta})_{j=1}^{s_i} \in \mathcal{N}_o^{\mathcal{A}}(x(t_i), w(x, t_i), t_i)$$

$$\Rightarrow (\sum_{j=1}^{s_i} m_j, \sum_{j=1}^{s_i} m_j^2, \sum_{j=1}^{s_i} q_j) = (\sum_{j=1}^{s_i} \frac{x(r_{l_i+j+1}) - x(r_{l_i+j})}{\delta_o}, \sum_{j=1}^{s_i} \frac{x(r_{l_i+j+1}) - x(r_{l_i+j})^2}{\delta_o^2}, \sum_{j=1}^{s_i} \frac{r_{l_i+j+1} - r_{l_i+j}}{\Delta})$$

(2.2),(2.3) and (2.4)

$$\Rightarrow (\sum_{j=1}^{s_i} m_j\delta_o, \sum_{j=1}^{s_i} m_j^2\delta_o^2, \sum_{j=1}^{s_i} q_j\Delta) = (x(t_{i+1}) - x(t_i), w(x, t_i) - w(x, t_{i+1}), t_{i+1} - t_i)$$

If $(x(t_i), w(x, t_i), t_i) = (S_i, W_i, T_i)$ we get that

$$\Rightarrow (S_i + \sum_{j=1}^{s_i} m_j\delta_o, W_i + \sum_{j=1}^{s_i} m_j^2\delta_o^2, T_i + \sum_{j=1}^{s_i} q_j\Delta) = (x(t_{i+1}), w(x, t_{i+1}), t_{i+1})$$

Then by the definition of a trajectory set there must be a $(S_{i+1}, W_{i+1}, T_{i+1}) \in \mathcal{N}_{\mathcal{A}}((S_i, W_i, T_i))$ such that $(x(t_{i+1}), w(x, t_{i+1}), t_{i+1}) = (S_{i+1}, W_{i+1}, T_{i+1})$

□

If our sampled chart fails to maintain the relationship between successive elements as laid out by the model hypothesis in the conditional set we will not be able to match it to a trajectory in the model and clearly identify where the model failed. This will be very useful later on when we will be able to clearly identify sources of risk in our models.

2.5 Modelling Considerations: Global vs. Local Variables

The Main Trajectory model was constructed in a general fashion with no reference to specifying the model any further than the existence of a conditional set (we will look at specific model implementations in Chapter 4). This conditional set locally determined the dynamics of trajectories. Here we will look at connection between the local dynamics of the trajectory set determined by the conditional sets, and the global features of the trajectories and resulting prices.

We will mainly explore the main trade-off of modelling in this way. Models that better capture unfolding trajectories will likely produce wider price bounds. Conversely, models that have tighter price bounds will have more difficulty matching unfolding charts. This tension is not easy to negotiate but by understanding the connection between local and global properties of trajectory sets it is possible.

2.5.1 Local Variables

We will start by taking a look at the local dynamics of trajectories and defining some key variables. Recall we defined our main trajectory model in an iterative fashion i.e we said that if we were at some "parent" node (S_i, W_i, T_i) that there was a set of some later "children" nodes $(S_{i+1}, W_{i+1}, T_{i+1}) \in \mathcal{N}_{\mathcal{A}}((S_i, W_i, T_i))$ defined locally. The set $\mathcal{N}_{\mathcal{A}}((S_i, W_i, T_i))$ was in turn defined using what was defined as the conditional set $\mathcal{N}_o^{\mathcal{A}}$.

The later set $\mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i)$ was defined as the possible sequences of pairs $(m_j, q_j)_{j=1}^{s_i}$, $s_i \in$

$[s^{min}, s^{max}]$. These are dependent on both assumptions about charts and investors. The s parameter is defined to be within a range of values representing the number of times an investor will sample the market for information before rebalancing his portfolio. The individual m_j's are dependent on δ_o from the range of the chart and δ chosen by an investor. Likewise the q_j's are dependent on Δ the smallest time resolution of the chart and τ the longest an investor will wait before rebalancing his portfolio regardless of the market's movement.

Given a set of hypotheses $\mathcal{A}$ and a $[s^{min}, s^{max}]$, according to Proposition 2.1 gives us a criteria to identify if a chart fails to be in a trajectory set.

We will need the following intermediate parameters:

$$m^{max} = \sup_{\mathbf{S}\in\mathcal{S}^{\mathcal{W}}} \left\{ \sup_{i\geq 0} \left\{ m_j | (m_j, q_j)_{j=1}^{s_i} \in \mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i) \right\} \right\}$$

$$q^{max} = \sup_{\mathbf{S}\in\mathcal{S}^{\mathcal{W}}} \left\{ \sup_{i\geq 0} \left\{ q_j | (m_j, q_j)_{j=1}^{s_i} \in \mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i) \right\} \right\}$$

and analougously

$$m^{min} = \inf_{\mathbf{S}\in\mathcal{S}^{\mathcal{W}}} \left\{ \inf_{i\geq 0} \left\{ m_j | (m_j, q_j)_{j=1}^{s_i} \in \mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i) \right\} \right\}$$

$$q^{min} = \inf_{\mathbf{S}\in\mathcal{S}^{\mathcal{W}}} \left\{ \inf_{i\geq 0} \left\{ m_j | (m_j, q_j)_{j=1}^{s_i} \in \mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i) \right\} \right\}$$

In most models we will consider, the conditional set will be the same for all nodes in the trajectory set, but this may not be true generally. We emphasize that the quantities $s_{min}, s_{max}, q^{min}, q^{max}, m_j^{min}$ and m_j^{max} are defined globally. So we do not need to worry about the differences node-wise in the trajectory set.

The extreme possible time steps the quantities given by,

$$\underline{q} = s_{min}q^{min}, \quad \overline{q} = s_{max}q^{max}, \tag{2.10}$$

will give us the longest possible time lapse between rebalancing times $\overline{q}\Delta$ and the smallest $\underline{q}\Delta$.

The extreme possible stock movements the quantities given by,

$$\underline{m} = s_{min}m^{min}, \quad \overline{m} = s_{max}m^{max}, \tag{2.11}$$

will give us the most extreme stock movements between rebalancing times given by $\overline{m}\delta$ and the smallest $\underline{m}\delta$.

2.5.2 Global Variables

2.5.2.1 Number of rebalancing nodes

The trajectory set, as defined does not make a distinction between the unfolding price process and the hedging portfolio. To put another way, chart movements and rebalances are independent but both are are simultaneously constrained by operational assumptions. Restricting the choices an investor can make as the chart unfolds through time is important in order to investigate the effects on the price bounds. When defining $\mathcal{S}^{\mathcal{W}}$ we made no assumptions about how often or how much the investor will rebalance his holdings while hedging. While we may wish to restrict the the investors activities in regards to rebalancing (in light of transaction costs or other practical concerns) we must first understand the effects of the chosen parameters.

We will start by defining $N(\mathbf{S})$ for some $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$ as the number of points in the trajectory S. Further define N_1 and N_2 (N.B. this is not necessarily equal to $N_H(\mathbf{S})$ as there is no dependence on portfolio functions) bounding the size of all trajectories $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$ i.e $N_1 \leq N(\mathbf{S}) \leq N_2$. Recall that given the lifetime of our option T we must have $K\Delta = T$:

$$K\Delta = \sum_{i=0}^{N(\mathbf{S})-1} \Delta_i t \tag{2.12}$$

$$K = \sum_{i=0}^{N(\mathbf{S})-1} \sum_{j=0}^{s_i-1} q_j \tag{2.13}$$

2.5.2.2 Global quadratic variation bounds

As we will see in the next section, when modelling, quadratic variation will be a powerful variable in controlling stock movements through time. The reason quadratic variation and controlling it is so important is that globally it will have huge effects on the range on the price bounds. To see why, abusing notation we look at the alternative definition of the conditional price bounds in Appendix B. (we will elaborate on the meaning in the next chapter).

$$\overline{V}_i(\mathbf{S}, Z, \mathcal{M}) = \sup_{\mathbf{S}^\bullet, \mathbf{S}^\circ} \{\overline{V}_{i+1}(\mathbf{S}^\bullet, Z, \mathcal{M}) - u_{(\mathbf{S}^\bullet, \mathbf{S}^\circ)} \Delta_i S^\bullet\}$$

The allowable quadratic variation has a direct impact on the set of possible $\mathbf{S}^\bullet, \mathbf{S}^\circ$, where larger allowable quadratic variation will lead to a larger possible range. Given the properties of the sup and inf it is obvious then that a larger allowable variation in the next step will increase the distance between price bounds. Globally this means that trajectory sets with larger total allowed quadratic variation will admit larger price bounds

We then desire to model constraints on $w(x, t_i)$ of the form:

$$w(x, t_i) \in Q_{t_i}.$$

Where Q_{t_i} is the globally allowed quadratic variation up to that point.

2.5.3 Connecting the Local and Global Views

Constricting the number of rebalancing nodes First say we can make further assumptions on $s \in [s_{min}, s_{max}]$ namely that $s_{min} \geq 1$ and $s_{max} \leq N_2$.

While we would nominally expect the following:

$$K = N_2 \underline{q} = \sum_{i=0}^{N(\mathbf{S})-1} \sum_{j=0}^{s_i - 1} q_j = N_1 \overline{q}$$

This however is false for a non-obvious reason. It is important to recall that q^{min} and q^{max} have clear definitions separate from these considerations and that no matter what choice of s_{min} or s_{max} it may be that $\underline{q}, \overline{q}$ simply does not divide K without remainder. While this may seem odd, recall (1) which defined our sampling times. We required separately that $r_L = T$. This is because after the penultimate sample, the chart may not have moved δ. In much the same way, our observed minimum and maximum times to move δ are going to have little to do with the independent fact of the remaining lifetime of the option and we may be forced to rebalance at the end without having moved δ. To calculate the N_1 and N_2 we need to then consider all possible cases:

(1) $\mod(K, \overline{q}) \geq \underline{q} \implies K = a\overline{q} + b\underline{q} + r$

Where $a = \dfrac{K - \mod(K, \overline{q})}{\overline{q}}$, $b = \dfrac{(K - a\overline{q}) - \mod(K - a\overline{q}, \underline{q})}{\underline{q}} = \dfrac{r_o}{\underline{q}}$

$r = \mod(r_o, \underline{q})$

(a) $N_1 = a + b + 1$ if $r > 0$

(b) $N_1 = a + b$ if $r = 0$

(2) $\mod(K, \overline{q}) \leq \underline{q} \implies K = a\overline{q} + r$

Where $a = \dfrac{K - \mod(K, \overline{q})}{\overline{q}}$, $r = \mod(K, \overline{q})$

(a) $N_1 = a + 1$ if $r > 0$

(b) $N_1 = \dfrac{K}{\overline{q}}$ if $r = 0$

And in either case:

a) $N_2 = \dfrac{K - \mod (K, \underline{q})}{\underline{q}} + 1$, if $\mod (K, \underline{q}) > 0$

b) $N_2 = \dfrac{K}{\underline{q}}$ otherwise

The s_{min} parameter essentially represents the amount of times the investor will sample the market before rebalancing and likewise s_{max} the most the investor will sample the market. These represent the preference of the investor to how little or how often to rebalance. Given K, choices of s give some control over N_2, N_1 as we can see above. However the chosen values for q will impact $N(\mathbf{S})$ greatly.

Constricting trajectories through quadratic variation If we desire to place constraints on $w(x, t_i)$ we will need to analyse the roles of m_j and q_j. Assume that there are sets Q_{t_i} as previously defined.

In order to control the elements of these sets we will need to perform a similar analysis to the one that provided us the forms of N_1 and N_2. Given that we can easily replace K with some n_i without loss of generality assume for now we are looking at $w(x, T) \in Q_T$. We get:

$$
\begin{aligned}
w(x, T) &= \sum_{i=0}^{N(\mathbf{S})-1} \Delta_i W \\
w(x, T) &= \sum_{i=0}^{N(\mathbf{S})-1} \sum_{j=0}^{s_i - 1} m_j^2 \delta_o^2
\end{aligned}
$$

Which will give us:

$$N_1 s_{min} z^2 \delta_o^2 \leq w(x,T) \leq N_2 s_{max} (m_j^{max})^2 \delta_o^2 \tag{2.14}$$

So it is immediately clear then that $Q_T \subseteq [N_2 s_{min} z^2 \delta_o^2, N_1 s_{max} (m_j^{max})^2 \delta_o^2]$ and that there is a strong relationship between this global feature of the trajectory set and the local parameters describing the dynamics of the underlying.

This is important for two reasons:

(1) This connection between local and global features provides a possible way to estimate/modify parameters through time.

(2) It allows us to dynamically update local parameters based on the observed dynamics of realized quadratic variation (for both points see Model 2 in Chapter 4).

Chapter 3

IMPLEMENTING THE MODEL

The trajectory set introduced in the last chapter needs to be implemented in a way that allows the information to be stored and computationally acted on. Graph Theory is the most natural way of doing this. In this chapter we will introduce some graph theoretic analogues of our model definitions. Afterwards, we will discuss how to evaluate the conditional price bounds using dynamic programming and the convex hull algorithm. Finally we will discuss the basic details of the algorithms that encode the trajectory set and the evaluate the price bounds and their performance.

3.1 Graph Theory Preliminaries

Definition 3.1.1 (Graph)**.** A graph $G = (V(G), E(G)) = (V, E)$ consists of a nonempty set of nodes V, and a set of edges E, which is a binary relation on V.

Definition 3.1.2 (Directed vs. Undirected Graph)**.** A graph G is said to be *undirected* if the binary relation E is symmetric. If this is not the case it is *directed*.

Figure 1 bellow displays an example of an undirected graph where $V = \{1, 2, 3\}$ and $E = \{\{1, 3\}, \{1, 2\}, \{2, 1\}, \{2, 3\}, \{2, 2\}, \{3, 1\}\}$

Definition 3.1.3 (Graph Path)**.** Given a graph G where $V = \{v_1, ..., v_n\}$ a *path* in the graph is a set of nodes $\{v_{p_1}, ..., v_{p_m}\} \in V^m$ where $< v_{p_{i-1}}, v_{p_i} >\in E$

Definition 3.1.4 (Adjacency Matrix)**.** Given a graph G where $V = \{v_1, ..., v_n\}$ an adjacency matrix $A = \{a_{i,j}\}$ is a representation of the graph where $a_{i,j} = 1$ if $\{v_i, v_j\} \in E$ and zero otherwise

For example the adjacency matrix of the graph in Figure 3.1 is given by:

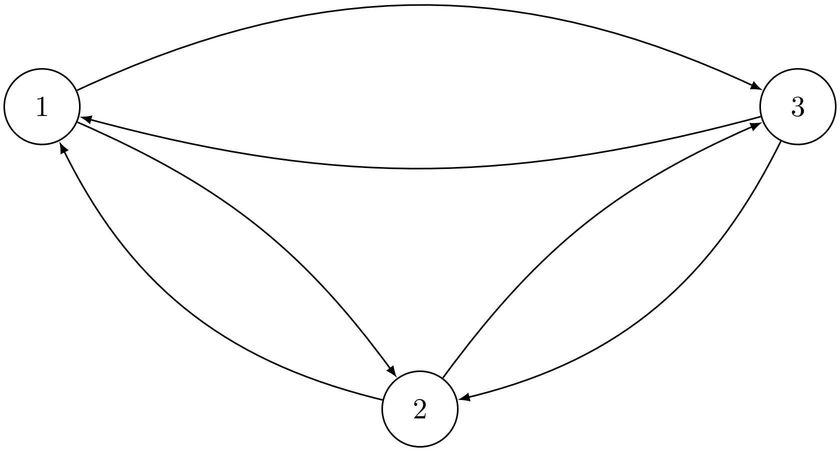


Figure 3.1: An undirected graph

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

N.B. This will be critically important. As it is the way in which we will store our trajectory sets.

Trajectory sets

Definition 3.1.5 (Integer Conditional Set)**.** Given parameters m^{min}, m^{max}, $q^{min}, q^{max}, s^{max}, s^{min}$ and a set of hypothesis $\mathcal{A}$ define the *integer conditional set* as

$$CS = \{(\sum_{j=0}^{s_i-1} m_j, \sum_{j=0}^{s_i-1} m_j^2, \sum_{j=0}^{s_i-1} q_j)|(m_j, q_j)_{j=1}^{s_i} \in \mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i)\} \tag{3.1}$$

Definition 3.1.6 (Integer Grid)**.** Given Integers $K_{max}, J_{max}, N_{max}$ referring to *the integer grid* will mean a subset of $\mathbb{Z}^3$ given by

$$IG(K_{max}, J_{max}, N_{max}) = \{(k, j, n)|k \in [-K_{max}, K_{max}], j \in [0, J_{max}], n \in [0, N_{max}]$$

Definition 3.1.7 (Trajectory Set)**.** Given some δ, Δ and $K_{max}, J_{max}, N_{max}$ define a set of nodes:

$$V = \{(S_i, W_i, T_i) = (k_i\delta_o, j_i\delta_o^2, n_i\Delta) | (k, j, n) \in IG(K_{max}, J_{max}, N_{max})\}$$

And given an integer conditional set CS a set of edges:

$$E = \{\langle (S_1, W_1, T_1), (S_2, W_2, T_2) \rangle = \langle (k_1\delta_o, j_1\delta_o^2, n_1\Delta), (k_2\delta_o, j_2\delta_o^2, n_2\Delta) \rangle | (k_2 - k_1, j_2 - j_1, k_2 - k_1) \in CS\}$$

The resulting graph $\{V, E\}$ encodes our trajectory set *Trajectory Set* $\mathcal{S}^{\mathcal{W}}$. To be clear, our trajectory set was originally defined as a set of sequences, the paths in this graph with starting node $(0, 0, 0)$ and ending where $n\Delta = T$ will represent these sequences.

3.2 Dynamic Programming and The Convex Hull

3.2.1 Dynamic Programming

Given a trajectory set as part of a discrete market model, it is not clear how to evaluate the conditional minmax bounds (Definition 1.0.1). A direct approach would involve calculating the value of the portfolio holdings over every possible trajectory for each portfolio function and keeping the optimal value. Naively this seems extremely computationally inefficient.

As it turns out for a class of optimization problems with a certain nested structure one can simply matters considerably. More specifically for a problem exhibiting what's called an optimal substructure the computational costs can be greatly reduced using a method called *dynammic programming* ([15]).

If an optimization problem has a series of sub problems and the optimal global solution consists of the optimal solutions of the sub problems individually then the problem is said to have optimal substructure. For example, route finding problems are likely to exhibit optimal substructure. Say one is looking for the shortest path between to points that passes through a third i.e the optimal path is $\{x_1, x_2, x_3\}$. If the shortest path between the endpoint and the intermediate point is the subsection of the optimal path i.e $\{x_1, x_2\}$ then there is optimal substructure.

Any problem that has optimal substructure can be solved using dynamic programming. Essentially this is a method of problem solving where each sub problem is solved once, the solution value stored and recalled as needed in the subsequent problem.

Evaluating the conditional price bounds exhibits optimal substructure and dynamic programming can be applied. To see how we must recast the price bounds in a different form.

Given a discrete market $\mathcal{M} = \mathcal{S} \times \mathcal{H}$ and $k \geq 0$, $S \in \mathcal{S}$, $j \geq k$ consider the set,

$$\mathcal{H}^j_{\mathcal{S}_{(S,k)}} \equiv \{h_j = H_j(S) : H \in \mathcal{H},\ S \in \mathcal{S}_{(S,k)}\}.$$

Additionally for convenience denote $(S_{i+1} - S_i)$ by $\Delta_i S$. Using Lemma 1.6.1 from [?] we can show how under certain features of $\mathcal{H}$ and $\mathcal{H}^j_{\mathcal{S}_{(S,k)}}$ the relate conditional bounds to one another.

$$\overline{V}_k(S, Z, \mathcal{M}) = \inf_{H \in \mathcal{H}} \Big\{ \sup_{\tilde{S} \in \mathcal{S}_{(S,k)}} \Big\{ Z(\tilde{S}) - \sum_{i=k}^{N_H(S)-1} H_i(\tilde{S})(\Delta_i \tilde{S}) \Big\} \Big\}$$

$$\overline{V}_k(S, Z, \mathcal{M}) = \inf_{H \in \mathcal{H}} \Big\{ \sup_{\tilde{S} \in \mathcal{S}_{(S,k)}} \Big\{ \sup_{\hat{S} \in \mathcal{S}_{(\tilde{S},k+1)}} \Big\{ Z(\hat{S}) - \sum_{i=k+1}^{N_H(S)-1} H_i(\hat{S})(\Delta_i \hat{S}) \Big\} - H_k(\tilde{S}_k)(\Delta_k \tilde{S}) \Big\} \Big\}$$

from Lemma 1.6.1 from ([?])

$$\overline{V}_k(S, Z, \mathcal{M}) = \inf_{H_k : H \in \mathcal{H}} \Big\{ \sup_{\tilde{S} \in \mathcal{S}_{(S,k)}} \Big\{ \inf_{(h_{k+1},\ldots) \in \bigcup_{j \geq k+1} \mathcal{H}^j} \Big\{ \sup_{\hat{S} \in \mathcal{S}_{(\tilde{S},k+1)}} \Big\{ Z(\hat{S}) - \sum_{i=k+1}^{N_H(S)-1} h_i(\Delta_i \hat{S}) \Big\}$$
$$- H_k(\tilde{S}_k)(\Delta_k \tilde{S}) \Big\} \Big\} \Big\}$$

Under appropriate hypotheses on $\mathcal{H}$ we can prove,

$$\overline{V}_k(S, Z, \mathcal{M}) = \inf_{H_k : H \in \mathcal{H}} \Big\{ \sup_{\tilde{S} \in \mathcal{S}_{(S,k)}} \Big\{ \overline{V}_{k+1}(\tilde{S}, Z, \mathcal{M}) - H_k(\tilde{S}_k)(\Delta_k \tilde{S}) \Big\} \Big\}$$

Admittedly we cannot trivially make these assumptions on $\mathcal{H}$. In order to show this equivalence in our setting we need to define some dynamic bounds similar to the ones in the last line above and then show that they are in fact equivalent to the global conditional bounds. For a more robust argument see appendix A which outlines the dynamic programming philosophy more specifically for our approach using arguments from [16].

Each conditional bound can be redefined as being dependent on the value of another bound at a later node along the trajectory. Rather than evaluating the bounds by considering every trajectory separately we can take advantage of the nested nature of the trajectory set and see its optimal substructure. Beginning with the terminal nodes in a trajectory set, which will simply take a value given by Z. We can then move backwards through the trajectory set using each previously solved sub problem to eventually evaluate $\overline{V}_0(S, Z, \mathcal{M})$. This will give us in financial terms the total starting capital required to upper hedge our pay outs.

3.2.2 Solving the Dynamic bounds: The Convex Hull Algorithm

Evaluating the following:

$$\overline{V}_k(S, Z, \mathcal{M}) = \inf_{H_k : H \in \mathcal{H}} \left\{ \sup_{\tilde{S} \in \mathcal{S}_{(S,k)}} \left\{ \overline{V}_{k+1}(\tilde{S}, Z, \mathcal{M}) - H_k(\tilde{S}_k)(\Delta_k \tilde{S}) \right\} \right\}$$

,

is not itself trivial. While a more robust proof is discussed in Appendix B from [16]. Here we will briefly discus the intuition behind the argument as it is relevant to analysing the implementation.

Assume we already have the optimal hedging value h^* that gives the infimum above. Then we naturally have:

$$\overline{V}_k(S, Z, \mathcal{M}) \geq \overline{V}_{k+1}(\tilde{S}, Z, \mathcal{M}) - h^*(\Delta_k \tilde{S}) \ , \forall \tilde{S} \in \mathcal{S}_{(S,k)}$$

assume S^* is the trajectory in $\mathcal{S}_{(S,k)}$ where we achieve the supremum

$$\Rightarrow \overline{V}_{k+1}(S^*, Z, \mathcal{M}) - h^*(\Delta_k S^*) \geq \overline{V}_{k+1}(\tilde{S}, Z, \mathcal{M}) - h^*(\Delta_k \tilde{S}) \ , \forall \tilde{S} \in \mathcal{S}_{(S,k)}$$

$$\frac{\overline{V}_{k+1}(S^*, Z, \mathcal{M}) - \overline{V}_{k+1}(\tilde{S}, Z, \mathcal{M})}{S^*_{k+1} - \tilde{S}_{k+1}} \geq h^*$$

So then by searching over all possible S_{k+1} we can find a upper bound on the hedging value. It is easy to picture this upper bound as the slope line under which all possible $\overline{V}_{k+1}(S, Z, \mathcal{M})$ lie below in $(S_{k+1}, \overline{V}_{k+1})$ space. If it were any lower, there would be a $\overline{V}_{k+1}(S, Z, \mathcal{M})$ over this line and it would no longer be the optimal value. So then intuitively the lower bound on $\frac{\overline{V}_{k+1}(S^*,Z,\mathcal{M})-\overline{V}_{k+1}(\tilde{S},Z,\mathcal{M})}{S^*_{k+1}-\tilde{S}_{k+1}}$ gives us the optimal value $h*$ and this is the result more or less from Appendix B:

$$\overline{V}_i(\mathbf{S}, Z, \mathcal{M}) = \sup_{S^\bullet, S^\circ} \{\overline{V}_{i+1}(\mathbf{S}^\bullet, Z, \mathcal{M}) - u_{(\mathbf{S}^\bullet, \mathbf{S}^\circ)} \Delta_i S^\bullet\}$$

Where $\mathbf{S}^\bullet$ and $\mathbf{S}^\circ$ are trajectories in $\mathcal{S}_{(S,k)}$ for which S_{k+1} is greater than S_k or less than S_k respectively.

3.3 Grid Construction and Pricing

Now that we have redefined the trajectory set in a way that is more amenable to work with directly we will briefly discuss the details of implementation before analysis. For a full description see the code involved see Appendix C.

The procedure for pricing a payoff is done in two steps first a adjacency matrix is constructed representing the edges of the graph theoretic trajectory set ($|E|$), and a second array called the `index` which essentially represents the set of nodes ($|V|$). The `index` stores the association between the node in the matrix and the (k, j, n) value.

The second step given the adjacency matrix and index is to move backwards through the trajectory set from the terminal nodes and apply the convex hull algorithm iteratively.

3.4 Performance Analysis

The limiting factor in implementation is the performance of the grid construction in time and space. Being a rather large data structure, memory concerns do come into play.

3.4.1 Complexity of Grid Construction

Given that you are at a node (k, j, n) the branching ratio is the maximum possible number of children nodes defined by $b = \sup_{\mathbf{S}}\{|CS|\}$. Where CS was defined in (3.1). Starting at an initial parent point at most you expect there to be b edges extending to children nodes, as well as at most b nodes for their children. Given that that each node will have fixed upper limit on the number of operations required to calculate the coordinates of its children we then expect the time performance to be limited by the upper limit of the edges in the graph. i.e we expect $\mathcal{O}(|E|)$. Given that $|E| \leq b^{N_2} \simeq b^{\frac{K}{q}}$ (for large K) we can see that the relative size of the conditional set and expiry time will be directly responsible for the time performance of grid constriction.

Space Complexity of grid construction Naively, like in most graphs we would expect the space complexity to also be $\mathcal{O}(|E|)$ however there is an added complication in our algorithm as the graph is being constructed iteratively using an `index` without foreknowledge of the eventual set. Therefore the `index` needs to contain information on how to potentially convert between the integer grid and the column/row number in the adjacency matrix. In this way we have that the space complexity become $\mathcal{O}|IG| \simeq (2K_{max} + 1)(J_{max})(N_{max})$

This is of great concern for implementing the algorithm as is means considerable memory must be put aside for grid construction before proceeding.

The values J_{max} and N_{max} connect to our aforementioned model parameters in the following way:

$$J_{max} = \frac{\sup\{Q_T\}}{\delta_o^2} \; N_{max} = \frac{K}{\Delta}$$

And the K_{max} which will be the maximum range of the stock values over $[0, T]$ in units of δ_o can be given by the largest possible local movement in units of δ_o times the number of possible time steps:

$$K_{max} = \overline{m} J_{max} = \overline{m} \frac{\sup\{Q_T\}}{\delta_o^2}$$

Impact on modelling It is noteworthy that the time complexity was dependent more on the local parameters of the model (namely $\underline{q}$ and the size of the conditional sets) whereas the space complexity was more dependent on the global parameters.

In the next chapter we will look at the local vs. global pictures in the context of modelling. An important concern then is designing models for which the parameters of the model relevant to performance do not produce results that are too computationally intense to find. To what degree would the models be relevant if the results took longer than an options lifetime to compute? Additionally we will see as an example in model 2 in chapter 4 that the space complexity concerns will forcefully dictate arbitrary choices of model parameters.

Chapter 4

SELECTED MODELS

In Chapter 2 the Main Trajectory model was constructed in a general fashion without specifying the model any further than the existence of a conditional set that locally determined the dynamics of trajectories. We then went on to define some local and global features of these trajectory sets and explored the different ways they interacted. Combined with the details discussed in the last chapter regarding implementation, we seek to construct more specific, easily implemented models.

Model 1 is defined via a conditional set with reference to the local parameters $\overline{m}, \underline{m}, \overline{q}, \underline{q}$ introduced in (2.11) and (2.10). Values of $\overline{m}, \underline{m}, \overline{q}, \underline{q}$ that better match unfolding charts to trajectories will likely produce wider price bounds and be harder to implement. Conversely, models that have tighter price bounds will have more difficulty matching unfolding charts but be easier to implement. This tension is not easy to negotiate but by understanding the connection between local and global properties of trajectory sets it is possible to do so. Using the insights from this model we will be able to recast the local/global dichotomy in financially more familiar risk/reward terminology.

Model 2 will be constructed by directly observing the local and global features of past trajectories and incorporating them into a coherent model. As we will see this can produce more accurate prices than the first model while not sacrificing the ability to match trajectories to unfolding charts. The reader should note that there is no issue of over fitting as we are defining future trajectories in a combinatorial way. That is to say all possibilities are allowed given the constraints.

The following selected models all share the simplifying assumption that sampling times and rebalancing times are one and the same. Mathematically this means that $s^{min} = s^{max} = 1$ in all models. This was done for simplicity but also so that the model's assumptions can be more easily visualized by looking at the conditional integer set. We will however discuss the implications of this further at a later point.

4.0.2 Data Employed

The two sources of data for this demonstration were:

- A 6 month long stretch of hourly tick data for Facebook Inc (FB). At the end of this period, call option ask prices were captured across a variety of strikes for comparison with an expiration 9 days in the future.
- A 6 month long stretch of hourly tick data for Biogen Inc (BIIB). At the end of this period, call option ask prices were captured across a variety of strikes for comparison with an expiration 15 days in the future.

4.1 Model 1: Worst Case Parameter Estimation and Risk

Investor/Chart assumptions:

(1) $s^{min} = s^{max} = 1$

(2) $\delta = \delta_o = 1$ recall $z = \frac{\delta}{\delta_o} = 1$

(3) $\tau = T$

Model Hypotheses, $\mathcal{A}$:

(1) $\exists\ m^{max}, m^{min}, q^{max}, q^{min} \in \mathbb{Z}$

(2) If $(m, q) \in \mathcal{N}_o^{\mathcal{A}}$ then:

$m^{max} \geq m \geq z$

$m^{min} \leq m \leq -z$

$q^{min} \leq q \leq q^{max}$

The first model is the simplest. The conditional set, given some extreme values, is simply the Cartesian product of a pair of ranges i.e $\mathcal{N}_o^{\mathcal{A}}(S_i, W_i, T_i) = \mathcal{N}_o^{\mathcal{A}} = ([m^{min}, -z] \cup [z, m^{max}]) \times [q^{min}, q^{max}]$.

4.1.1 Connections Between Investor/Chart Assumptions and Model Parameters

Before continuing into the model details it is worth pausing to reflect on the connections between the subjective investor/chart assumptions in our models δ and δ_o and the model parameters $m^{max}, m^{min}, q^{max}, q^{min}$. In the interests of simplicity, we assumed $\delta_o = 1$ as this is a close enough approximation to values in our charts and greatly simplifies the trajectory sets and aids in comparison and demonstration.

Recall the subjective assumption on the investor, δ is the minimum amount of movement in the market required before our investor will consider rebalancing. The freedom to choose δ is extremely useful. Our choice of δ allows us to control the size of the conditional set without sacrificing potential trajectory matching opportunities. The reason for this is that a larger δ will mean smaller movements will be ignored, meaning our trajectory set does not need to include them.

We can see this in the Figures (4.1) and (4.2). A larger δ means that a longer period of time can pass before sampling, whereas the size of movements we include grows rather slowly with the m^{max} parameter. This intuitively makes sense as it will take most stocks a longer time to move past a greater δ.

By increasing δ we can include larger movements and ignore smaller ones. In more mercurial markets where periods of relative calm can follow large price movements increasing δ will be extremely useful. This is because we can ignore the calmer periods effectively and efficiently. While it is true that increasing δ will increase the size of the conditional set-a detail to take into account in implementation for very practical computational reasons- this is of no great concern if the timespan over which you are pricing are not prohibitively long.

Notice too that choosing a δ too large will cause our conditional set to collapse as there will be δ large enough that we get only one sampling time and we will get $q^{max} = q^{min}$ and $m^{max} = m^{min}$. Increasing δ further means that we will get no points at all. This and the fact that we will usually have a time horizon $T = K\Delta$ over which we will want some $N_1, N_2 \propto \frac{1}{\underline{q}}$ will place a upper limit on our choice of δ.

4.1.2 Parameter Selection

The key idea behind selecting our parameters $m^{max}, m^{min}, q^{max}, q^{min}$ or populating our conditional set generally is the hope of producing easy to implement models that will contain trajectories that we can map unfolding charts onto with confidence that the necessary trajectories will exist.

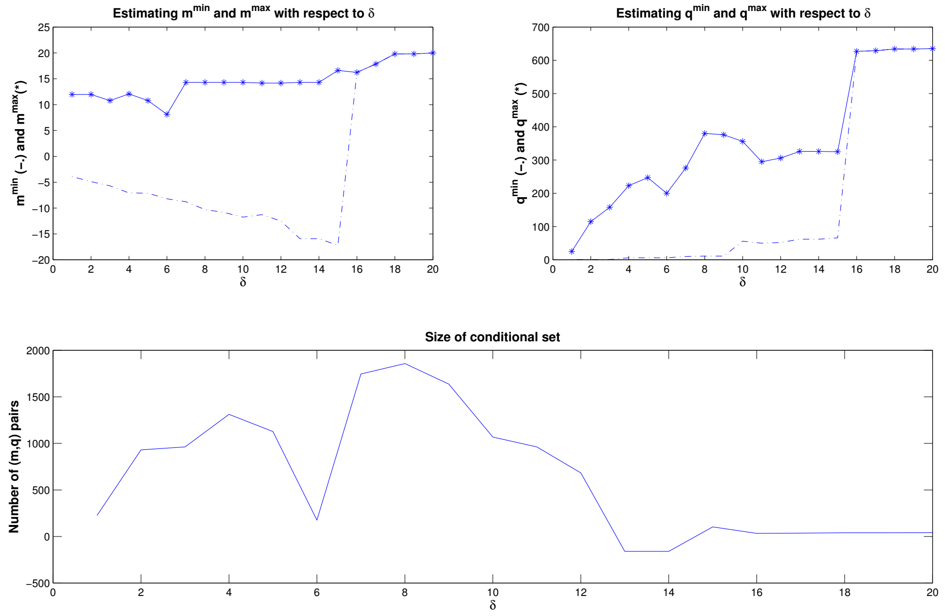


Figure 4.1: Connections between δ and model parameters, FB data set

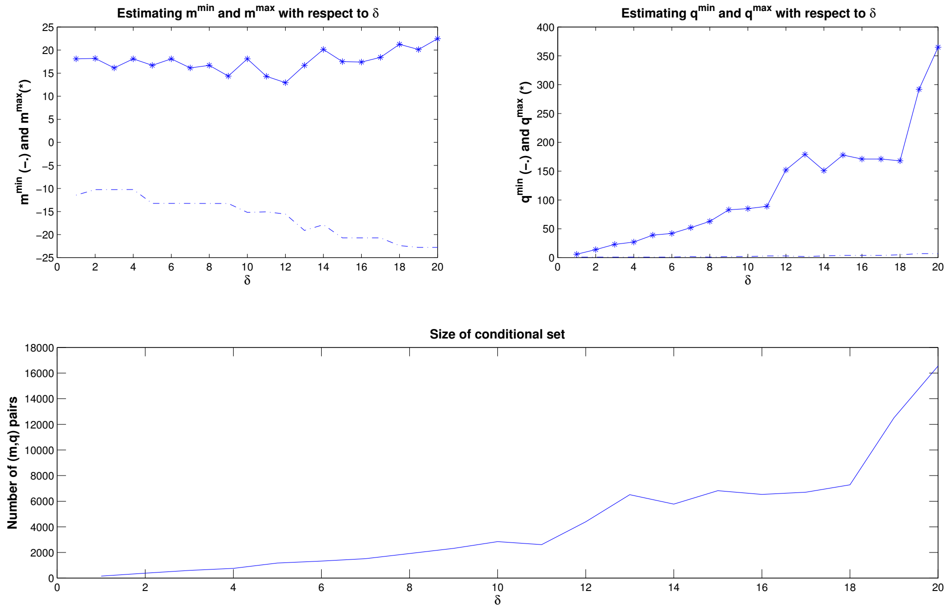


Figure 4.2: Connections between δ and model parameters, BIIB data set

Obviously the larger the m^{max} and q^{max} parameters we choose, hence the larger conditional set, the more likely it is that trajectory matching is possible. However as discussed earlier, for convex payoffs (that is most vanilla options), the larger our parameters the greater the size of our price bounds. This is not by itself a problem. That said, if we could be confident in our abilities to match trajectories it would be desirable to produce tight bounds that highlight arbitrage opportunities or at the very least are useful to pricing.

To that end we will discuss two important proposed methods of parameter selection. The first and most tangible is to observe past data and select the worst case values of $m^{max}, m^{min}, q^{max}, q^{min}$. The second idea is the that of an arbitrary $m^{max}, m^{min}, q^{max}, q^{min}$ within the range of the worst case parameter selection. This second proposal is done with the naive expectation that a less than worst case parameter selection invites risk in exchange for potential profits. We will examine this later claim in more detail.

4.1.2.1 Worst Case Parameter Calibration

Under our assumptions it is easy to make worst case parameter estimations from past chart data. Given a chart $x(t)$ and a choice of δ we get sampling times $\{r_l\}$ that depend on our choice of δ. Then for each $(x(r_l, w(x, r_l), r_l)$ and $(x(r_{l+1}, w(x, r_{l+1}), r_{l+1})$ we can see that a possible value for (m, q) that would match this chart is given by $(\frac{x(r_{l+1})-x(r_l)}{\delta_o}, \frac{r_{l+1}-r_l}{\Delta})$

So then if we wanted from the data a more robust "worst case" estimate we would find the extreme values over the whole set of data:

$$
\begin{aligned}
m^{max} &= \sup_{\{r_l\}}\{\frac{x(r_{l+1}) - x(r_l)}{\delta_o}\}, \\
m^{min} &= \inf_{\{r_l\}}\{\frac{x(r_{l+1}) - x(r_l)}{\delta_o}\},
\end{aligned}
$$

and similarly

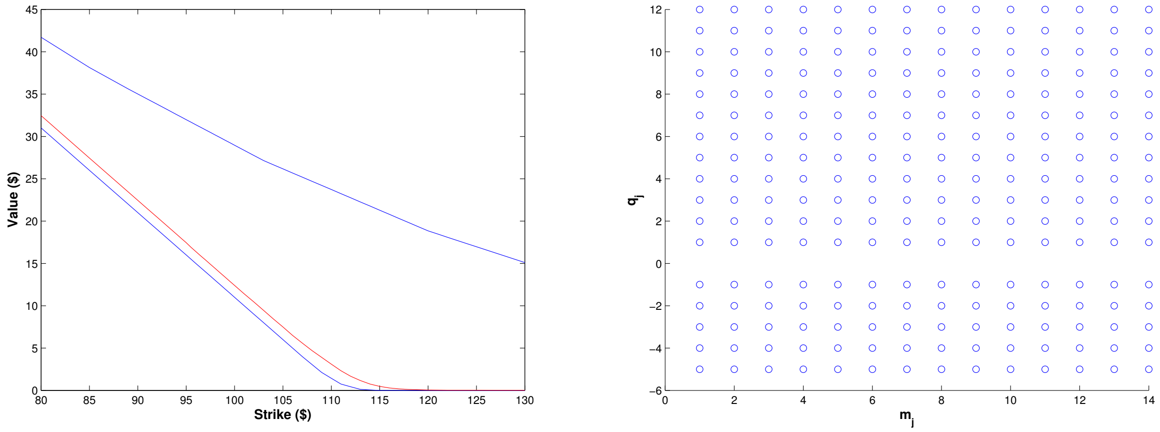
$$
\begin{aligned}
q^{max} &= \sup_{\{r_l\}}\{\frac{r_{l+1} - r_l}{\Delta}\}, \\
q^{min} &= \inf_{\{r_l\}}\{\frac{r_{l+1} - r_l}{\Delta}\}.
\end{aligned}
$$

The above definitions are historically estimated from chart data. The reader should not confuse the above definitions with the ones introduced in Section 2.5.1 which were defined as worst case parameters for a given trajectory set.

The worst case estimation as defined will be very robust in matching future unfolding trajectories. Failure to match the unfolding chart to trajectories will only happen if the combined time and stock movements of the unfolding chart are more extreme than any in the past data (in our case, 6 months). The main flaws are that we have some very extreme price movements leading to very high ranges for Q_{t_i} over the option lifetime and correspondingly very large price bounds. This is problematic as the bounds will be unlikely to be informative for option pricing or profitable hedging strategies. See Figures (4.3) and (4.4).

4.1.2.2 Speculative Parameter Estimation

Keeping in mind the fact that the worst case bounds will most likely be useless we can foreshadow the risk/reward tradeoff that exists when choosing model parameters and the tension we described at the beginning of the section. Looking ahead to Figure 4.12 at the observed conditional set, this is the set of all (m, q) observed in the data. We may decide stock movements of more than $\pm 2\delta$ are comparatively rare (In this case there exists a $\sim 10\%$ chance given past data of larger movements).



(a) Worst case price bounds against FB market ask prices

(b) The Conditional Set

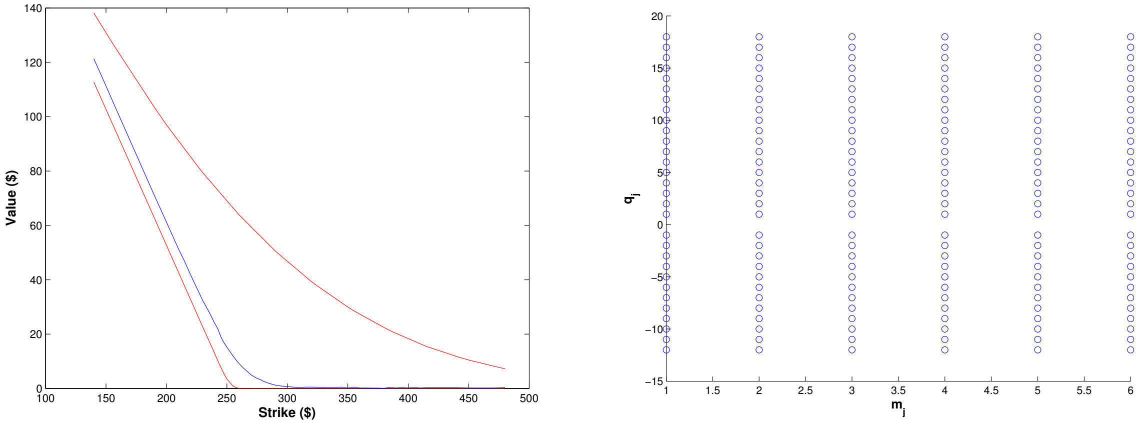
Figure 4.3: Worst case parameter estimation and prices for FB data

We may speculate that over the rest of the lifetime of the option on FB that events where the stock moves more than $\pm 2\delta$ will not occur and select $m^{max} = 2$ as a parameter value, all else equal to the worst case estimate.

Recall that the hedging price bounds give the initial value of a hedging portfolio. When looking at the price bounds in the $m^{max} = 2$ case (Figure 4.5 (a), (c), in blue) we see that the bounds are much tighter to the point of arbitrage opportunities. In fact if we look at the real unfolding trajectory over the lifetime of the option we see it matches closely and yields hedging profits where we would expect (black line in Figure 4.5 (a)). This is expected as the unfolding chart was properly matched to a trajectory, hence the assumptions the price bounds were calculated under were true.

However, we picked a segment of the same length from the past and it is easy to show that even though the trajectory set eventually begins approximating this simulated trajectory well. The initial jump and corresponding violation of the model's hypothesis is enough to create a loss. See Figure 4.5 (c) (d).

Modifying the local features of trajectory sets (i.e the conditional set) has effects on the potential



(a) Worst case price bounds against BIIB market ask prices

(b) The Conditional Set

Figure 4.4: Worst case parameter estimation and prices for BIIB data

to match trajectories going forward, hence means taking on risk of violating a the hypotheses defining the conditional set. However by modifying the local features in this way we can get tighter price bounds where if the model hypotheses defining the conditional set hold, we will see profits.

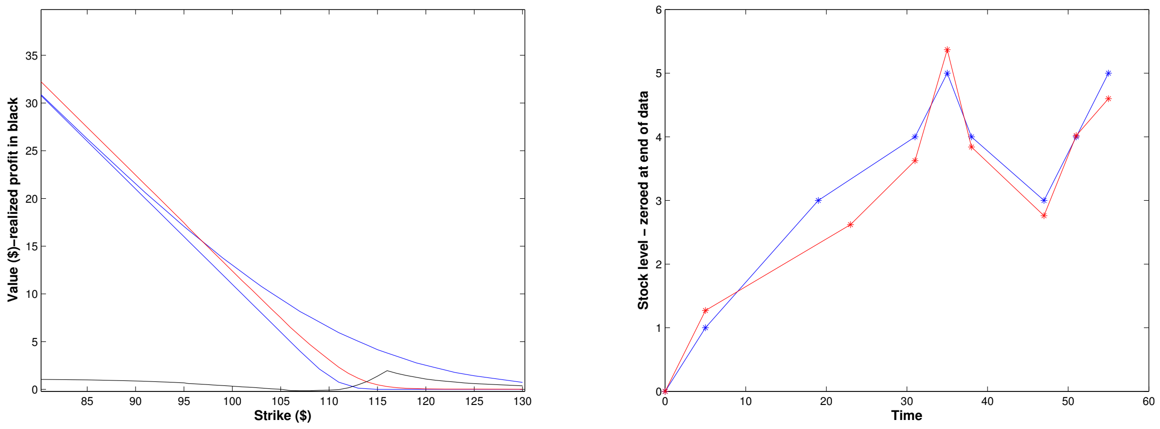
4.1.2.3 Model Discussion: Risk vs. Reward

In order to build on the insights of the last section we will define risk as it is used here.

Risk is defined as the probability, understood here as a frequency on past occurrences, that no trajectory in a model's trajectory set matches an unfolding chart. We will estimate this by examining past data and seeing where model hypotheses break down. Expressed more precisely we will look at past chart data and estimate the probability that the model fails to agree with the chart in the sense laid out in Proposition 2.1.

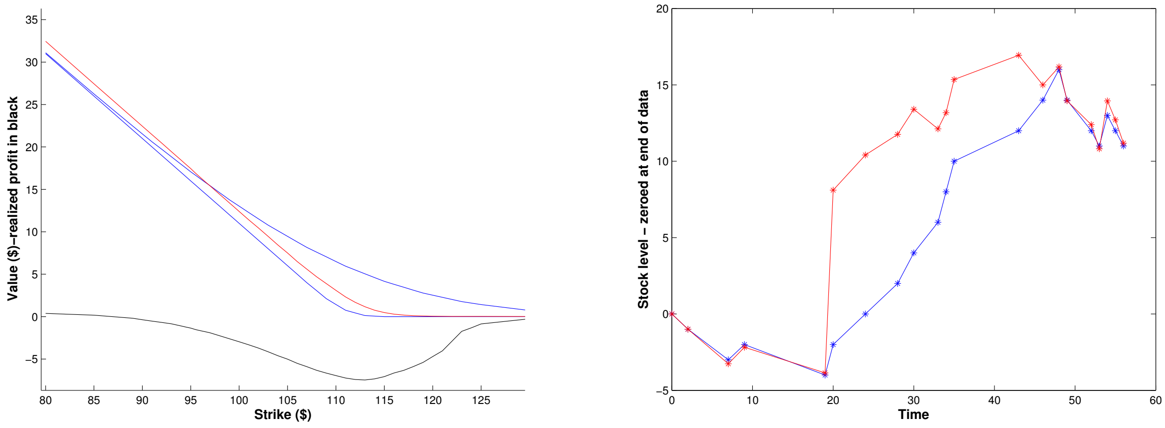
In Model 1 there are two ways to violate the hypotheses:

(1) Hypothesis violation by the time restriction: The trajectory fails to move δ before q^{max} time steps.



(a) Price bounds and ask prices with modified parameters. Hedging profit from real trajectory in black

(b) Comparison with real unfolding trajectory, fairly close to model trajectories



(c) Price bounds and ask prices with modified parameters. Hedging profit from trajectory in in black

(d) Comparison with simulated trajectory, failure to match to model trajectory after 5 re hedging times

Figure 4.5: Model 1 with FB data where $m^{max} = 2$, $m^{min} = -2$

If an unfolding chart violates the model hypothesis in this way the price bounds for a convex payoff are actually unaffected. This may seem surprising until we recall the alternative definition of the price bounds referenced from Appendix B earlier.

Say we are considering the upper conditional price bound at some node (S_i, W_i, T_i) that is $\overline{V}_i(\tilde{S}, Z, \mathcal{M})$, $\tilde{S} \in \mathcal{S}_{(S,i)}$. Let $S^+ \in \mathcal{S}_{(S,i-1)}$ where $S_i^+ \geq S_i$ and likewise $S^- \in \mathcal{S}_{(S,i-1)}$ where

$S_i^- \leq S_i$

Recall that if Z is convex then subsequent conditional bounds will be as well. That is:

$$\frac{\overline{V}_i(S^+, Z, \mathcal{M}) - \overline{V}_i(S^-, Z, \mathcal{M})}{S_i^+ - S_i^-} \geq \frac{\overline{V}_i(S^+, Z, \mathcal{M}) - \overline{V}_i(S, Z, \mathcal{M})}{S_i^+ - S_i}.$$

So we can show:

$$\overline{V}_i(S, Z, \mathcal{M}) \geq \overline{V}_i(S^+, Z, \mathcal{M}) - \frac{\overline{V}_i(S^+, Z, \mathcal{M}) - \overline{V}_i(S^-, Z, \mathcal{M})}{S_i^+ - S_i^-}(S_i^+ - S_i)$$

$$\overline{V}_i(S, Z, \mathcal{M}) \geq \sup_{S^+, S^-} \{\overline{V}_i(S^+, Z, \mathcal{M}) - \frac{\overline{V}_i(S^+, Z, \mathcal{M}) - \overline{V}_i(S^-, Z, \mathcal{M})}{S_i^+ - S_i^-}(S_i^+ - S_i)\}$$

$$\overline{V}_i(S, Z, \mathcal{M}) \geq \overline{V}_{i+1}(S, Z, \mathcal{M}).$$

Therefore, when pricing backwards through the grid subsequent price bounds will grow larger in value. Intuitively this makes sense, as the further from expiry the more initial capital we will require to upper hedge. The most important thing to note is that the bounds entirely depend on the stock level. So if we are at some later time holding the stock level constant the closest nodes in time will dominate the supremum above.

While the bounds themselves will not change an unfolding chart that falls outside the trajectory set will make trajectory matching impossible and force us to take on risk. See Figure 4.6.

As we will see this carries as much risk as that the next case with no potential profits and is the reason why these two sources of risk should be considered separately. As changing the q^{max} parameter does nothing to create investment opportunities and instead simply makes it harder to match trajectories.

(2) Hypothesis violation by the stock restriction given by m^{max}: We have already seen in the last section the second hypothesis to violate is if the unfolding chart jumps higher or lower than

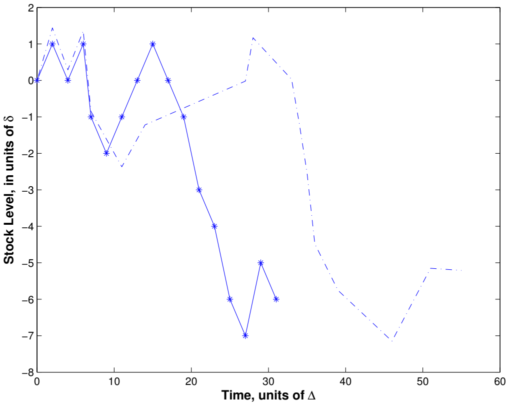
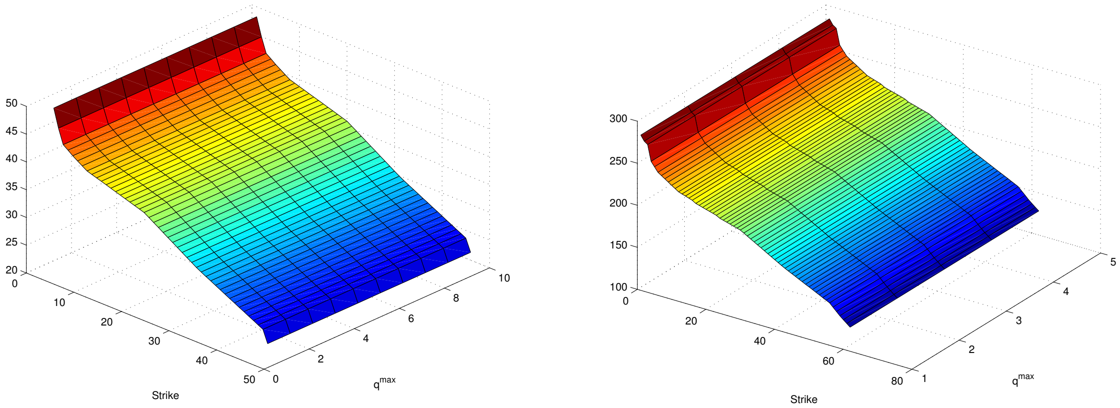


Figure 4.6: Example from FB data of a failure to match trajectories due to q^{min} too small. Unfolding chart is dotted, model trajectory is starred.

any trajectories we would match to it. See example already included in Figure 4.5. Changing the m^{max} parameter lowers the price bounds (Figure 4.8). This makes sense considering how the price bounds are evaluated via the convex hull. Essentially decreasing the m^{max} has the effect of shrinking the range of possible values of the trajectories $(S^{\bullet}, S^{\circ})$ and decreasing the possible values of the bound (as the sup of a smaller set is less than or equal to the original). The effect is more pronounced at higher strikes as it means fewer nodes in the trajectory set have a non-zero payoff to super hedge.

As the conditional bounds are designed to give us enough starting capital to upper hedge the payout of a derivative, then the potential profit if we are below the market asking price is proportional to the difference of our starting capital. This is done by selling at the market price and hedging against the payout with some of the proceeds of the sale, pocketing the difference. Naively we would expect taking on more risk will lead to higher potential payouts and this is what we more or less observe in Figure 4.11.

Figure 4.11 was constructed by varying the parameters $m^{max}, m^{min}, q^{max}, q^{min}$ over some ranges in combination and observing in the past data the probability of the model failing to match charts



(a) FB data

(b) BIIB data

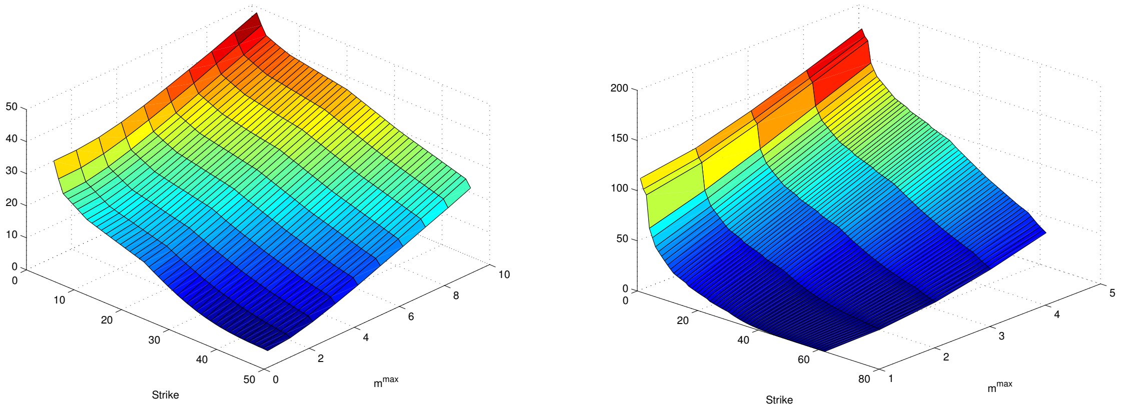
Figure 4.7: Upper hedging bounds with varying q^{min} and strike holding m^{max} constant

i.e the risk. We then compared the supper hedging price bound at different strikes to the market asking price. If the asking price was greater, we then calculated the difference of our bounds and the asking price to estimate the expected return.

An interesting feature of note in the risk/reward profiles in Figure 4.11 is that there are points at equal reward levels with different risk values. This is reflective of the ways our model hypothesis can be violated in the first sense i.e. hypothesis violation by the time restriction. Recall in that case that the model would fail to match trajectories for some values of q^{max} but would have the same price bounds as models where the value of q^{max} would be more likely to match trajectories.

However as we will discuss this is an incomplete analysis without a better understanding of how large potential losses can be (that is, the actual impact of the hypothesis being violated).

This model has made it possible to introduce a risk/reward or local/global framework as well as explore some of the features of the price bounds and trajectory matching. That said it was a rather crude model based on very simple assumptions. We were only able to tighten the price bounds by introducing risk. We would like to make more accurate bounds this while maintaining our ability to match trajectories separate from the goal of searching for profit. This is achieved in Model 2



(a) FB data

(b) BIIB data

Figure 4.8: Upper hedging bounds with varying m^{min} and strike holding q^{max} constant

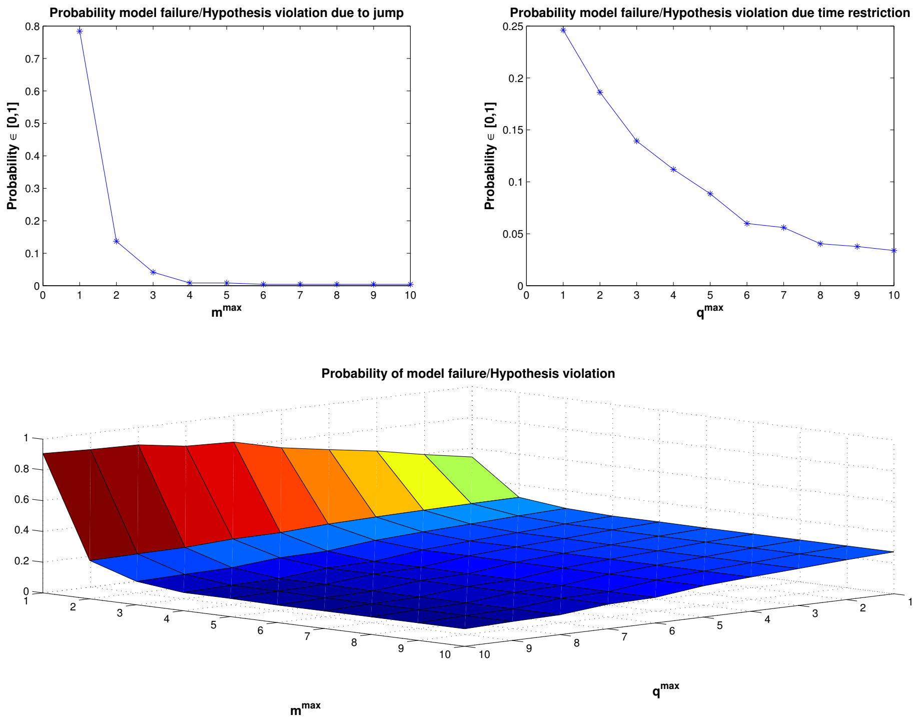


Figure 4.9: Risk estimation for FB Data set: individual and joint probability of failure to match unfolding charts to trajectories given model parameters

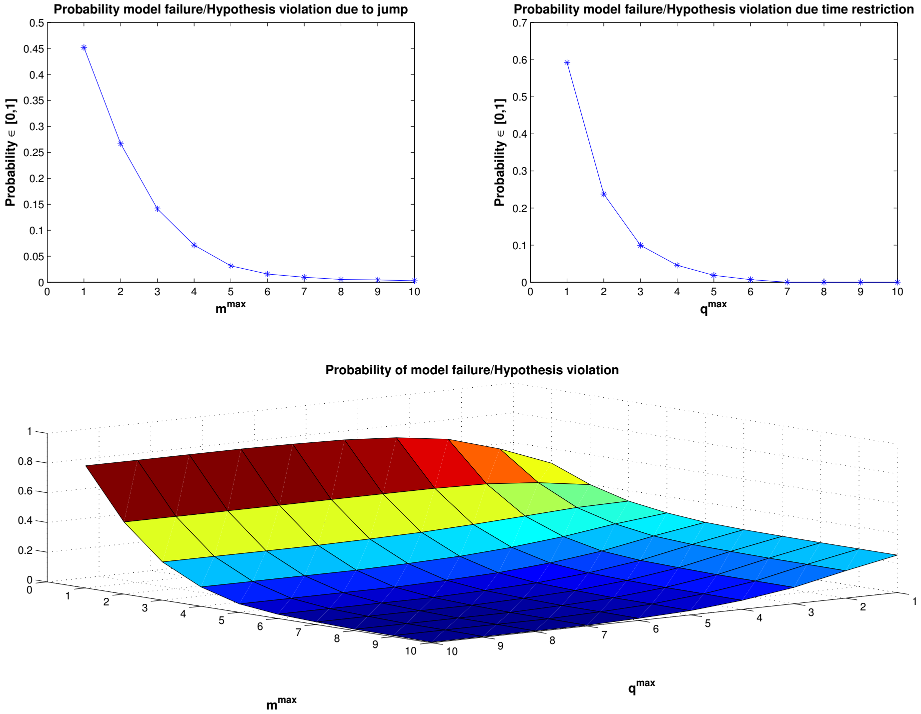
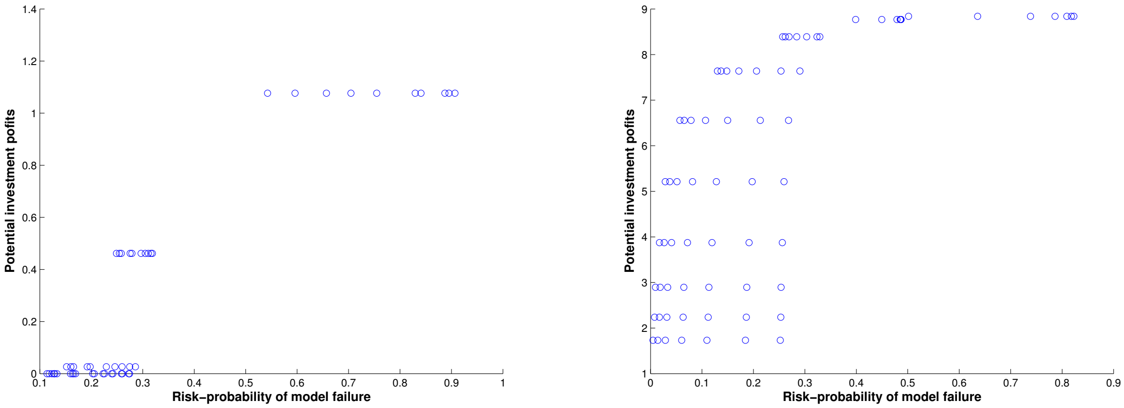


Figure 4.10: Risk estimation for BIIB Data set: individual and joint probability of failure to match unfolding charts to trajectories given model parameters

next.



(a) FB data

(b) BIIB data

Figure 4.11: Risk and reward profiles in both data sets

4.2 Model 2: Observed Trajectory Features

In this model we will not focus on the risk reward machinery of the first model, but rather simply explore another way the local/global dichotomy allows us to build better models.

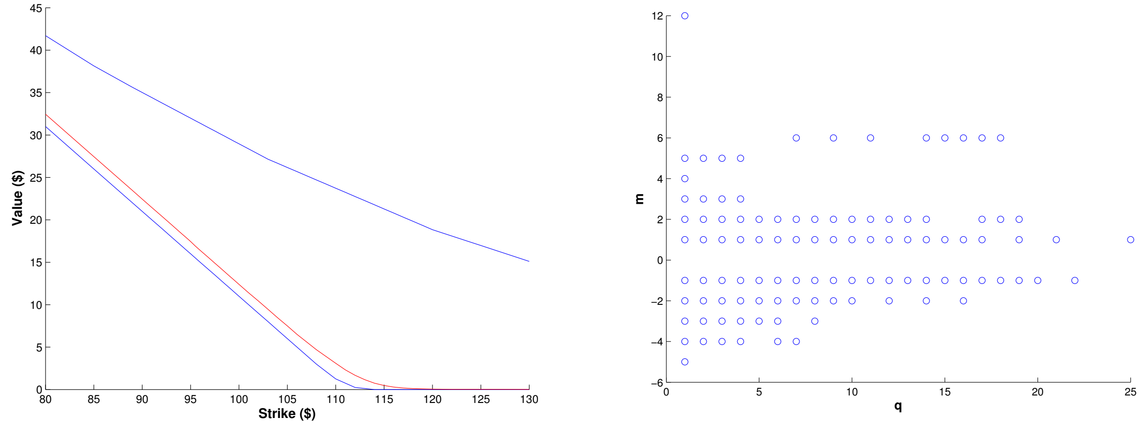
4.2.1 Model 2a: Local Observations

Initial model Hypothesis:

Given a past observed trajectory $x(t)$ and a corresponding set of sampling times $\{r_l\}$, we model the conditional set from past observations.

Through Model 1 we highlighted how one can take on risk by shrinking the conditional set. Where there are more possibilities locally (i.e a larger conditional set) there is less risk as it is far less likely that the model hypotheses will be violated and unfolding charts will match trajectories in the set, it is also true that the resulting bounds will be larger as a result.

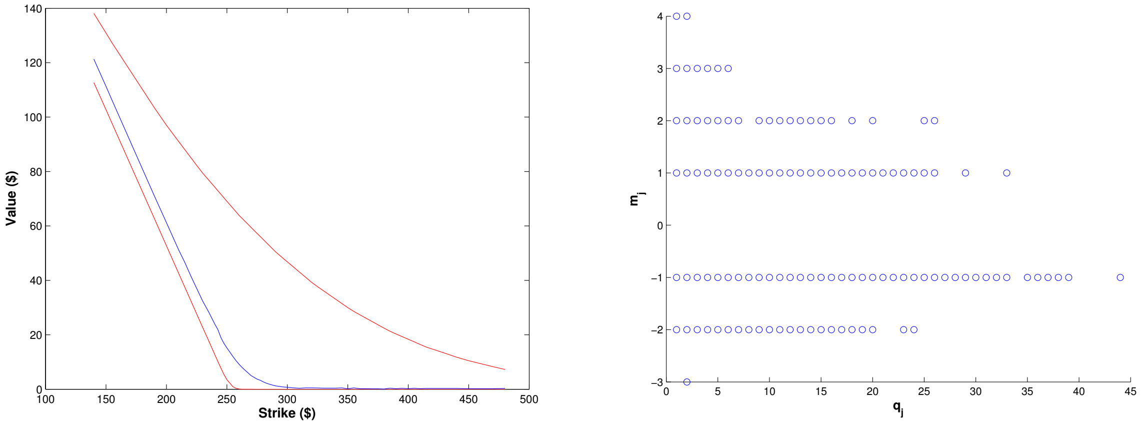
Model 2 tries to negotiate this in a new way, rather than developing local hypotheses we simply observe the past and reconstruct a conditional set based on past chart movements. We do this by looking at the observed chart starting at every available point in our past data. The local dynamics will be fairly robust as there would need to be an hourly chart movement that was completely unobserved in a relative sense over a 6 month period. We can see as well, while there are some extreme movements, they are limited in number. See Figures 4.12 ans 4.13.



(a) Price Bounds using observed conditional set

(b) Observed Conditional set

Figure 4.12: Model 2a with FB data



(a) Price Bounds using observed conditional set

(b) Observed Conditional set

Figure 4.13: Model 2a with BIIB data

The resulting bounds are still rather large, no better than Model 1 in the worst case. In fact they are the exact same for the same reason that changing q^{max} had no effect on the bounds. The observed set is the same for pricing as the worst case conditional set from Model 1 with some points deleted at later times.

The reason for this for example in the FB data is the $(12, 1)$ and $(-5, 1)$ elements of the conditional set. Those extreme price movements are being repeated at every possible node and dominate the price bounds when using the convex hull method to evaluate the price bounds.

It would be desirable in that case then to impose constraints on the global quadratic variation to eliminate such extreme scenarios from happening repeatedly while maintaining robust hypotheses for charts.

4.2.2 Model 2b: Local and Global Observations

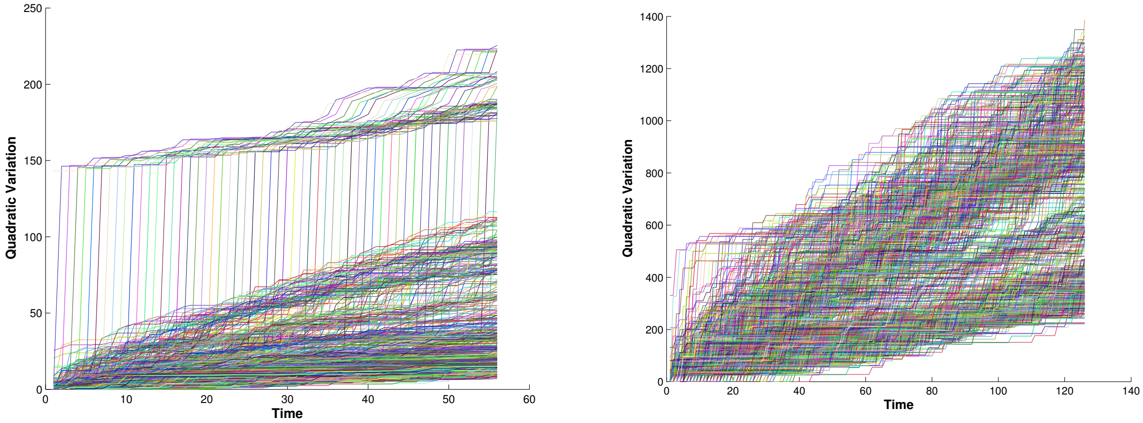
Model Hypothesis:

(1) Given a past observed trajectory $x(t)$ and a corresponding set of sampling times $\{r_l\}$, we model the conditional set from past observations

(2) Given $S \in \mathcal{S}^{\mathcal{W}}$ we construct from past observations and a node $(S_i, W_i, T_i) \in S$ then $W_i \in Q_{t_i}$. Where Q_{t_i} is the range of the quadratic variation taken from past intervals the same length as the option lifetime.

Note: As in Model 1 we assumed that $\delta = \delta_0 = 1$ for the FB data set. However the maximum quadratic variation measured in units of δ_o^2 in the grid was too large for the BIIB data set for $\delta_o = 1$-basically we ran out of memory creating the grid (see the analysis in chapter 3 on space complexity of grid construction). For this reason we took $\delta_o = 5$ to avoid this memory issue. While not of theoretical relevance to the model it is worth noting that under certain parameters, Model 2 is very difficult to implement.

Model 2b uses the same observed conditional set as Model 2a we limit the number of times extreme events can occur along a trajectory. This is achieved by imposing a *global* constraint on the possible quadratic variation through time of the type introduced in Chapter 2 Section 2.5.2.2.



(a) FB data

(b) BIIB data

Figure 4.14: Observed quadratic variation of observed data over option lifetime

See Figures 4.14. Observing the past from various starting points we looked at the the resulting quadratic variation over time. Note that in the FB data there is a jump in many past charts that seems to split the resulting behaviour of the charts into two groups. Those are the ones that contained the extreme event in the conditional set previously pointed out, the $(12, 1)$ event.

We defined some sets Q_{t_i} with this in mind. We defined

$$Q_{t_i} = [Q_o(t_i), Q_1(t_i)] \cup [Q_1(t_i), Q_2(t_i)]$$

Where each $Q(t)$ function (displayed in Figure 4.15) crudely defines the allowed W_i values given T_i. These restrictions were fairly easy to implement during grid construction. We simply wrote an `if` statement in the model file.

For the BIIB data which did not contain any jumps that were extreme enough to separate the quadratic variation over time into disjoint regions. We simply restricted the quadratic variation to the values that had been observed in the past at that time. Put another way we defined some:

$$Q_{t_i} = [Q_o(t_i), Q_1(t_i)]$$

Where $Q_o(t_i)$ was the minimum value observed at t_i and $Q_1(t_i)$ the maximum.

The more interesting case however was the FB data set. The $Q_n(t_i)$ functions we defined in this case approximately enclosed the possible quadratic variation in the trajectory set into two regions through time. One region would allow the extreme jump once, and the lower one would enclose the rest. See Figure 4.15.

In both data sets,the results were the tightest bounds of our examples so far. Refer to Figures 4.16 and 4.17. This is to be expected as we are imposing a global restriction on quadratic variation. However our model hypotheses are still relatively robust at the local level. In order for this model to be violated we need an even more extreme price movement, or another to occur multiple times in a very short time span that was more extreme than one observed in the past.

This model and the others demonstrates that there is not a clear efficient trade-off between risk and reward when considering local and global information separately. But that by incorporating global information into the local dynamics of trajectories one can get closer.

The main drawback of this model however is that the hypothesis are too minimal to firmly explore the risk/reward profile more fully. Aside from restricting Q_{t_i} to ahistorical levels which would undermine the model's core philosophy of using observed quantities.

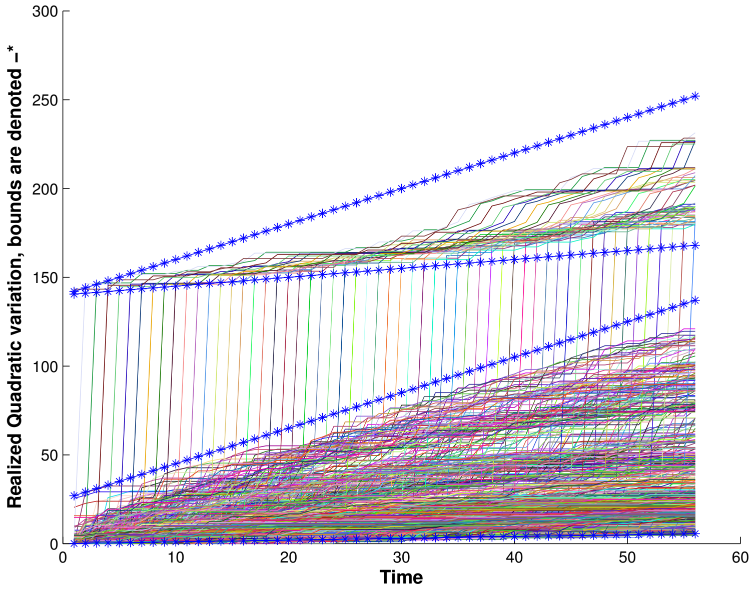


Figure 4.15: Variation with bounds for FB, allowed variance is within the two triangular regions

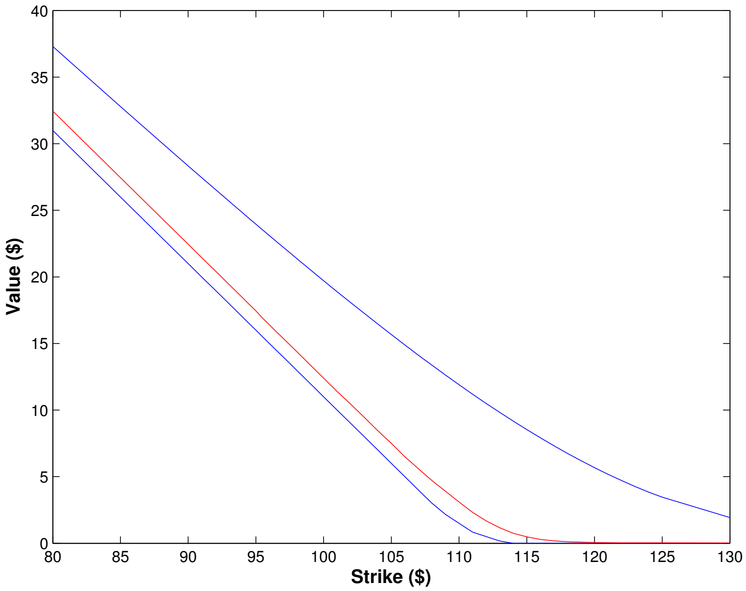


Figure 4.16: Model 2b bounds for FB

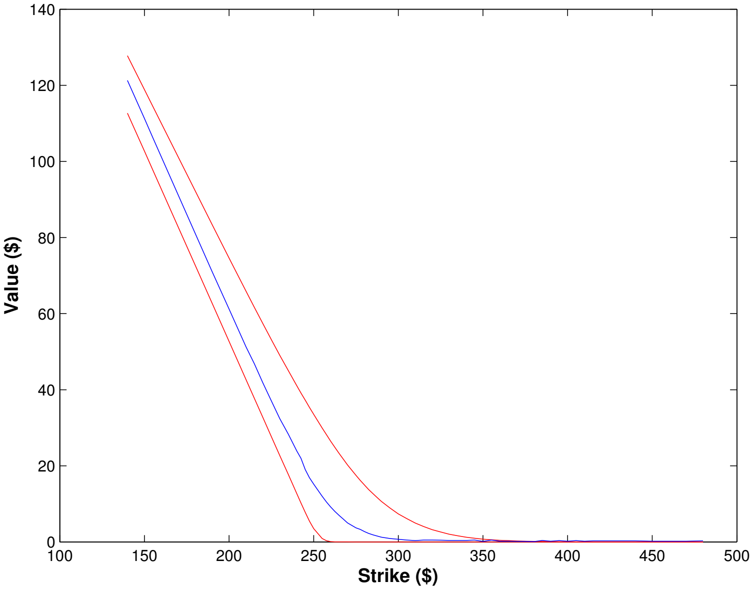


Figure 4.17: Model 2b bounds for BIIB

4.3 General Discussion of Models and Conclusion

4.3.1 The Rebalancing (s) Parameter

For simplicity we set $s = 1$, the reason was primarily for the purposes of easily visualizing the conditional set without any complication. However the same effect could have been achieved with a non unitary s and the Integer Conditional Set (3.1) in Chapter 3.

Increasing s mainly would affect the local parameter $\overline{q}$ and the global parameters N_1 and N_2 (2.5.2.1). These are practical considerations that would need to be taken into account for a potential investor wanting to use these models to actually set up a hedging portfolio.[1]

4.3.2 Conclusion

The models introduced here show that it is possible to implement in a practical way the approach described in [8]. We have introduced a framework for model construction and an implementation paradigm that is relatively easy to use. It is our hope that the next step can be taken and this trajectory based approach can attempt to remedy the model risk of standard Mathematical Finance approaches described at the beginning of this thesis.

We suggest the following approaches be tried. It would be desirable, we think, to incorporate a probabilistic model in the modelling of losses in the case of a chart failing to match a trajectory. As it stands now all estimates of loss are empirical and based in past occurrences-a stochastic model may be able to estimate the losses in novel and yet to be observed circumstances. We also suggest the addition of other model variables. Perhaps in order to model clustering or other phenomenon one can introduce an element of path dependency stronger than the kind modelled by the allowable regions in our second model.

---

[1] For more information, contact AFleck Inc.

APPENDIX A: DYNAMIC PROGRAMMING

In the following section we will need the following idea. We state them here without justification which can be found in [16] along with a much more detailed analysis.

Definition A.0.1 (n-Bounded Market)**.** Given a market $\mathcal{M}$ the mapping $M(\mathbf{S})$ defined as the point in a trajectory where the portfolio functions ceases to be non-zero i.e $N_H(\mathbf{S})$. A market $\mathcal{M} = \mathcal{S} \times \mathcal{H}$ is called n-bounded if there exists a constant n so that:

$$\sup_{\mathbf{S} \in \mathcal{S}} \quad M(\mathbf{S}) \leq n.$$

This definition is introduced by [16] as there is no explicit incorporation of time in their trajectory sets like ours. In our case we can define $M(\mathbf{S})$ as follows:

$$T_{M(\mathbf{S})} = T$$

The intuition remains the same however, there is some point along a trajectory where rebalancing stops and positions are liquidated to cover the payout of a contingent claim.

Dynamic Minmax Bounds

As discussed previously a direct evaluation of the conditional bounds is a daunting task. Additionally given the formulation of the problem it is not clear how to construct the hedging values $H_i(\mathbf{S})$, for a given payoff Z, by means of the unfolding path values $S_0, S_1, S_2, \ldots$

Consider next another pair of numbers namely $\underline{U}_0(S_0, Z, \mathcal{M})$ and $\overline{U}_0(S_0, Z, \mathcal{M})$. These new bounds are defined in a dynamic, or iterative, definition each instance involving a local optimization akin to solving a local sub problem as described earlier.

We discuss here part of the argument for why these Dynamic minmax bounds are equal to our globally defined conditional ones.

We will deal with n-bounded markets (again, equivalent to our assumption that every portfolio is liquidated on the expiration time T) that is for each $H \in \mathcal{H}$, it holds that $N_H(\mathbf{S}) = M(\mathbf{S}) = m$. We will later introduce further restrictions on the set of portfolios $\mathcal{H}$.

Definition A.0.2 (Dynamic Bounds)**.** Consider an n-bounded, discrete market $\mathcal{M}$; for a given function Z defined on $\mathcal{S}^{\mathcal{W}}$, any $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$, and $0 \leq i \leq n$ set

$$\overline{U}_i(\mathbf{S}, Z, \mathcal{M}) = \begin{cases} \inf\limits_{H \in \mathcal{H}} \sup\limits_{\mathbf{S}' \in \mathcal{S}^{\mathcal{W}}_{(\mathbf{S},i)}} [\overline{U}_{i+1}(\mathbf{S}', Z, \mathcal{M}) - H_i(\mathbf{S})\Delta_i S'] & \text{if} \quad 0 \leq i < M(\mathbf{S}) \\ Z(\mathbf{S}) & \text{if} \quad i = M(\mathbf{S}) \\ 0 & \text{if} \quad i > M(\mathbf{S}). \end{cases} \tag{1}$$

Also define $\underline{U}_i(\mathbf{S}, Z, \mathcal{M}) = -\overline{U}_i(\mathbf{S}, -Z, \mathcal{M})$.

Recall that we define by $I^k_{\mathbf{S}}$ the set of the portfolio ranges of $\mathbf{S}$ on the stage k, in other words

$$I^k_{\mathbf{S}} \equiv \{H_k(\mathbf{S}) : H \in \mathcal{H}\} \subseteq \mathbb{R}. \tag{2}$$

Thus we can rewrite the expression in (1) for $0 \leq k < M(\mathbf{S})$,

$$\overline{U}_k(\mathbf{S}, Z, \mathcal{M}) = \inf_{u \in I^k_{\mathbf{S}}} \sup_{\mathbf{S}' \in \mathcal{S}^{\mathcal{W}}_{(\mathbf{S},k)}} [\overline{U}_{k+1}(\mathbf{S}', Z, \mathcal{M}) - u \ \Delta_k S']. \tag{3}$$

Theorem A.1. For any function Z defined on a discrete n-bounded market $\mathcal{M} = \mathcal{S}^{\mathcal{W}} \times \mathcal{H}$, the following inequality holds:

$$\overline{U}_0(S_0, Z, \mathcal{M}) \leq \overline{V}(S_0, Z, \mathcal{M}), \tag{4}$$

and hence $\underline{U}_0(S_0, Z, \mathcal{M}) \geq \underline{V}(S_0, Z, \mathcal{M})$ is also valid.

The reverse inequality requires a great deal of machinery and an explicit description of the support for all the $H \in \mathcal{H}$. It is far outside the scope and interest of this thesis. More details can be found in [16].

APPENDIX B: CONVEX HULL ALGORITHM

This section outlines how to calculate the dynamic bounds $\overline{U}_i(\mathbf{S}, Z, \mathcal{M})$ introduced in the previous appendix. Reproduced here from [16].

Again considering a n-bounded discrete market $\mathcal{M} = \mathcal{S}^{\mathcal{W}} \times \mathcal{H}$. For $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$, and $0 < i < M(\mathbf{S})$ we are going to expand on a method introduced in [**?**], in order to resolve (3). For an arbitrary, but momentarily fixed, $\mathbf{S}' \in \mathcal{S}^{\mathcal{W}}_{(S,i)}$, set

$$\ell(x) = \overline{U}_{i+1}(\mathbf{S}', Z, \mathcal{M}) - u_i(S'_{i+1} - x),$$

i.e. the line in the plane, through the point $(S'_{i+1}, \overline{U}_{i+1}(\mathbf{S}', Z, \mathcal{M}))$ with slope u_i. Thus,

$$\overline{U}_{i+1}(\mathbf{S}', Z, \mathcal{M}) - u_i(S'_{i+1} - S_i)$$

is the intersection of ℓ with the vertical straight line $x = S_i$. Therefore, for each fixed $u_i \in I^i_{\mathbf{S}}$:

$$\sup_{\mathbf{S}' \in \mathcal{S}^{\mathcal{W}}_{(\mathbf{S},i)}} \left\{ \overline{U}_{i+1}(\mathbf{S}', Z, \mathcal{M}) - u_i(S'_{i+1} - S_i) \right\}$$

is the largest of the ordinates of the points of intersection between the straight lines ℓ and $x = S_i$. Then $\overline{U}_i(\mathbf{S}, Z, \mathcal{M})$ becomes the lowest value of these largest intersections.

To complete the geometric procedure, assume both of the following sets are non-empty,

$$\mathcal{S}^{\mathrm{do}}_{(\mathbf{S},i)} = \left\{ \mathbf{S}' \in \mathcal{S}^{\mathcal{W}}_{(\mathbf{S},i)} : S'_{i+1} \leq S_i \right\}, \quad \text{and} \quad \mathcal{S}^{\mathrm{up}}_{(\mathbf{S},i)} = \left\{ \mathbf{S}' \in \mathcal{S}^{\mathcal{W}}_{(\mathbf{S},i)} : S'_{i+1} > S_i \right\}.$$

For $\mathbf{S}^{\mathrm{up}} \in \mathcal{S}^{\mathrm{up}}_{(\mathbf{S},i)}$ and $\mathbf{S}^{\mathrm{do}} \in \mathcal{S}^{\mathrm{do}}_{(\mathbf{S},i)}$ denote by $u_{(\mathbf{S}^{\mathrm{up}},\mathbf{S}^{\mathrm{do}})}$ the slope of the straight line in the plane through the points $(S^{\mathrm{up}}_{i+1}, \overline{U}_{i+1}(\mathbf{S}^{\mathrm{up}}, Z, \mathcal{M}))$ and $(S^{\mathrm{do}}_{i+1}, \overline{U}_{i+1}(\mathbf{S}^{\mathrm{do}}, Z, \mathcal{M}))$:

$$u_{(\mathbf{S}^{\mathrm{up}},\mathbf{S}^{\mathrm{do}})} = \frac{\overline{U}_{i+1}(\mathbf{S}^{\mathrm{up}}, Z, \mathcal{M}) - \overline{U}_{i+1}(\mathbf{S}^{\mathrm{do}}, Z, \mathcal{M})}{S^{\mathrm{up}}_{i+1} - S^{\mathrm{do}}_{i+1}}.$$

Next we are going to show that

$$L_i(\mathbf{S}, Z, \mathcal{M}) \equiv \overline{U}_i(\mathbf{S}, Z, \mathcal{M}) = \sup_{\substack{\mathbf{S}^{\mathrm{up}} \in \mathcal{S}^{\mathrm{up}}_{(\mathbf{S},i)} \\ \mathbf{S}^{\mathrm{do}} \in \mathcal{S}^{\mathrm{do}}_{(\mathbf{S},i)}}} [\overline{U}_{i+1}(\mathbf{S}^{\mathrm{up}}, Z, \mathcal{M}) - u_{(\mathbf{S}^{\mathrm{up}},\mathbf{S}^{\mathrm{do}})} \Delta_i S^{\mathrm{up}}] \tag{5}$$

That is, $\overline{U}_i(\mathbf{S}, Z, \mathcal{M})$, is the largest intersection of the referred lines with the line $x = S_i$.

The next Theorem requires extra assumptions and presents a way to solve the optimization problem for $I^i_{\mathbf{S}} = \mathbb{R}$, which we assumed in this thesis. This assumption $I^i_{\mathbf{S}} = \mathbb{R}$ is a convenient way of guaranteeing $u_{(\mathbf{S}^{\mathrm{up}}, \mathbf{S}^{\mathrm{do}})} \in I^i_{\mathbf{S}}$.

Theorem B.2. Let $\mathcal{M} = \mathcal{S}^{\mathcal{W}} \times \mathcal{H}$ a n-bounded discrete market. If for any $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$, $I^i_{\mathbf{S}} = \mathbb{R}$, and either one the two following conditions for $\mathbf{S} \in \mathcal{S}^{\mathcal{W}}$ below hold,

(1) $L_i(\mathbf{S}, Z, \mathcal{M}) = \overline{U}_{i+1}(\mathbf{S}^{\bullet}, Z, \mathcal{M}) - u_{(\mathbf{S}^{\bullet}, \mathbf{S}^{\circ})} \Delta_i S^{\bullet}$ for some $\mathbf{S}^{\bullet} \in \mathcal{S}^{\mathrm{up}}_{(\mathbf{S},i)}$ and $\mathbf{S}^{\circ} \in \mathcal{S}^{\mathrm{do}}_{(\mathbf{S},i)}$.

(2) For any $\mathbf{S}' \in \mathcal{S}^{\mathcal{W}}_{(\mathbf{S},i)}$, $0 < a \leq |S'_{i+1} - S_i| \leq b$ (a and b may depend on $\mathbf{S}$).

Then,

$$\overline{U}_i(\mathbf{S}, Z, \mathcal{M}) = L_i(\mathbf{S}, Z, \mathcal{M}).$$

for proof details see [16].

APPENDIX C: MATLAB CODE

Pricing in the Trajectory Based Framework

Program 1: creatematrix

Plan of the program The program generates from a model of the type (insert equation #) a set of discrete trajectories of rebalancing times recursively construed from nodes in a predefined integer grid. These trajectories can naturally be represented as a directed graph. While MATLAB has little support for graphs as a data type, it is very easy to store an adjacency matrix containing the same information. The only added complication is how to assign labels to nodes $v_i, v_j \in V \subset \mathbf{Z}^n$ in the graph in such a way as to easily represent edges in the matrix $A = \{a_{i,j}\}$

The way this is done, depends on the dimensionality of the model. While this thesis only covers models of at most 3 dimensions per rebalancing node (stock level, realized quadratic variation, time) theoretically the code can handle more complicated models of higher dimension. The first dimension is always assumed to be the stock level and thus in the integer grid will take on negative and positive values in the range $[-K_{max}, K_{max}] \in \mathbb{Z}$. The second dimension is assumed to be in the range $[0, J_{max}] \in \mathbb{Z}$ and for models of 3 or more dimensions the dimensions will have the range in the grid $[0, N(i)] \in \mathbb{Z}$ where N is a vector of inputs. The way these points in the grid are mapped onto graph labels is simple. The point $(-K_{max}, 0, ..., 0)$ is assigned the value 1, then counting along the range of the first coordinate until the maximum before starting again with the adjacent dimension's next value. That is, the point $(K_{max}, 0, ..., 0)$ is assigned the value K_{max}. Then the process started again at $(-K_{max}, 1, ..., 0)$ with $K_{max} + 1$ and $(K_{max}, 1, ..., 0)$ with $2K_{max}$ and so on until $(K_{max}, J_{max}, ..., 0)$ with $2K_{max}J_{max} + 1$. This process is repeated for as many dimensions the model will need in the integer grid.

The program stores this information in an array called the index, where the rows in the array contain the tuples of the grid. The row value is the label in the graph. The program then finds the children of the initial node in the grid, finds the corresponding points in the index and stores this information in an adjacency matrix. This process is repeated for each child node and so on until the grid is full.

The inputs and index The program has 3 inputs: 2 integer arrays and one string. The first array `N0` contains the initial point from which the trajectories will be constructed in the grid (typically [0,0] or [0,0,0]). The second array `N`, contains the maximum values of the dimensions in the grid. For an input of $[K_{max}, J_{max}]$ it will create an integer grid of size $2K_{max}J_{max} + 1$ in $[-K_{max}, K_{max}] \times [0, J_{max}]$. With higher dimensions where $N = [N(1), ..., N(i)]$ it will create an integer grid of size $(2N(1)N(2) + 1)N(3)...N(i))$ in $[-N(1), N(2)] \times [0, N(2)] \times .... \times [0, N(i)]$

The last input is a string, corresponding to the name of a file in the same directory of the form:

```
function [n1]=function_name(n)


code where given node n, calculates array of children nodes n1


end
```

Where given a node in the integer grid, it calculates the possible children according to some model. For example, the BJN model with jump parameter p=2 (eq #) is given by:

```
function [n1]=BJNp2(n)

p=2;


k=n(1);
```

```
j=n(2);


n1=[];


mJ=permn([-p:-1,1:p],1);

mJsquared=mJ.^2;

l=length(mJ);


n1=[n1; mJ + k*ones(l,1) , mJsquared + j*ones(l,1) ];

n1=unique(n1,'rows','stable');


end
```

So given these inputs the program begins by creating some arrays of predetermined size. These will be used to store a sparse adjacency matrix representing the possible trajectories in our pricing grid and an index storing the relationship between a node label and its place in the grid. The last output, an array called here XY contains information for plotting the grids.

```
function [As,I,XY]=creatematrix(N0,N,model)


I=index(N);

Nmax=length(I);


As=sparse(Nmax,Nmax);


%makes coordinates for ploting
```

```
....
```

The function `index(N)` creates the index array where points in the grid are stored as rows and the index of a row in in the array represent the node label in the graph. This is done as mentioned before. For example:

```
>> I=index([1,1])


I(:,:,1) =           I(:,:,3) =          I(:,:,5) =


    -1     0           1     0             0     1


I(:,:,2) =           I(:,:,4) =           I(:,:,6) =


     0     0          -1     1             1     1
```

So then if $[-1, 1]$ is in one of our trajectories it will be labeled in the graph by the number 4.

Generating the Matrix Given these initial inputs and output arrays, the code then applies the model in an iterative fashion to each node and it's children,

```
....


l=length(I);
```

```
L=length(N)+1;

n1=index([N,0]);

for i=1:l

    if I(1,:,i)==N0

    o=i;

    break

    end

end


CHILD=feval(model,zeros(size(N0)));

INT=ones(size(CHILD));


n1(1,length(N)+1,o)=1;

newntest=index([N,0]);
```

[[Double brackets of the form [[...]] will enclose comments relating to the ideas behind program itself. At this point the code has only gone through some necessary preliminaries. It searched the index to find he node in the graph corresponding to the initial point for the trajectories. Then it found the children for said node.

Next it created two arrays similar to the index. n1 will store the parent nodes being considered and list a 1 or 0 in an additional column to the index to do so. newntest will do the same to store the children of said parent nodes. Then with each parent considered n1 will take on the values of newntest and so on]]

```
while any(n1(1,L,:)==1) )

%as long as there are parents
```

```
%whose children have yet to be determined


newntest(1,L,newntest(1,L,:)==1)=0; %clears newntest for a new pass


%un comment this in order to watch
%grid construction in real time
%%%%%%%%%%%%%%%
%         hold on; plottree(As,XY,'b')
%         drawnow


%loops over all the parents, in the first case this will just be i=o
for i=transpose(find(n1(1,L,:)==1))


        n=n1(1,1:L-1,i); %gives the points in the grid
```

[[Rather than using the model function repeatedly in a loop, we recognize that the array of the children at any point will be an affine transformation of the first node]]

```
        Icon=INT*diag(n)+CHILD;


        %add info to matrix:


        if size(Icon)==[1,length(N)]
            lIcon=1;
        else
            [lIcon,~]=size(Icon) ;
```

```
end


fnew=[]; %will store the node labels of the children


for ii=1:lIcon


 nprime=Icon(ii,:);


          if any(nprime>N)==0


          f=indexing(nprime,N);

          %function indexing maps points in the grid to labels

           %NB: index is needed to go from labels to grid points.


          if f<=Nmax   %to make sure the children are in the grid

          fnew=[fnew,f];

          newntest(1,L,f)=1;

          end


          end

end


As(i,fnew)=1; %store information that there is an edge from i->f
```

```
end


if n1==newntest %prevents infinite loop

break

end
```

[[The children nodes now become the new parents, at the top of the loop newntest is cleared and the process starts again]]

```
n1=newntest; %prepare for next pass


z1=max(n1(1,1,n1(1,L,:)==1));

z2=min(n1(1,2,n1(1,L,:)==1));


if isempty( [z1,z2] )==1 %stops the loop in specific cases that cause issues

    break

end


end

end
```

Program 2: Valueindex

Plan of the program The first program `creatematrix` outputs a graph in the form of an adjacency matrix `As` representing a trajectory space in the integer grid. The program `valueindex` gives a financial meaning to this data. Starting at the end points of the trajectories it maps onto the integer grid stock levels and calculates the option payoffs. Then it searches for the parents of nodes at the end of a trajectory, maps onto them stock levels, and performs the convex hull algorithm (see sec... ch 2) to calculate the hedging values. It then moves onto the parents of those nodes and repeats the process backwards through the trajectory set until the final initial node is reached, this is the dynamic programming methodology described in (see sec... ch 2). The result is a set of trajectories that specify hedging values and the initial capital at each point required to upper or lower hedge the payoff of a derivative. In order to store this information, the program adds 3 columns to the already existing index and stores the stock information and hedging values next to the points they belong to in the integer grid.

The inputs The program has seven inputs. The first two `A` and `I` are simple the adjacency matrix and index from the first program. The input `K` is the strike price if we are pricing options and `payoff` is a string corresponding to the name of a file in the same directory with that payoff function. For example look at the file `call.m`:

```
function [z]=call(S,K)


z=max(S-K,0);


end
```

The remaining inputs `s0`, `delta` and `nc` are closely related. Given the integer grid we produced in the last program we can map stock levels onto the grid in the following way:

```
function [S]=ktoS(k,delta,s0,nc)


if strcmp(nc,'linear')==1
```

```
    S=s0+k*delta;

elseif strcmp(nc,'exp')==1

    S=s0*exp(k*delta);

end
end
```

This function `ktoS` will be used in the rest of the program when assigning values to grid points. The input `nc` denotes the difference between a linear grid or an exponential one and `k` is the first coordinate of the point in the grid of interest.

The first thing the program `valueindex` does similar to program `creatematrix` is create arrays of predetermined size designed to store the outputs. It first looks at the index and adds 3 extra columns to store the output for the stock level, stock holdings (h in the models) and the cost of hedging $\overline{V}$ or $\underline{V}$

```
function [V]=valueindex(A,I,s0,K,delta,nc,payoff)

[r1,r2,r3]=size(I);

V=zeros(r1,r2+3,r3);

V(1,1:r2,:)=I;
```

...

Populating the value index using the grid matrix Next the program goes through the matrix and populates the last nodes of the grid. Calculating the stock level and option payoff. The last nodes are found relatively easily, just look at the nodes where the last coordinate is largest. In out models that would correspond to the maximum quadratic variation or time.

```
%fill last nodes (that are reached) with option payoffs


zrange=find(I(1,r2,:)==max(I(1,r2,:)));


for i=transpose(zrange)
```

[[This guarantees that the nodes we look at indeed have parents, that is there is a j so that at lease one `A(j,i)=1`]]

```
    if sum(A(:,i))>0


    k=I(1,1,i);
    Sk=ktoS(k,delta,s0,nc);
    z=feval(payoff,Sk,K);
    V(1,r2+1,i)=Sk;
    V(1,r2+2,i)=z
```

[[No need to worry about h in the last nodes, will have sold off holdings then]]

```
    %make sure we can't have negatives
    if Sk<=0
    V(1,3,i)=0;
    V(1,4,i)=0;
    end
```

[[Make negative nodes 0, will have the same effect as not even being there in the convex hull algorithm]]

```
    end
end


%price the rest


l=length(I);
last=length(zrange);


%look at nodes working backwards


 for r=l-last:-1:1 %bottom rows of matrix


     clear col
     col=find( A(r,:) );


     if size(col)>0 %make sure node is connected


     n=[];
     k=I(1,1,r);
```

```
Si=ktoS(k,delta,s0,nc);
V(1,r2+1,r)=Si;

%find its children
```

[[Given the way we are going backwards through the grid we are always guaranteed to have already priced any children nodes, as the labeling goes forward in the grid]]

```
for k=1:length(col) %over the different columns

if A(r,col(k))==1
    vS=V(1,r2+2,col(k));
    S =V(1,r2+1,col(k));

    n=[n;vS,S];

end
end

if size(n)==[1,2] %flat nodes keep the same value
V(1,4,r)=n(1);
else

[v,h]=convexhull(n,Si);

V(1,r2+2,r)=v;
```

```
        V(1,r2+3,r)=h;


        end


        end
    end



  end
```

REFERENCES

[1] Friedman, M. ”The methodology of positive economics.” *The philosophy of economics: An anthology 2* (1994): 180-213.

[2] Pfleiderer, P. ”Chameleons: The Misuse of Theoretical Models in Finance and Economics”. March 2014, Working Paper No. 3020

[3] Müller, Ulrich A., et al. ”Volatilities of different time resolutions—analyzing the dynamics of market components.” *Journal of Empirical Finance 4.2* (1997): 213-239.

[4] B.B. Mandelbrot. ”Scaling in financial prices: I. Tails and dependence”. *Quantitative Finance* Vol. 1 , Iss. 1,2001

[5] Mandelbrot, Benoit B., and Nassim Nicholas Taleb. ”Focusing on Those Risks That Matter.” *The Known, the Unknown, and the Unknowable in Financial Risk Management: Measurement and Theory Advancing Practice* (2010): 47.

[6] Taleb, Nassim N. ”Risk Neutral Option Pricing With Neither Dynamic Hedging nor Complete Markets.” arXiv preprint arXiv:1405.2609 (2014).

[7] P. Cheridito, M. Kupper and L. Tangpi (2016), *Duality formulas for robust pricing and hedging in discrete time*, arXiv:1602.06177v2 [q-fin.PR].

[8] S. E. Ferrando, A. Gonzalez, I. Degano and M. Rahsepar (2014). *Discrete, Non Probabilistic Market Models.*

Arbitrage and Pricing Intervals. Available at arXiv:1407.1769v3 [q-fin.MF], July, 2014.

[9] Goldstein D.G., Taleb N.N. (2007), *We don't quite know what we are talking about when we talk about volatility.* J. Portfolio Management, vol. 33, pp. 84-86.

[10] M. Nutz (2014). *Superreplication under model uncertainty in discrete time*, Finance and Stochastics, 791-803.

[11] Ingo Muller (2007). *A History of Thermodynamics The Doctrine of Energy and Entropy.* Springer Verlag.

[12] R. Rebonato (2004). *Volatility and Correlation: The Perfect Hedger and the Fox, 2nd Edition.* Wiley, New York.

[13] Duane. E. Roller (1961). *The early development of the concepts of temperature and heat*, Harvard University Press).

[14] Ya. A. Smorodinsky, *Temperature.* MIR 1984.

[15] Sniedovich, M. (1991). *Dynamic Programming: Foundations and Principles, Second Edition*, CRC Pure and Applied Mathematics, **(Book 154)**

[16] I. Degano, S. E. Ferrando and A. Gonzalez (2014). *Trajectory based models. Evaluation of minmax pricing bounds.* Available at `arXiv:1511.01207v1 [q-fin.MF]`, November 4, 2015.