Forecasting Odds Movements in Horse Racing

Master Thesis in Mathematics-Economics

Martin G. R. Maillard

Supervisor: Esben Høg

2020 June

Department of Mathematical Sciences
Aalborg University

This page is intentionally left blank.

School of Engineering and Science
Mathematics-Economics
Aalborg University
`http://www.ses.aau.dk`

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Forecasting Odds Movements in Horse Racing

Theme:
Master Thesis

Project Period:
Summer Semester 2020

Project Group:
1.213a

Participant(s):
Martin G. R. Maillard

Supervisor(s):
Esben Høg

Copies: 4

Page Numbers: 58

Date of Completion:
2020-06-02

Abstract:

In this thesis we consider the odds movements in horse racing on a betting exchange. Due to the odds structure on a betting exchange we choose to model the odds movements as the tick increments rather than the change in odds. This leads us to construct the odds trajectory model. Since the parameters in the odds trajectory model are not time-varying we explore an alternative state-space model, namely the dynamic Skellam model. The dynamic Skellam model is a nonlinear non-Gaussian model and this presents some challenges in filtering and smoothing the data. As a smoothing method we maximize the likelihood function where the evaluation of the likelihood function is done by using importance sampling method. The filtering of the data also utilizes importance sampling more specifically we use a particle filter, namely the bootstrap filter. Lastly, we forecast the odds movements by combining the estimated filtering values with the odds trajectory model.

Preface

This master thesis of the Master's program in the Mathematics- Economics degree at the Department of Mathematical Sciences at Aalborg University. It is compiled during the summer semester of 2020.

The thesis is composed of numbered chapters with corresponding sections. Citations and external references are listed in brackets with the author and year of publication noted, mainly at the beginning of each section. References to equations within the project are in parentheses with the first number indicating the chapter and the second number indicating the equation within that chapter. References to theorems, definitions, examples, etc. are indicated by two numbers, where the first is the chapter and the second is the number of the reference within that chapter. Certain proofs are omitted due to the scope of this project.

All code implementation in this project is done with R, [R Core Team, 2019]. The scripts used in the modeling part of the thesis can be shared upon a request to the author.

Furthermore, I would like to extend my greatest appreciation to my supervisor, Esben Høg.

Aalborg University, 2020-06-02

Martin Maillard

Martin G. R. Maillard

<mmaill15@student.aau.dk>

Contents

Introduction 1

When it comes to gambling, one sport gamblers are advised to avoid is horse racing. Horse racing is said to be the sport with most uncertainty due to the many variables that will influence a race. These variables do not only include the horse and the jockey but also the weather, the firmness of the ground, the trainer, the racing track, the distance, etc. So the likelihood of a bettor finding a profitable edge seems slim to none. In 2018 Bloomberg Businessweek published an interview with the gambler William Benter who managed to find an edge by building a statistical model. The motivation for William Benter was not the money, but to solve a challenge. In the interview they wrote "Benter wanted to conquer horse betting not because it was hard, but because it was said to be impossible"[1]. He managed to solve this challenge and yield a profit of nearly a billion US dollars. How did he find an edge? He calculated the win probability for each horse in a given race by using a multinomial logit model based on the work of Bolton and Chapman [1986]. The model assumed that the "utility" of a horse h can be measured by an existing function:

$$U_h = U(x_h, y_h),$$

where x_h is an attribute vector of the horse h and y_h is an attribute vector of the jockey riding the horse in a given race. Furthermore, they decomposed the function as:

$$U_h = V_h + \epsilon_h,$$

where V_h are deterministic components such as win rate, ground condition, trainer etc. and ϵ_h is a random component. By doing so they could model the probability for the horse h as:

$$P_{h^*} = \frac{\exp(V_{h^*})}{\sum_{h=1}^{H} \exp(V_h)}, \quad \text{for } h^* = 1, 2, \ldots, H, \tag{1.1}$$

where H is the total number of horsed in a given race. This was the foundation for William Benter's winning model. In 2008 he revealed in a report (see Benter [2008]) that he made significant improvements to the model by incorporating the public's odds. With the included public odds his probability estimate was calculated as:

$$c_{h^*} = \frac{\exp(\alpha f_{h^*} + \beta \pi_{h^*})}{\sum_{h=1}^{H} \exp(\alpha f_h + \beta \pi_h)}, \quad \text{for } h^* = 1, 2, \ldots, H,$$

where f_{h^*} is the log of the probability of the fundamental model, (1.1), π_{h^*} is the public's implied

[1] `https://www.bloomberg.com/news/features/2018-05-03/the-gambler-who-cracked-the-horse-racing-code`

probability estimate (this implied probability will be elaborated in greater details in Section 2.1) and c_{h^*} is the combined probability estimate.

Other attempts to model the winning probabilities of horses were done by building a ranking model for horse racing (see Ali [1998]). In a race with k horses the model is specified with $X_1, \ldots, X_k$ independent random variables and probability distribution functions $F(x; \alpha_i)$ for $i = 1, \ldots, k$. Let $\pi = (\pi_1, \ldots, \pi_k)$ serve as a permutation of k objects where π_j has rank j for $j = 1, \ldots, k$. The model determined the probability of the π permutation as:

$$P(\pi) = P(X_{\pi_1} < X_{\pi_2} < \cdots < X_{\pi_k}),$$

where X_i was a gamma distributed random variable. Thereby, the model was a gamma distributed ranking model. Unfortunately, the model did not lead to positive returns when they tested the model on an out-of-sample data set.

There have also been some studies done on the market efficiency within horse racing. Interestingly the markets are quite efficient, which can be the explanation that the public implied probability improved William Benter's model. However, it was found that there seems to be a long-odds bias in the public odds, meaning that high odds are overbought and therefore reflect a higher win probability than the empirical win probability (see Snyder [2008]). Another innovative approach to predict the winning horse before the race ends was done with in-running data. More precisely, the data was in-running market data from a betting exchanges (a betting exchange will be explained in greater details in Section 2.3) to forecast the winner (see Bunyan [2015]). By using the first 80% of the length of the race to fit a multiple linear regression model. This regression model was given as:

$$y \sim x_1 + x_2 + \cdots + x_N,$$

where y acts as the timestamp series and $x_1, x_2, \ldots, x_N$ represent the odds of the horses in the race. The intuition behind the model was that the winning horse will end on odds 1. Therefore, the coefficient of the winning horse should be most negative. Unfortunately, this approach did not yield good predictions for the last 20% of the race. But the idea of analyzing the odds movements from a betting exchange to forecast future odds was very intriguing.

In the literature, the main focus of researching horse racing has been efficiency and winning probabilities. In this thesis, we will take another approach, namely only focusing on the odds dynamics and how they move before a horse race begins. We formulate the following problem statement for this thesis.

1.1 Problem Statement

On the Betfair betting exchange the odds are determined by supply and demand. Furthermore, the betting exchange allows a bettor to "buy" and "sell" odds. Due to such a setup, the odds tend to move with respect to the current supply and demand in the market. We want to model these movements within the United Kingdom horse racing markets. More specifically, we wish to model odds movements for the favorite horse as a stochastic process. By developing such a model we will also forecast the odds movement at the starting time in a given horse race for the favorite horse.

2 Betting on Horses

In this chapter we will briefly clarify different betting markets for horse racing and the betting terminology that will be used throughout this thesis.

2.1 Odds & Bookmakers

When one places a bet the profit or loss is determined by the odds. We will now clarify this payoff structure. Sports betting odds differ from statistical odds. Even though they both reflect the chance of a win (or success). The fair odds o_i in sports betting are formulated as

$$o_i = \frac{1}{p_i},$$

where p_i is the probability of the event i occurring. The implied probability from the market odds are therefore given as $\widetilde{p}_i = \frac{1}{o_i}$. On the other hand odds in statistical modeling are usually given as $\frac{p}{1-p}$.

There are mainly two ways to represent odds, fractional odds and decimal odds. The relation between the two odds are given as: $(\textit{Fractional odds}+1) = \textit{Decimal odds}$, e.g. the odds $\frac{1}{1}$ and 2.00 give the same payout. The profit for a winning bet is calculated as $\textit{stake} \cdot (\textit{Decimal odds} - 1) = \textit{stake} \cdot \textit{Fractional odds}$ and a lost bet is simply $-\textit{stake}$. To put it simply, fractional odds reflect the net profit of a bet and decimal odds reflect the bookmaker's payout, i.e. the stake plus the profit. Throughout this thesis the odds are primarily represented as decimal odds. In theory the odds are the inverse probability of the chance to win. This, however, is not true in practice. The odds are set by a bookmaker and a bookmaker wants to make a profit no matter the result of an event. A simple event is a coin-flip with a fair coin. Then the bookmaker finds two people that place an equal-sized bet on heads and tails respectively. Since it is a fair coin the probability of heads and tails are both 50%, therefore odds should be $\frac{1}{0.5} = 2.00$. In this case the bookmaker would not make money in the long run since one of the bettor's loss pays for the other bettor's profit. This is not a good business model for a bookmaker, therefore the bookmakers add a *vig*. A vig is a small number added to the fair probability. If we add 1% to the fair probability of both outcomes the odds will change to $\frac{1}{0.51} \approx 1.96$ then two bets on £10 on each outcome will give a profit to the bookmaker. The bookmaker now gets to keep £10 from the person who lost the bet but has to pay $£10 \cdot (1.96 - 1) = £9.6$ to the person who won the bet. This is a total profit for the bookmaker of $£10 - £9.6 = £0.4$ no matter the outcome of the coin-flip. Now, this is a good business model! When one observes actual bookmaker odds it is clear that they add a vig. This can be determined by calculating the sum of all the implied probabilities. The total sum of the implied probabilities in one market is always above 100%. Mathematically this does not make any sense because it implies that there is more than 100% chance of any event happening in a market.

2.2 Betting Markets in Horse Racing

This section is based on information provide by Betfair [1].

There are mainly three markets to bet on in horse racing, namely: Win, Place and each way. We will now describe the rules for these three markets.

Win Market In the win market there is only one winner which is the first horse that crosses the finish line. It is often seen that the win market attracts the most liquidity. It is also the market that has the highest odds and therefore the highest potential payout of these three markets.

Place Market The place market bets on a given horse to place in the race. The payout is the same whether or not a given horse wins or places in the race. Unlike the win market, the place market is not always available for all horse races. The place market depends on how many runners there are in the race. For races with 1-4 runners there is only a win market. For non-handicap races with 5-7 horses the 2 fastest horses are placed and non-handicap races with eight or more horses the top three are placed.

If it is a handicap race then races with 16 or more horses the top four will be placed.

Each Way Market Each way bet is a combination of a win bet and a place bet. The bettor will place the same stake on a win and a place outcome. The odds on each way bets are a fixed fraction of win odds. This fraction is often $\frac{1}{4}$ of the win odds. For example if we have each way odds on 5.00, converting them to fractional odds we have $\frac{4}{1}$. The place odds are then fixed to the fractional $\frac{1}{4} \cdot \frac{4}{1} = \frac{1}{1}$ or equivalent to decimal odds of 2.00. We now make a $£5$ each way bet, meaning a win bet of $£5$ for odds 5.00 and a bet of $£5$ for odds 2.00 for the horse to place. There are three outcomes, we can win both bets, we only win the place bet or we lose both bets. If the horse wins the race the profit is $£5 \cdot (5-1) + £5 \cdot (2-1) = £25$. If the horse places but does not win we lose the win bet but we profit on the place bet. In this scenario we break even: $£-5 + £5 \cdot (2-1) = £0$. Lastly, if the horse does not place both bets are lost and a total loss is $£-10$. Each way bets can, therefore, be regarded as an insurance bet for the win market. Note that each way bets do not always break even if the horse places. If the win odds in the above

[1] `https://support.betfair.com/app/answers/detail/a_id/6489/`, last visited 2020-02-07

example were higher a placed horse that did not win would create a total positive profit. If the odds were lower a placed horse would lead to a total loss for the bets.

For the rest of this thesis, we only focus on the win market since this market exists for all UK horse races.

2.3 Betting Exchanges

An alternative to placing bets from a traditional bookmaker is by using a betting exchange. As the name suggests, a betting exchange is an exchange for odds. This means that the odds are determined through supply and demand. Naturally, this also means that there are two ways to place a bet i.e. buy or sell the odds. If we buy odds it is called a back bet. A back bet has the same liability as a bet at a regular bookmaker i.e. the liability is the *stake* and the profit is $stake \cdot (Decimal\ odds - 1)$. Therefore the seller of the odds will have the opposite payoff structure. If we sell the odds, this is also called a lay bet, our liability is now $-stake \cdot (Decimal\ odds - 1)$. The profit, on the other hand, is always fixed as the *stake*. By placing a lay bet we essentially act as a bookmaker. A lay bet on a horse makes it possible to profit as long as one of the other horses win.

The ability to buy and sell odds creates opportunities to hedge profit before a race ends and even before a race begins. Such profit is made by trading one's potential profits to a lower liability of a certain outcome. To clarify this we create a small example. Let us assume we back a horse at odds 3.45 for a stake at £100. Then our potential profit is $(3.45 - 1) \cdot £100 = £245$. If we want to trade this to a lower liability we need to make a lay bet for the same stake at odds less than 3.45. Therefore, if the odds move to 3.40 and we make a lay bet for £100 at these odds we have a liability for $£100 \cdot (3.40 - 1) = £240$. We can now calculate our profit or loss if the horse wins or does not win. If the horse wins we lose our lay bet and profit on our back bet, the total profit is now $£245 - £240 = £5$. On the other hand we break even if the horse does not win: $£ - 100 + £100 = 0£$. Essentially we have created at risk-free bet. From such a risk-free bet we can also hedge our profit, such that, no matter the outcome we will have a profit. The hedge bet is calculated as $(risk\ free\ profit)/\ (new\ odds)$. In this example, the hedge bet would be $£5/3.40 \approx £1.47$. With this additional bet we have a profit on £1.47 no matter the outcome. The calculations for the hedged profit is shown in Table 2.1.

	Win	Lose
Back bet	$(3.45 - 1) \cdot £100 = £245$	$£ - 100$
Lay bet	$-(3.40 - 1) \cdot £100 = £ - 240$	$£100$
Total	$£5$	0
Hedge (Lay)	$£5/3.40 \approx £1.47$	
Total w. hedge	$£5 + (-((3.40 - 1) \cdot £1.47)) = £1.47$	$£0 + £1.47$

Table 2.1: The table shows an example of a profitable hedged trade.

In the above example, the odds moves in our favor. In the case where the odds moves against us, i.e. we place a lay bet at higher odds than our back bet, we will lose £5 if the horse win. In this scenario, the hedge bet is a back bet and this will guarantee a small loss on either outcome. In summary, we can create profits if we can take advantage of the moving odds by placing a back bet when the odds are high and placing a lay bet when the odds are low. We have chosen to use the terminology "back" and "lay" bet instead of "buy" and "sell" odds simply to avoid using sentences like "we can make a profit by buying high and selling low". Using this kind of formulations can create confusion and from an economical point of view it makes no sense to profit by buying high and selling low.

On the Betfair exchange the odds can only take a finite amount of values. These values are fixed accordingly to how much they move for one tick. The odds movement per tick depends on how large or small the odds are[2]. The movement per tick in each odds interval are shown in Table 2.2.

From	To	Tick size
1.01	2	0.01
2	3	0.02
3	4	0.05
4	6	0.10
6	10	0.20
10	20	0.50
20	30	1.00
30	50	2.00
50	100	5.00
100	1000	10.00

Table 2.2: The table shows the increment of odds.

Due to these predetermined odds increments, the odds move in a discrete path. An example of such a path is shown in Figure 2.1. The data is from a race taking place in Southwell and the odds are for the horse *Smart Getaway* observed each minute 3 hours (or 180 minutes) before the race starts.

[2] `https://docs.developer.betfair.com/display/1smk3cen4v3lu3yomq5qye0ni/placeOrders`, last visited 2020-02-07

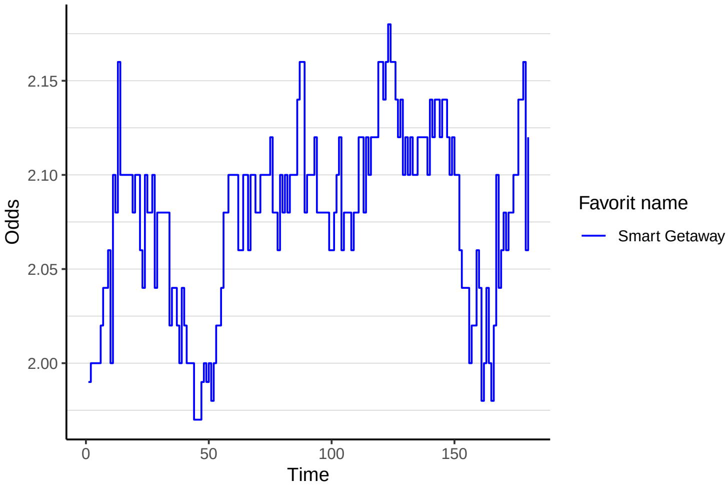


Figure 2.1: Pre race odds on a favorite. The odds are observed each minute 180 minutes before the race begins.

A betting exchange creates a whole new way of betting. It is no longer necessary to evaluate the winning probabilities, the fairness of the odds, or knowing every single detail of a sporting event. It is enough to find patterns in the odds trajectory and trade the odds movements. The motivation of this thesis is to model and forecast odds trajectories through quantitative methods.

3 Building Odds Trajectory

The goal of this chapter is to build a stochastic process that has the desired properties of odds trajectories from a betting exchange.

3.1 Compound Poisson Process

This section is based on [Pedersen, 2017] and [Tankov and Cont, 2004].

As seen previously, in Figure 2.1, the odds on a favorite only move in jumps. The most basic function of this type is a *càdlàg function* short for continue à droite limite à gauche which translate to right-continuous with left limits. It is defined as:

Definition 3.1 (Càdlàg function)
A function $f : [0,T] \to \mathbb{R}^d$ is said to be càdlàg if it is right-continuous with left limits: for each $t \in [0,T]$ the limits

$$f(t-) = \lim_{s \to t, s<t} f(s) \quad f(t+) = \lim_{s \to t, s>t} f(s)$$

exist and $f(t) = f(t+)$.

Càdlàg functions can be discontinue. If f is discontinue at point t we denote $\Delta f(t) = f(t) - f(t-)$ by a jump of f at t. A simple example of a càdlàg function is a piecewise constant step function. One of the most basic processes with càdlàg sample path is the Poisson process and we define it as:

Definition 3.2 (Poisson Process)
Let $\nu > 0$. $(N_t)_{t\geq 0}$ is called a Poisson process with parameter ν if:

(i) $N_0 = 0$ a.s.

(ii) $t \mapsto N_t$ is càdlàg, non-decreasing, and $\mathbb{N}_0$-valued for $\omega \in \Omega$.

(iii) $N_t - N_s \sim \text{Poi}(\nu(t-s))$ for $0 \leq s \leq t$.

(iv) $(N_t)_{t\geq 0}$ has independent increments.

The way the Poisson process is constructed is with a sequence of independent exponential random variables, $(\tau_i)_{i\geq 1}$, with parameter ν and the n'th jump have the value $T_n = \sum_{i=1}^{n} \tau_i$ i.e. the jump size is constant and equal to 1. Then the Poisson process is defined as $N_t = \sum_{n\geq 1} \mathbb{1}_{t\geq T_n}$, therefore, the Poisson process is a counting process that counts the number of random times, T_n, between 0 and t. Moreover N_t has the Markov property:

$$\forall t > s, \, \mathbb{E}\left[f(N_t)|N_u, u \leq s\right] = \mathbb{E}\left[f(N_t)|N_s\right].$$

This property follows from the independence of increments:

$$\begin{aligned}
\mathbb{E}\left[f(N_t)|N_u, u \leq s\right] &= \mathbb{E}\left[f(N_t - N_s + N_s)|N_u, u \leq s\right] \\
&= \mathbb{E}\left[f(N_t - N_s + N_s)|N_s\right],
\end{aligned}$$

since $N_t - N_s$ is independent of $N_u, u \leq s$.

We know that the odds can move up and down, therefore, we wish to make the jump size more flexible. For this reason we model the jump size as independent and identically distributed random variables. By doing so we create a compound Poisson process. Formally defined as:

Definition 3.3 (Compound Poisson Process)
Let $d \in \mathbb{N}$, and $(N_t)_{t\geq 0}$ denote a Poisson process with parameter ν. Let $(Z_n)_{n\geq 1}$ denote an i.i.d sequence of $\mathbb{R}^d$-valued random vectors independent of $(N_t)_{t\geq 0}$. An $\mathbb{R}^d$-valued process càdlàg process $(Y_t)_{t\geq 0}$ is called a compound Poisson process if, almost surely,

$$Y_t = \sum_{j=1}^{N_t} Z_j, \qquad \text{for } t \geq 0 \tag{3.1}$$

where we use the convention $\sum_{j=1}^{0} = 0$.

If we let $Z_n \sim N(0,1)$ and jump intensity $\nu = 5$ we can simulate two sample paths of a Poisson process and compound Poisson process.

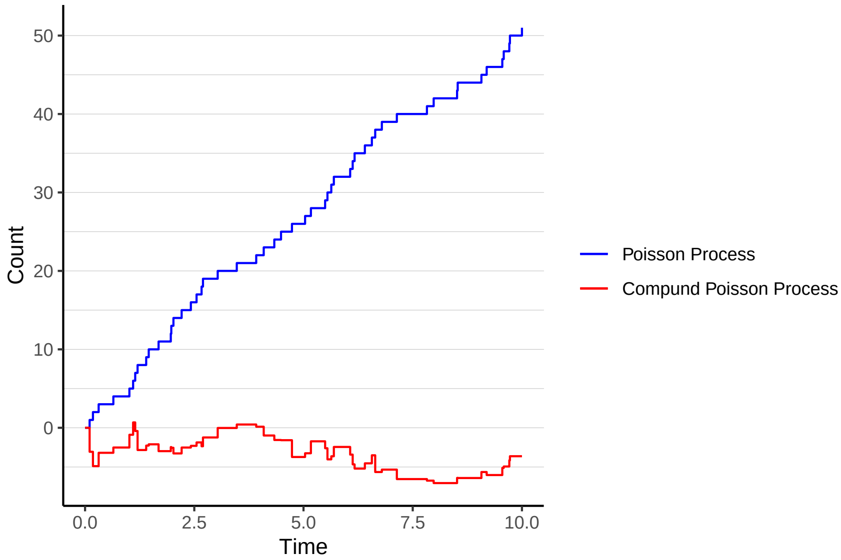


Figure 3.1: Blue line: a Poisson process with intensity $\nu = 5$. Red line: Compound Poisson process with intensity $\nu = 5$ and $Z_n \sim N(0, 1)$.

In Figure 3.1 we can observe how the Poisson process' sample path completely changes when we make the jump size stochastic. The expectation of Y_t heavily relies on the expectation of Z_n. This can be elaborated by the characteristic function of a compound Poisson process. From the characteristic function we can calculate the moments which can be useful if we want to estimate the parameters of a sample path.

Proposition 3.4 (Characteristic Function of a Compound Poisson Process)**.** *Let $(Y_t)_{t\geq 0}$ be a compound Poisson process on $\mathbb{R}^d$. Its characteristic function has the following representation:*

$$\Phi_{Y_t}(z) = \mathbb{E}\left[e^{i\langle z, Y_t\rangle}\right] = \exp\{\nu t(\Phi_Z(z) - 1)\} \quad \textit{for } t \geq 0 \textit{ and } z \in \mathbb{R}^d, \tag{3.2}$$

where $\Phi_Z(z) = \mathbb{E}\left[e^{i\langle z, Z_n\rangle}\right]$ is the characteristic function of the Z_n's.

Proof. Fix a $t \geq 0$ and $z \in \mathbb{R}^d$. We now condition on $N_t = n$ for $n \in \mathbb{N}_0$.

$$\begin{aligned}
\mathbb{E}\left[e^{i\langle z, Y_t\rangle} | N_t = n\right] &= \mathbb{E}\left[e^{i\langle z, \sum_{j=1}^n Z_j\rangle} \middle| N_t = n\right] \\
&= \mathbb{E}\left[e^{i\langle z, \sum_{j=1}^n Z_j\rangle}\right] \\
&= \prod_{j=1}^n \mathbb{E}\left[e^{i\langle z, Z_j\rangle}\right] \\
&= (\Phi_Z(z))^n.
\end{aligned}$$

Then by applying the law of total expectation:

$$\begin{aligned}
\mathbb{E}\left[e^{i\langle z, Y_t\rangle}\right] &= \mathbb{E}\left[\mathbb{E}\left[e^{i\langle z, Y_t\rangle} | N_t = n\right]\right] \\
&= \sum_{n=0}^{\infty} (\Phi_Z(z))^n \mathbb{P}(N_t = n)
\end{aligned}$$

$$= \sum_{n=0}^{\infty} (\Phi_Z(z))^n e^{-\nu t} \frac{(\nu t)^n}{n!}$$
$$= \exp\{\nu t(\Phi_Z(z) - 1)\}$$

■

From the characteristic function we can define the moment generating function, if it exists. The moment generating function does not always exist but if it is well-defined its relation with the characteristic function is given as:

$$M_X(z) = \Phi_X(-iz). \tag{3.3}$$

The last thing we need to be able to model the odds is the distribution of Z_n's. Due to the exchange structure we want discrete jump sizes that only can take odds defined in Table 2.2. In the next section we will discuss the challenges of this problem and present a solution to solve this problem.

3.2 Jump Sizes of Odds Trajectories

This section is also based on [Gan and Kolaczyk, 2018].

To model the odds trajectories with a compound Poisson process we need to determine an appropriate distribution for the jump sizes. In this section we will present such a discrete distribution. Lastly, we will formally define a continuous model for the odds trajectories and use our model to simulate a sample path of odds movement.

There are multiple challenges involved with modeling the jump sizes of odds. First and foremost the realizations of odds in the model have to be tradable. But there is one big problem of finding a distribution that only takes tradable odds. The problem is, that increment of odds can not be modeled with identically distributed random variables since the odds jump sizes depend on the odds at time t. For example, assume the odds are 1.58 then the jump size is 0.01, however, if the odds increase to 2.06 then the jump size is 0.02. By observing Table 2.2 on page 7, we can determine that there are multiple levels where the increment changes, these levels are sometimes referred to as *crossover points*. To work around this problem we suggest modeling the jump size as the number of ticks instead of odds increments. To do this we need a discrete distribution that can take positive and negative integer values. This can be done by choosing a Skellam distribution. The Skellam distribution is defined as the difference of two independent Poisson distributions.

Definition 3.5 (Skellam Distribution)
Let $X \sim Pois(\lambda_1)$ and $Y \sim Pois(\lambda_2)$. Furthermore, let X and Y be independent and define $Z \sim X - Y$, then $Z \sim \mathrm{Sk}(\lambda_1, \lambda_2)$, for $\lambda_1, \lambda_2 > 0$. The PMF is then given as:

$$\mathbb{P}(Z = z) = e^{-(\lambda_1+\lambda_2)} \left(\frac{\lambda_1}{\lambda_2}\right)^{z/2} I_z(2\sqrt{\lambda_1\lambda_2}), \quad \forall z \in \mathbb{Z} \tag{3.4}$$

where $I_z(\cdot)$ is the modified Bessel function of the first kind. We will denote the PMF of a Skellam distribution as $Sk(\lambda_1, \lambda_2)$.

Because the Skellam distribution is constructed by two independent Poisson distributions the Skellam distribution can be skewed and have nonzero mean values. Therefore, it is possible to create trends in the odds trajectory. Some examples of the skellem distribution are shown in Figure 3.2.

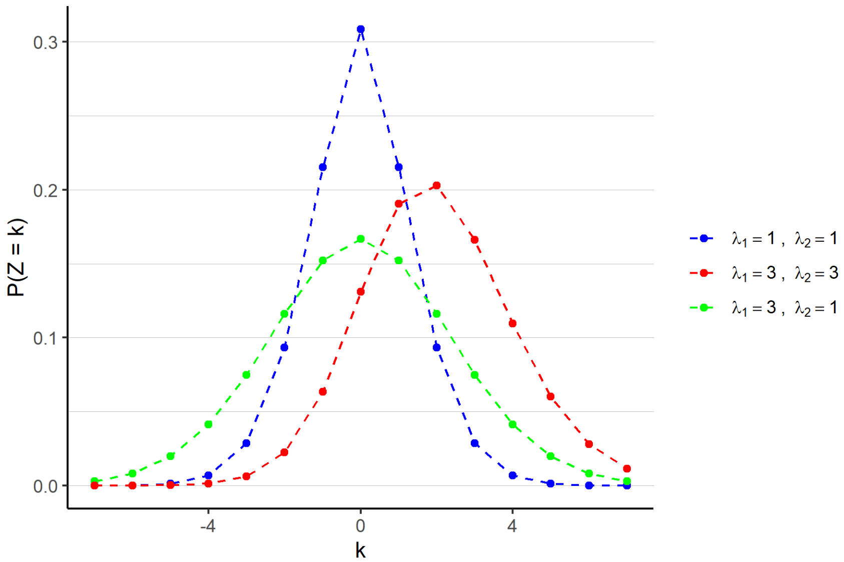


Figure 3.2: The figure shows three different densities for a Skellam distribution.

The Skellam distribution makes it possible to model the number of ticks the odds move since it can take both positive and negative integer values. We also notice that the jump size can be 0, this is interpreted as a trade that does not move the price.

If we wish to use the moments we need the characteristic function. The characteristic function of a Skellam distribution is defined as:

Proposition 3.6 (Characteristic Function of a Skellam Distribution). *Let $X \sim Sk(\lambda_1, \lambda_2)$, then the characteristic function of X is:*

$$\Phi_X(z) = \mathbb{E}\left[e^{i\langle z,X\rangle}\right] = \exp\{-(\lambda_1 + \lambda_2) + \lambda_1 e^{iz} + \lambda_2 e^{-iz}\} \tag{3.5}$$

Now we can formally define a model for the odds trajectory. In the next section we combine our findings, define the model and present some basic properties of our odds trajectory model.

3.3 Modeling the Odds Trajectories

This section is also based on [Ozel, 2013].

Our model of odds trajectory is defined as a compound Poisson process with the jump distribution defined as a Skellam distributed random variables. We define the odds trajectory model as

Definition 3.7 (The Odds Trajectory Model)
Let $(N_t)_{t\geq 0}$ denote a Poisson process with parameter ν. Let $(Z_n)_{n\geq 1}$ denote i.i.d. Skellam distributed random variables with parameters $\lambda_1, \lambda_2 > 0$. We define the odds trajectory model as:

$$O_t = \sum_{j=1}^{N_t} Z_j, \tag{3.6}$$

where $Z_j \sim Sk(\lambda_1, \lambda_2)$ and we use the convention $\sum_{j=1}^{0} = 0$.

Because the model is a compound Poisson Process it is fairly easy to determine the moments of this model. The moments can be calculated by using the moment generating function. The relationship between the characteristic function and moment generating function is given by (3.3). By using the characteristic function for a compound Poisson process in (3.2), the moment generating function (MGF) of O_t is therefore given as:

$$M_{O_t}(z) = \exp\{\nu t[M_Z(z) - 1]\}, \tag{3.7}$$

where $M_Z(z)$ is the MGF of a Skellam distribution. The MGF of Z_n is calculated by using (3.5) and (3.3), that is:

$$\begin{aligned} M_Z(z) = \Phi_Z(-iz) &= \exp\{-(\lambda_1 + \lambda_2) + \lambda_1 e^{i(-iz)} + \lambda_2 e^{-i(-iz)}\} \\ &= \exp\{-(\lambda_1 + \lambda_2) + \lambda_1 e^{z} + \lambda_2 e^{-z}\} \end{aligned}$$

The n'th moment is calculated as:

$$m_n = \mathbb{E}\left[Z^n\right] = M_Z^{(n)}(0) = \frac{d^n M_Z}{dz^n}\bigg|_{z=0}.$$

Therefore, we can determine the first three raw moments Z_n as:

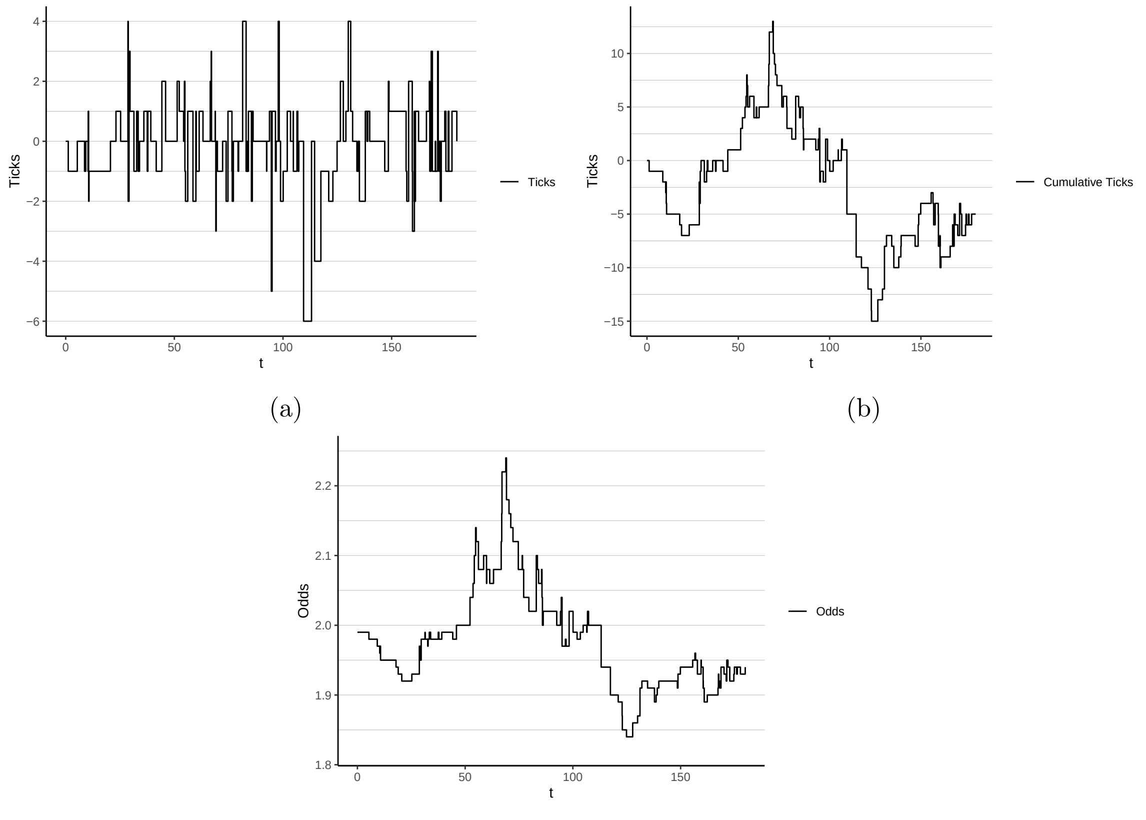
$$\begin{aligned} m_1 &= \lambda_1 - \lambda_2 \\ m_2 &= (\lambda_1 - \lambda_2)^2 + (\lambda_1 + \lambda_2) \\ m_3 &= (\lambda_1 - \lambda_2) + (\lambda_1 + \lambda_2)(\lambda_1 - \lambda_2) + 2(\lambda_1 + \lambda_2)(\lambda_1 - \lambda_2) + (\lambda_1 - \lambda_2)^3 \\ &= (\lambda_1 - \lambda_2)(1 + 3(\lambda_1 - \lambda_2) + (\lambda_1 - \lambda_2)^2) \end{aligned}$$

Now, let the rth raw moment of O_t be denoted as $\xi_r = \mathbb{E}\left[O_t^r\right] = \frac{d^r}{dt^r} M_{O_t}(t)|_{t=0}$, $r = 1, 2, ..., n$. The first three moment of O_t can be obtained as:

$$
\begin{aligned}
\xi_1 &= (\nu t m_1) \\
\xi_2 &= (\nu t m_1)^2 + (\nu t m_2) \\
\xi_3 &= (\nu t m_1)^3 + 3(\nu t m_1)(\nu t m_2) + (\nu t m_3)
\end{aligned}
\tag{3.8}
$$

The main advantage by modeling the odds change as ticks increment is that we will always land on tradable odds. Another advantage is that we can better compare characteristics, such as mean values and volatility of different odds paths, because the movements are proportional to each other. In a sense this is similar to working with % returns instead of prices, in other financial markets. However, with our model, we can not compare profit and loss with respect to ticks. Simply because, per units, one tick on 0.01 will yield less return than one tick on 1.00.

To visualize our model we will simulate it and then transform this tick path into odds. The simulation is made by having N_t being a Poisson process with intensity $\nu = 5$ and the jump distribution be $Z_j \sim Sk(1.5, 1.5)$. In Figure 3.3c we simulate a sample path from our model.

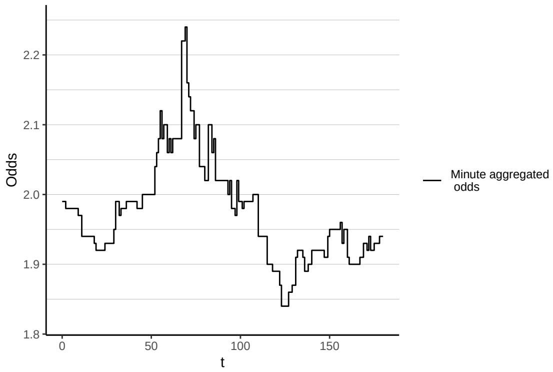


(a) (b) (c)

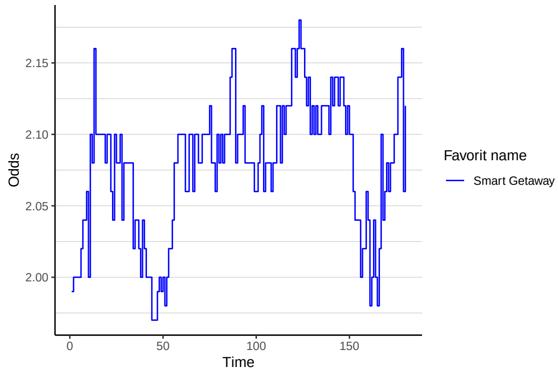
Figure 3.3: (a) Shows the tick increments from our model. (b) Shows the cumulative tick increments from our model. (c) We transform the ticks into odds increments. We used the starting odds $t_0 = 1.99$

From Figure 3.3 we can observe how the crossover point, at odds 2, changes the odds increments. Around $t = 100$ we observe in Figure 3.3b how there occur some large negative jumps, but in Figure 3.3c the negative movements do not look out of the ordinary, due to the change in odds increments. Next, we will compare the simulated odds path with actual data from the race in

Southwell. Since our data is minute data we can aggregate the number of ticks each minute. By doing so it reflects one-minute observations and thereby is more comparable with our data.



(a)



(b)

Figure 3.4: (a) Show the simulated odds path with an aggregation of odds each minute. (b) Show an observed odds path from a race in Southwell.

In Figure 3.4 we see the odds trajectory model compared to actual odds. We believe that this model is a good foundation to model the odds trajectories. The model does have some limitations. Firstly we do not have any boundaries, therefore the odds in our model can, in theory, be greater than $1,000$ and less than 1.01, which are the boundaries of the tradable odds. Furthermore, the odds can even move into negative. This does not make any sense because no inverse probability can reflect negative odds. Another concern about this model is the fact that the parameters are not time dependent. We believe this to be a major issue with the model. In the next chapter we will build a state-space model that has Skellam distributed observations. By doing so we can introduce time-varying parameters.

4 State-Space Models

As mentioned previously, one major drawback of the compound Poisson model is the fixed parameters. The odds in practice does not reflect such stationary parameters. For example the odds can become more volatile or can even change trend direction. In Section 6.1 we will show such examples. One way to let λ_1 and λ_2 in Definition 3.7 be more dynamic is by assuming that there is an underlying market state that creates the observations. This motivates a state-space model. Since we want the observations to be Skellam distributed we will build a non-Gaussian state-space model. Therefore, we introduce the dynamic Skellam model in Chapter 5 which is a state-space model with Skellam distributed observations. It will later be discussed how the likelihood of this Dynamic Skellam model is difficult to evaluate. Therefore, we adopt the method of importance sampling in Section 5.1 to calculate the likelihood. One key component of calculating the likelihood via importance sampling is to calculate the likelihood of the importance density. In our case the importance density is Gaussian and therefore we will start by introducing the linear Gaussian state-space model. Next, the Gaussian linear state-space model can be extended to a general non-linear non-Gaussian state-space model. The purpose of this chapter is to give some basic results that will later be used in Chapter 5.

4.1 Dynamic Linear Gaussian Model

This section is based on [Shumway and Stoffer, 2017, pp.290-304]

In state-space models we work with two equations, a state equation (4.2) and a observation equation (4.1). The state equation can not be observed. The task of a state-space model is to determine the state equation that produces the observations. We define the two equations in a general case where exogenous variables can enter the equations.

$$y_t = A_t x_t + \Gamma u_t + v_t \quad v_t \sim \text{i.i.d.} N_q(0, R), \tag{4.1}$$

$$x_t = \Phi x_{t-1} + \Upsilon u_t + w_t, \quad w_t \sim \text{i.i.d.} N_p(0, Q), \tag{4.2}$$

where $x_t \in \mathbb{R}^{p\times 1}$, $\Phi \in \mathbb{R}^{p\times p}$ and the covariance matrix $Q \in \mathbb{R}^{p\times p}$. We refer to $u_t \in \mathbb{R}^{r\times 1}$ as a vector of inputs and $\Upsilon \in \mathbb{R}^{p\times r}$ is a coefficient matrix for the state equation. The observation equation consists of $y_t \in \mathbb{R}^{q\times 1}$, $\Gamma \in \mathbb{R}^{q\times r}$ is a coefficient matrix, $A_t \in \mathbb{R}^{q\times p}$ is called an observation matrix and the covariance matrix $R \in \mathbb{R}^{q \times q}$ for the observation equation.

The aim is to estimate the unobserved state, x_t, for some observations up to time s, given the observed data $y_{1:s} = (y_1, \dots, y_s)'$. When we estimate these parameters for $s < t$ we refer to it as prediction. When $s = t$ we call it filtering and lastly when $s > t$ it is called smoothing. In order to produce estimators for filtering and smoothing we use the Kalman filter and smoother (KFS).

For this chapter we will adopt the definitions:

$$x_t^s = \mathbb{E}\left[x_t | y_{1:s}\right] \tag{4.3}$$

and we also define:

$$P_{t_1,t_2}^s = \mathbb{E}\left[(x_{t_1} - x_{t_1}^s)(x_{t_2} - x_{t_2}^s)'\right]. \tag{4.4}$$

In the case where $t_1 = t_2$ we will for convenience write P_t^s. The estimators we obtain with KFS are the mean-squared error estimators. Therefore, we can consider (4.3) as projection operator rather than an expectation and $y_{1:s}$ as the space of linear combinations of $\{y_1, \dots, y_s\}$. On these terms, P_t^s is the corresponding mean-squared error. Because we assume the process to be Gaussian we have that (4.4) is also the conditional error covariance, given as:

$$P_{t_1,t_2}^s = \mathbb{E}\left[(x_{t_1} - x_{t_1}^s)(x_{t_2} - x_{t_2}^s)' | y_{1:s}\right]. \tag{4.5}$$

The estimators for the linear Gaussian model is derived by minimizing (4.5). Lastly, before we present the Kalman filter and the proof of the Kalman filter we will state a key result for a conditional expectation of a jointly Gaussian distribution.

Let $y = (y_1, \dots, y_m)'$ and $x = (x_1, \dots, x_n)'$. Suppose that x and y are jointly Gaussian distributed:

$$\begin{bmatrix} y \\ x \end{bmatrix} \sim N_{m+n}\left(\begin{bmatrix} \mu_y \\ \mu_x \end{bmatrix}, \begin{bmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{bmatrix} \right),$$

then $y|x$ is normal with

$$\mu_{y|x} = \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x - \mu_x) \tag{4.6}$$

$$\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}, \tag{4.7}$$

where Σ_{xx} is assumed to be nonsingular. We will now proceed to present the Kalman Filter.

Proposition 4.1 (The Kalman Filter). *For the state-space model specified in* (4.2) *and* (4.1), *with initial conditions* $x_0^0 = \mu_0$ *and* $P_0^0 = \Sigma_0$, *for* $t = 1, \dots, n$,

$$x_t^{t-1} = \Phi x_{t-1}^{t-1} + \Upsilon u_t, \tag{4.8}$$

$$P_t^{t-1} = \Phi P_{t-1}^{t-1}\Phi' + Q, \tag{4.9}$$

with

$$x_t^t = x_t^{t-1} + K_t\left(y_t - A_t x_t^{t-1} - \Gamma u_t\right), \tag{4.10}$$

$$P_t^t = \left[I - K_t A_t\right] P_t^{t-1}, \tag{4.11}$$

where

$$K_t = P_t^{t-1} A_t' \left[A_t P_t^{t-1} A_t' + R\right]^{-1}, \tag{4.12}$$

is called the Kalman gain. Prediction for $t > n$ is accomplished via (4.8) *and* (4.9) *with initial conditions x_n^n and P_n^n. Important byproducts of the filter are the innovations*

$$\epsilon_t = y_t - \mathbb{E}\left[y_t | y_{1:t-1}\right] = y_t - A_t x_t^{t-1} - \Gamma u_t,$$

and the corresponding variance-covariance matrices

$$\Sigma_t \stackrel{def}{=} \operatorname{var}(\epsilon_t) = \operatorname{var}\left[A_t\left(x_t - x_t^{t-1}\right) + v_t\right] = A_t P_t^{t-1} A_t' + R,$$

for $t = 1, \ldots, n$. We assume Σ_t to be positive definite, which is guaranteed, for example, if $R > 0$. This assumption is not necessary and may be relaxed.

Proof. We begin by deriving (4.8) and (4.9), from (4.2) we have

$$x_t^{t-1} = \mathbb{E}\left[x_t | y_{1:t-1}\right] = \mathbb{E}\left[\Phi x_{t-1} + \Upsilon u_t + w_t | y_{1:t-1}\right] = \Phi x_{t-1}^{t-1} + \Upsilon u_t$$

thus

$$\begin{aligned} P_t^{t-1} &= \mathbb{E}\left[(x_t - x_t^{t-1})(x_t - x_t^{t-1})'\right] \\ &= \mathbb{E}\left[\left[\Phi(x_{t-1} - x_{t-1}^{t-1}) + w_t\right]\left[\Phi(x_{t-1} - x_{t-1}^{t-1}) + w_t\right]'\right] \\ &= \Phi P_{t-1}^{t-1} \Phi' + Q. \end{aligned}$$

Next, we derive (4.10). We note that $\operatorname{cov}(\epsilon_t, y_t) = 0$ for $s < t$, which implies that the innovations are independent of the past observations. Furthermore, the conditional covariance between x_t and ϵ_t given $y_{1:t-1}$ is

$$\begin{aligned} \operatorname{cov}(x_t, \epsilon_t | y_{1:t-1}) &= \operatorname{cov}(x_t, y_t - A_t x_t^{t-1} - \Gamma u_t | y_{1:t-1}) \\ &= \operatorname{cov}(x_t - x_t^{t-1}, y_t - A_t x_t^{t-1} - \Gamma u_t | y_{1:t-1}) \\ &= \operatorname{cov}(x_t - x_t^{t-1}, A_t(x_t - x_t^{t-1}) + v_t) \\ &= P_t^{t-1} A_t'. \end{aligned}$$

With these results we have the joint conditional distribution of x_t and ϵ_t given $y_{1:t-1}$ is Gaussian, that is:

$$\begin{bmatrix} x_t \\ \epsilon_t \end{bmatrix} \Bigg| y_{1:t-1} \sim N\left(\begin{bmatrix} x_t^{t-1} \\ 0 \end{bmatrix}, \begin{bmatrix} P_t^{t-1} & P_t^{t-1} A_t' \\ A_t P_t^{t-1} & \Sigma_t \end{bmatrix}\right). \tag{4.13}$$

Now, by using (4.6), we can write

$$x_t^t = \mathbb{E}\left[x_t | y_{1:t}\right] = \mathbb{E}\left[x_t | y_{1:t-1}, \epsilon_t\right] = x_t^{t-1} + K_t \epsilon_t,$$

where

$$K_t = P_t^{t-1} A_t' \Sigma_t^{-1} = P_t^{t-1} A_t' \left[A_t P_t^{t-1} A_t' + R\right]^{-1}.$$

The evaluation of P_t^t is now computed from (4.13), by using (4.7) as

$$P_t^t = \text{cov}(x_t|y_{1:t-1}, \epsilon_t) = P_t^{t-1} - P_t^{t-1}A_t'\Sigma_t^{-1}A_tP_t^{t-1},$$

and can be simplified to (4.11). ■

We can also formulate The Kalman filter with time-varying parameters or where the observation dimension changes with time, which is given in the following corollary.

Corollary 4.2 (Kalman Filter: The Time-Varying Case)**.** *If,* (4.2) *and* (4.1)*, any or all of the parameters are time dependent, $\Phi = \Phi_t, \Upsilon = \Upsilon_t, Q = Q_t$ in the state equation or $\Gamma = \Gamma_t, R = R_t$ in the observation equation, or the dimension of the observational equation is time dependent, $q = q_t$, Proposition 4.1 holds with the appropriate substitutions.*

In summary, the Kalman filter estimates the present state using the past and present information. The current estimation of x_t relies on the Kalman gain (4.12). If the previous predicted estimate of covariance P_t^{t-1} is small then the Kalman gain in (4.12) goes to zero i.e.:

$$\lim_{P_t^{t-1} \to 0} K_t = 0,$$

and a consequence of $K_t = 0$ is that $x_t = x_t^{t-1}$. If the covariance matrix for the observations, R, goes to zero then the Kalman gain will be equal to the inverse of the observation matrix that is:

$$\lim_{R \to 0} K_t = A_t^{-1},$$

which implies that we mostly rely on our observations to estimate the state. In this way the Kalman filter estimates the state values by adjusting the Kalman gain.

When we use the Kalman smoother we are allowing the estimation to be a function of the past, present, and future. The Kalman smoother is calculated as:

Proposition 4.3 (The Kalman Smoother)**.** *For the state-space model specified in* (4.2) *and* (4.1)*, with initial conditions x_n^n and P_n^n obtained via Proposition 4.1, for $t = n, n-1, \ldots, 1$,*

$$x_{t-1}^n = x_{t-1}^{t-1} + J_{t-1}\left(x_t^n - x_t^{t-1}\right), \tag{4.14}$$

$$P_{t-1}^n = P_{t-1}^{t-1} + J_{t-1}\left(P_t^n - P_t^{t-1}\right)J_{t-1}', \tag{4.15}$$

where

$$J_{t-1} = P_{t-1}^{t-1}\Phi'\left[P_t^{t-1}\right]^{-1}.$$

Proof. First, for $1 \le t \le n$, define

$$y_{1:t-1} = (y_1, \ldots, y_{t-1})' \quad \text{and} \quad \eta_t = (v_t, \ldots, v_n, w_{t+1}, \ldots, w_n)'$$

with $y_{1:0}$ being empty, and let

$$m_{t-1} = \mathbb{E}\left[x_{t-1}|y_{1:t-1}, x_t - x_t^{t-1}, \eta_t\right]$$

Then, because $y_{1:t-1}, (x_t - x_t^{t-1})$, and η_t are mutually independent, and x_{t-1} and η_t are independent, using (4.6) we have

$$m_{t-1} = x_{t-1}^{t-1} + J_{t-1}\left(x_t - x_t^{t-1}\right)$$

where

$$J_{t-1} = \mathrm{cov}(x_{t-1}, x_t - x_t^{t-1})\left[P_t^{t-1}\right]^{-1} = P_{t-1}^{t-1}\Phi'\left[P_t^{t-1}\right]^{-1}$$

Finally, because $y_{1:t-1}, (x_t - x_t^{t-1})$, and η_t generate $y_{1:n} = (y_1, \ldots, y_n)*'$,

$$x_{t-1}^n = \mathbb{E}\left[x_{t-1}|y_{1:n}\right] = \mathbb{E}\left[m_{t-1}|y_{1:n}\right] = x_{t-1}^{t-1} + J_{t-1}\left(x_t^n - x_t^{t-1}\right),$$

which establishes (4.14).

The recursion for the error covariance, P_{t-1}^n, is obtained by using (4.14) we obtain

$$x_{t-1} - x_{t-1}^n = x_{t-1} - x_{t-1}^{t-1} - J_{t-1}\left(x_t^n - \Phi x_{t-1}^{t-1}\right),$$

or

$$\left(x_{t-1} - x_{t-1}^n\right) + J_{t-1}x_t^n = \left(x_{t-1} - x_{t-1}^{t-1}\right) + J_{t-1}\Phi x_{t-1}^{t-1}. \tag{4.16}$$

Multiplying each side of (4.16) by the transpose of itself and taking expectation, we have

$$P_{t-1}^n + J_{t-1}\,\mathbb{E}\left[x_t^n x_t^{n'}\right]J_{t-1}' = P_{t-1}^{t-1} + J_{t-1}\Phi\,\mathbb{E}\left[x_{t-1}^{t-1}x_{t-1}^{t-1'}\right]\Phi' J_{t-1}', \tag{4.17}$$

using that the cross-product terms are zero. But,

$$\mathbb{E}\left[x_t^n x_t^{n'}\right] = \mathbb{E}\left[x_t x_t'\right] - P_t^n = \Phi\,\mathbb{E}\left[x_{t-1}x_{t-1}'\right]\Phi' + Q - P_t^n,$$

and

$$\mathbb{E}\left[x_{t-1}^{t-1}x_{t-1}^{t-1}\right] = \mathbb{E}\left[x_{t-1}x_{t-1}'\right] - P_{t-1}^{t-1}$$

so (4.17) simplifies to (4.15). ■

In order to build a state-space model with non-Gaussian observations we will in the next section introduce the nonlinear non-Gaussian model.

4.2 Nonlinear non-Gaussian State-Space Models

This section is based on [Durbin and Koopman, 2012, pp. 209]

In this section we introduce the general form of a nonlinear non-Gaussian distributed state-space model. When we are working with non-Gaussian state-space models we will adopt the notation from Durbin and Koopman [2012] and we define the nonlinear non-Gaussian state-space model as:

$$
\begin{aligned}
y_t &\sim p(y_t|\alpha_t), \\
\alpha_{t+1} &\sim p(\alpha_{t+1}|\alpha_t), \quad \alpha_1 \sim p(\alpha_1)
\end{aligned}
$$

for $t = 1, \ldots, n$. Furthermore, we assume that

$$
\begin{aligned}
p(y_{1:n}|\alpha) &= \prod_{t=1}^{n} p(y_t|\alpha_t), \\
p(\alpha) &= p(\alpha_1) \prod_{t=1}^{n-1} p(\alpha_{t+1}|\alpha_t),
\end{aligned}
$$

where $y_{1:n} = (y'_1, \ldots, y'_n)'$ and $\alpha = (\alpha'_1, \ldots, \alpha'_n)'$. The observation density $p(y_t|\alpha_t)$ implies the link between the observation vector y_t and state vector α_t. The state update density $p(\alpha_{t+1}|\alpha_t)$ is the relationship between the state vector of the next period and the current period.

This state-space model allows us to model the observations as being Skellam distributed. Furthermore, because it is a state-space model we can make the parameters time dependent. This solves one of the biggest concerns for the odds trajectory model, defined in Definition 3.7. However, the non-Gaussian state models also present some new challenges when it comes to filtering and smoothing. We can no longer rely on the Kalman filter smoother method because these methods heavily rely on the Gaussian assumptions. In the next chapter we will present an alternative method for filtering and smoothing of the nonlinear non-Gaussian state-space model.

Dynamic Skellam model 5

This chapter is based on [Koopman, Lit, and Lucas, 2014], [Koopman, Lit, and Lucas, 2017] and [Durbin and Koopman, 2012, ch. 11]

In the case of the linear state-space model we can apply the Kalman filter smoother to estimate the state equation but in our case where the observation equation is Skellam distributed we need another method for filtering and smoothing. These methods will be discussed in more depth throughout this chapter. But first we define the dynamic Skellam model.

The Dynamic Skellam Model

In this state-space model we wish to model a Skellam distribution with time-varying parameters. Therefore, we replace Z, z, λ_1 and λ_2 from (3.4) on page 13 with their time-varying counterparts Z_t, z_t, $\lambda_{1,t}$ and $\lambda_{2,t}$ respectively. The dynamic Skellam model is denoted as:

$$Z_t \sim Sk(\lambda_{1,t}, \lambda_{2,t}), \quad t = 1, 2, \dots, n, \tag{5.1}$$

where n is the length of the time series. The dynamics of $\lambda_{1,t}$ and $\lambda_{2,t}$ is modeled by the nonlinear transformation of the autoregressive process,

$$\lambda_{it} = s_i(\theta_t), \tag{5.2}$$

$$\theta_t = c_t + M_t\alpha_t, \tag{5.3}$$

$$\alpha_{t+1} = d_t + T_t\alpha_t + \eta_t, \quad \eta_t \sim N(0, Q_t), \tag{5.4}$$

for $i = 1, 2$ and $t = 1, \dots, n$, where $s_i(\cdot) : \mathbb{R}^{p\times 1} \to \mathbb{R}^+$ is the exponential link function, $\theta_t \in \mathbb{R}^{p\times 1}$ is a signal vector, $\alpha_t \in \mathbb{R}^{m\times 1}$ is referred to as the state vector, $c_t \in \mathbb{R}^{p\times 1}$ and $d_t \in \mathbb{R}^{m\times 1}$ are vectors of intercepts, $M_t \in \mathbb{R}^{p\times m}$ is a matrix, $T_t \in \mathbb{R}^{m\times m}$ is referred to as a transition matrix and the disturbances η_t are Gaussian and independently distributed with mean zero and the covariance matrix $Q_t \in \mathbb{R}^{m\times m}$ is semi-definite.

The model 5.1 falls within the class of non-Gaussian nonlinear state space models that is models with the structure:

$$y_t \sim p(y_t|\theta_t;\psi), \tag{5.5}$$

$$\theta_t = c_t + M_t\alpha_t, \tag{5.6}$$

$$\alpha_{t+1} \sim p_g(\alpha_{t+1}|\alpha_t;\psi), \quad t = 1,2,\dots,n, \tag{5.7}$$

where $p_g(\cdot)$ denotes a Gaussian density, the parameter vector ψ contains unknown and fixed parameters gathering all the parameters in c_t, Z_t, d_t, T_t, Q_t, and the link functions $s_i(\cdot)$ for $i = 1,2$. The Gaussian state density $p_g(\alpha_{t+1}|\alpha_t;\psi)$ refers to the linear Markov process (5.4), and $p_g(\alpha_1;\psi)$ is the initial condition for α_1. We assume that for given realizations of the signal vector $\theta = (\theta_1', \dots, \theta_n')'$ the observations $y = (y_1, \dots, y_n)'$ are conditionally independent. The joint conditional density for all observations and the marginal density for all states can be written as

$$p(y|\theta;\psi) = \prod_{t=1}^{n} p(y_t|\theta_t;\psi), \tag{5.8}$$

$$p_g(\alpha;\psi) = p_g(\alpha_1;\psi)\prod_{t=2}^{n} p_g(\alpha_t|\alpha_{t-1};\psi). \tag{5.9}$$

The main challenge for the dynamic Skellam model is to evaluate the likelihood function $\int p(y|\theta;\psi)p_g(\alpha;\psi)d\alpha$ which is difficult to evaluate analytically. Our solution is to evaluate the likelihood function via importance sampling. In the next section we will introduce the general method of importance sampling and how to use this method to evaluate the likelihood function of the dynamic Skellam model.

5.1 Likelihood Evaluation via Importance Sampling

This section is also based on [Iacus, 2009, pp. 9-11].

The likelihood function for (5.1) has the form of a non-linear state-space model and we can express it as:

$$L(y;\psi) = \int p(y,\theta;\psi)d\theta = \int p(y|\theta;\psi)p_g(\theta;\psi)d\theta, \tag{5.10}$$

which is an expectation and can be evaluated with the use of Monte Carlo simulation. Using naive Monte Carlo simulation we can evaluate the likelihood with the estimate:

$$\frac{1}{M}\sum_{k=1}^{M} p(y|\theta^{(k)};\psi), \quad \theta^{(k)} \sim p_g(\theta;\psi), \tag{5.11}$$

Where $p_g(\theta;\psi)$ is the joint density given by (5.9). This is not a feasible approach to evaluate the likelihood due to the slow converting rate of naive Monte Carlo method which is of order $\frac{1}{\sqrt{M}}$. For this reason we introduce importance sampling.

Consider $Z \sim f$ and suppose we want to evaluate $\mathbb{E}[h(Z)]$ where $h(Z)$ has large variance.

Furthermore, suppose we can find a density g such that

$$\frac{h(z)f(z)}{g(z)} \approx \text{const} \quad \text{and } h(z)f(z) > 0 \Rightarrow g(z) > 0.$$

Then we can evaluate $\mathbb{E}\left[h(z)\right]$ as:

$$\mathbb{E}\left[h(z)\right] = \int h(z)f(z)dz = \int \frac{h(z)f(z)}{g(z)}g(z)dz = \mathbb{E}\left[\frac{h(Y)f(Y)}{g(Y)}\right],$$

where $Y \sim g$. Since $\frac{h(z)f(z)}{g(z)} \approx$ const then the Monte Carlo error is small hence the convergence rate is must faster. For our implementation of importance sampling we choose a Gaussian importance density $g(\theta|y;\psi^*)$. From the Gaussian importance density θs are sampled conditional on the observations vector y. Let ψ^* denote a fixed parameter vector containing ψ as well as parameters $\widetilde{\psi}$ to the importance density $g(y|\theta;\widetilde{\psi})$, i.e. $\psi^* = (\psi', \widetilde{\psi}')'$. With this importance density we can express the likelihood (5.10) as:

$$L(y;\psi) = \int \frac{p(y,\theta;\psi)}{g(\theta|y;\psi^*)}g(\theta|y;\psi^*)d\theta, \tag{5.12}$$

where the likelihood estimate is given by:

$$\frac{1}{M}\sum_{k=1}^{M}\omega\left(y,\theta^{(k)};\psi^*\right), \quad \omega\left(y,\theta;\psi^*\right) = \frac{p(y,\theta;\psi)}{g\left(\theta|y;\psi^*\right)}, \quad \theta^{(k)} \sim g\left(\theta|y;\psi^*\right), \tag{5.13}$$

where $\theta^{(k)}$ is drawn independently for $k = 1,\ldots,M$. We assume that $p_g(\theta;\psi) = g(\theta;\psi)$. It now follows that

$$\begin{aligned}
\omega(y,\theta;\psi^*) &= \frac{p(y,\theta;\psi)}{g(\theta|y;\psi^*)} && (5.14)\\
&= \frac{p(y|\theta;\psi)p_g(\theta;\psi)}{g(y|\theta;\widetilde{\psi})g(\theta;\psi)/g(y;\psi^*)} && (5.15)\\
&= g(y;\psi^*)\frac{p(y|\theta;\psi)}{g(y|\theta;\widetilde{\psi})}, && (5.16)
\end{aligned}$$

where the second equality uses Bayes' theorem $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$. By substituting (5.16) into (5.12) we can write the likelihood as:

$$\begin{aligned}
L(y;\psi) &= \int \frac{p(y,\theta;\psi)}{g(\theta|y;\psi^*)}g(\theta|y;\psi^*)d\theta && (5.17)\\
&= g(y;\psi^*)\int \frac{p(y|\theta;\psi)}{g(y|\theta;\widetilde{\psi})}g(\theta|y;\psi^*)d\theta && (5.18)\\
&= L_g(y;\psi^*)\mathbb{E}_g[\omega(y,\theta;\psi^*)], && (5.19)
\end{aligned}$$

where $L_g(y;\psi^*)$ is the likelihood of the linear Gaussian model obtained by the importance density $g(\theta|y;\psi^*)$, $\mathbb{E}_g[\cdot]$ is the expectation with respect to density $g(\theta|y;\psi^*)$. The likelihood estimate is therefore given as:

$$\widehat{L}(y;\psi) = L_g(y;\psi^*)\overline{\omega}, \tag{5.20}$$

where $\overline{\omega}$ is the same as (5.13) and $L_g(y;\psi^*)$ can be calculated with the Kalman filter smoother since it is assumed to be a linear Gaussian model. In practice it is more numerically stable to estimate the loglikelihood. The loglikelihood is given as:

$$\log \widehat{L}(y;\psi) = \log L_g(y;\psi^*) + \log \overline{\omega}. \tag{5.21}$$

In the next section we will present how importance sampling can be used in filtering for the dynamic Skellam model.

5.2 Filtering Nonlinear Non-Gaussian State-Space Models

This section is also based on [Durbin and Koopman, 2012, ch. 12] and [Gordon, Salmond, and Smith, 1993].

In the literature there are proposed different methods for filtering non-Gaussian state-space models such as the extended Kalman filter and the unscented Kalman filter. We will only focus on a method named particle filtering. The particle filtering also utilizes the method of importance sampling. It turns out that this method can be simplified to the bootstrap filter. The advantage of the bootstrap filter is that it uses less memory and it is fast. In the end of this section we will implement the bootstrap filter and make illustrative comparison with the Kalman filter in a simulated example.

In this section we will define a collection of state vectors as:

$$\alpha = (\alpha_1', \dots, \alpha_t')', \tag{5.22}$$

and the collection of the observations as:

$$y = (y_1, \dots, y_t)'.$$

Particle Filtering

In the previous sections we used importance sampling to evaluate the likelihood of the dynamic Skellam model. In this section we will show how we can use importance sampling technique for the filtering of the same model. More precise we are going to estimate the conditional mean:

$$\overline{x}_t = \mathbb{E}\left[x_t(\alpha)|y\right] \tag{5.23}$$

$$= \int x_t(\alpha)p(\alpha|y)d\alpha. \tag{5.24}$$

for $t = \tau+1, \tau+2, \dots$ where τ is fixed and can be zero, $x(\alpha)$ is an arbitrary function of α. Using importance sampling method we can express $\overline{x}_t$ as an expectation with respect to the importance density $g(\alpha|y)$.

$$\overline{x}_t = \mathbb{E}_g\left[x_t(\alpha)\frac{p(\alpha|y)}{g(\alpha|y)}\right] \tag{5.25}$$

By using $p(\alpha, y) = p(y)p(\alpha|y)$ and denote $\widetilde{w}_t = \frac{p(\alpha,y)}{g(\alpha|y)}$ we obtain

$$\overline{x}_t = \frac{1}{p(y)}\mathbb{E}_g[x_t(\alpha)\widetilde{w}_t]. \tag{5.26}$$

If we let $x_t(\alpha) = 1$ then it follows that $p(y) = \mathbb{E}_g[\widetilde{w}_t]$. We can now write (5.25) as:

$$\overline{x}_t = \frac{\mathbb{E}_g[x_t(\alpha)\widetilde{w}_t]}{\mathbb{E}_g[\widetilde{w}_t]}. \tag{5.27}$$

We estimate $\overline{x}_t$ by means of a random sample $\alpha^{(1)}, \dots, \alpha^{(N)}$ draw from $g(\alpha|y)$. Our estimator is

$$\begin{aligned}\widehat{x}_t &= \frac{N^{-1}\sum_{i=1}^N x_t(\alpha^{(i)})\widetilde{w}_t^{(i)}}{N^{-1}\sum_{i=1}^N \widetilde{w}_t^{(i)}} &(5.28)\\ &= \sum_{i=1}^N x_t(\alpha^{(i)})w_t^{(i)}, &(5.29)\end{aligned}$$

where

$$\widetilde{w}_t^{(i)} = \frac{p(\alpha^{(i)}, y)}{g(\alpha^{(i)}|y)}, \quad w_t^{(i)} = \frac{\widetilde{w}_t^{(i)}}{\sum_{j=1}^N \widetilde{w}_t^{(j)}}.$$

We refer to $\widetilde{w}_t^{(i)}$ as importance weights and the values $w_t^{(i)}$ are normalized importance weights. Next, we wish to calculate the importance weight recursively. Therefore, we begin by creating a recursion for the importance density:

$$\begin{aligned}g(\alpha^{(i)}|y) &= \frac{g(\alpha^{(i)}, y)}{g(y)} &(5.30)\\ &= \frac{g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y)g(\alpha_{1:t-1}^{(i)}, y)}{g(y)} &(5.31)\\ &= g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y)g(\alpha_{1:t-1}^{(i)}|y). &(5.32)\end{aligned}$$

Suppose that $\alpha_{1:t-1}^{(i)}$ only uses the information up to $t-1$. Therefore the observation y_t do not depend on the simulation $\alpha_{1:t-1}^{(i)}$. Thus the density $g(\alpha_{1:t-1}^{(i)}|y_{1:t-1})$ is not affected by including y_t on the set of variables $y_{1:t-1}$. Hence, $g(\alpha_{1:t-1}^{(i)}|y) \equiv g(\alpha_{1:t-1}^{(i)}|y_{1:t-1})$. Using this equality in (5.32), we obtain

$$g(\alpha^{(i)}|y) = g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y)g(\alpha_{1:t-1}^{(i)}|y_{1:t-1}).$$

We can now calculate $w_t^{(i)}$ recursively.

$$\begin{aligned}\widetilde{w}_t^{(i)} &= \frac{p(\alpha^{(i)}, y)}{g(\alpha^{(i)}|y_t)}\\ &= \frac{p(\alpha_{1:t-1}^{(i)}, y_{1:t-1})p(\alpha_t^{(i)}, y_t|\alpha_{1:t-1}^{(i)}, y_{1:t-1})}{g(\alpha_{1:t-1}^{(i)}|y_{1:t-1})g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y)}\\ &= \widetilde{w}_{t-1}^{(i)}\frac{p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})p(y_t|\alpha_t^{(i)})}{g(\alpha_t^{(i)}|\alpha_{1:t-1}^{(i)}, y_{1:t-1})}.\end{aligned}$$

Where the third equation uses the Markovian property in (5.4), that is:

$$p(\alpha_t^{(i)}, y_t|\alpha_{1:t-1}^{(i)}, y_{1:t-1}) = p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})p(y_t|\alpha_t^{(i)}). \tag{5.33}$$

By letting the importance density $g(\alpha_t^{(i)}|\alpha_{t-1}^{(i)}, y_t) = p(\alpha_t^{(i)}|\alpha_{t-1}^{(i)})$ in the recursion of $\widetilde{w}_t^{(i)}$ we arrive at the bootstrap filter and the recursion of $\widetilde{w}_t^{(i)}$ can be expressed by a much simpler form:

$$\widetilde{w}_t^{(i)} = \widetilde{w}_{t-1}^{(i)} p(y_t|\alpha_t^{(i)}). \tag{5.34}$$

This filter is sometimes referred to as *sampling importance resampling* (SIR) and was develop by Gordon, Salmond, and Smith [1993]. In practice we resample $\alpha_t^{(i)}$ at each time t. The weights are reset after the resampling of $\alpha_{t-1}^{(i)}$ at $w_{t-1}^{(i)} = N^{-1}$ and the normalized weights becomes

$$w_t^{(i)} = \frac{p(y_t|\alpha_t^{(i)})}{\sum_{j=1}^N p(y_t|\alpha_t^{(j)})}, \quad i = 1, \dots, N. \tag{5.35}$$

An advantage of the bootstrap filter is that it does not need a lot of storage because we reset the weight at each time and therefore it makes the calculations fast.

We now proceed with a small example where we compare the bootstrap filter to the Kalman filter described in Proposition 4.1. To illustrate an example we simulate a local lever model and apply the bootstrap filter as well as the Kalman filter. The model is defined as:

$$y_t = \mu_t + w_t \quad w_t \sim N(0, 1), \tag{5.36}$$

$$\mu_t = \mu_{t-1} + v_t \quad v_t \sim N(0, 0.5). \tag{5.37}$$

We simulate 100 observations and create a bootstrap filter with $N = 10,000$.

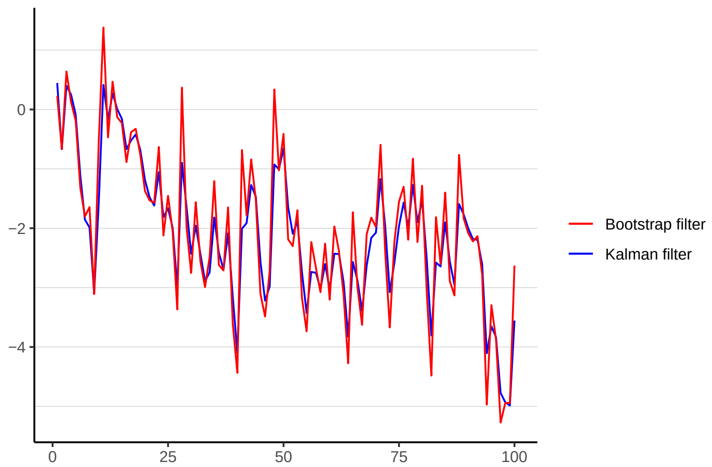


Figure 5.1: A comparison between the Kalman filter and Bootstrap filter.

We can observe in Figure 5.1 that the filtered states from the two filter method are similar. This also shows that the bootstrap method is a good alternative to the Kalman filter. Lastly we summarize the bootstrap filter by writing out the algorithm for the bootstrap filter.

Algorithm for the bootstrap filter

i) Sample α_t: draw N values $\widetilde{\alpha}^{(i)}$ from $p(\alpha_t|\alpha_{t-1}^{(i)})$.

ii) Compute $\widetilde{w}_t^{(i)}$ as:

$$\widetilde{w}_t^{(i)} = p(y_t|\widetilde{\alpha}_t^{(i)}), \quad i = 1, 2, \ldots, N, \tag{5.38}$$

and normalize the weights as in (5.35) to obtain $w_t^{(i)}$.

iii) Given a set of particles $\left\{\widetilde{\alpha}_t^{(1)}, \ldots, \widetilde{\alpha}_t^{(N)}\right\}$, compute

$$\widehat{x}_t = \sum_{i=t}^{N} w_t^{(i)} x_t(\widetilde{\alpha}_t^{(i)}). \tag{5.39}$$

iv) Draw N new independent particles $\alpha_t^{(i)}$ from $\left\{\widetilde{\alpha}_t^{(1)}, \ldots, \widetilde{\alpha}_t^{(N)}\right\}$ with replacement and with corresponding probabilities $\left\{w_t^{(1)}, \ldots, w_t^{(N)}\right\}$.

6 The UK Market

In this chapter we start by presenting the data. Next, we present our results from the implementation of the bootstrap filter, described in Section 5.2. We also present our smoothing results by maximizing the loglikelihood function, described in Section 5.1, for the dynamic Skellam model.

6.1 UK Data

The data are downloaded from `https://historicdata.betfair.com`. We chose to use data from September 2019 and only use races from the United Kingdom. In this data set there is a total of 468 races and we have selected five races to work with. These races are:

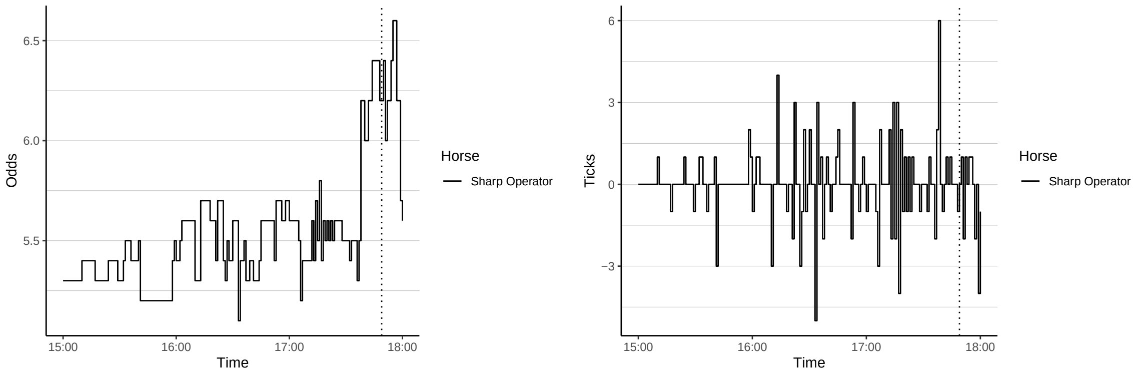
Location	Date	Start time	Horse
Chelmsford City	2019-09-24	18:00	Sharp Operator
Newcastle	2019-09-20	19:20	Fard
Pontefract	2019-09-19	14:40	Sonja Henie
Southwell	2019-09-04	15:10	Smart Getaway
Stratford	2019-09-07	16:10	Bagan

Table 6.1: The table shows which races we will work with in this chapter.

The data contains the last traded price with a frequency of one minute for each of these races. We will use the data three hours, or 180 minutes, before the races starts. Furthermore, the data is split into an in-sample and out-of-sample set where the cutoff is 10 minutes before the races starts. It is important to note that the data does not contain back and lay prices, therefore we do not know if the last traded price is a back bet or a lay bet. We will interpret the last traded price as a middle price between the back and lay prices.

Chelmsford City

Below we shown the plot of the odds trajectory of the horse *Sharp Operator*.



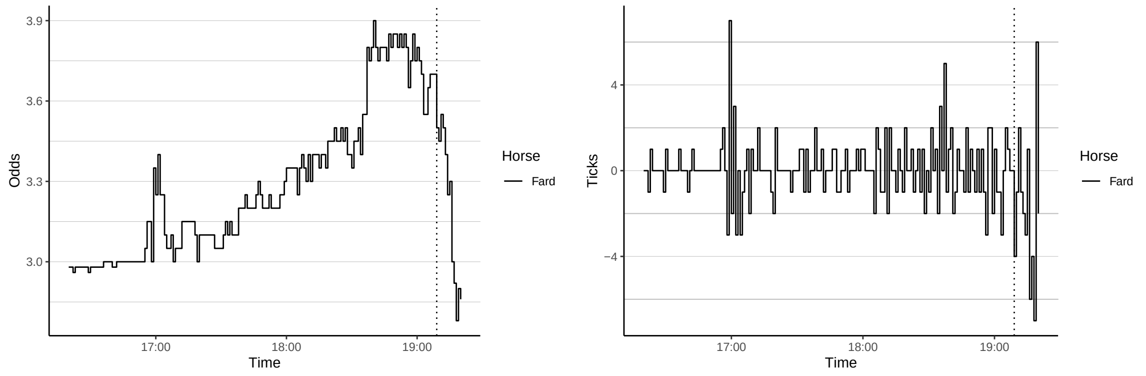
(a) (b)

Figure 6.1: The figure shows how the odds move 180 minutes before the race starts in Chelmsford City. (a) Shows the odds path before the race starts. (b) Shows the tick change each minute. The dotted line illustrates the 10 minute mark before the race starts.

The path in Figure 6.1a is fairly stationary until a large jump occurs. In Figure 6.1b we observe that this jump is 6 ticks from one minute to the next minute. A spike in the odds like this can occur if a large lay order enters the market. If this is in fact the case it explains why the price corrects itself a couple of minutes afterward. A correction like this happens when there is not enough liquidity to support this new price level. Due to the limitation of the data we can only speculate that a large lay order is the reason for this spike. In general the price swings for the favorite *Sharp Operator* moves in a relatively volatile fashion.

Newcastle

Below we have shown the plot of the odds trajectory of the horse *Fard*.



(a) (b)

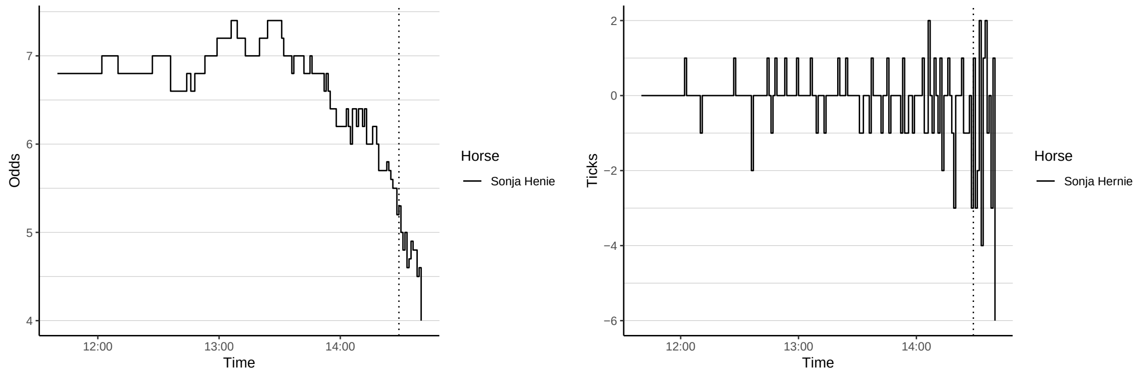
Figure 6.2: The figure shows how the odds move 180 minutes before the race starts in Newcastle City. (a) Shows the odds path before the race starts. (b) Shows the tick change each minute. The dotted line illustrates the 10 minute mark before the race starts.

The race from Newcastle is characterized by a change of trend direction. In the first 2.5 hours before the race starts the market regards the favorite *Fard* to be overpriced i.e. the market assumes that the probability of a win is too high. Therefore, the market, in general, will lay the horse and this increases the odds. All of a sudden the drift changes direction and the price is backed down again. If we investigate the race further we find that *Fard* did have very short odds compared to the other runners[1]. Furthermore, in the this race there is a total of 13 runners which can explain why there is more value in lay bets on this favorite. It is also worth noticing in Figure 6.2b how the volatility seems to increase roughly an hour before the race starts.

[1] `https://www.skysports.com/racing/results/full-result/908347/newcastle/20-09-2019/poppys-delight-handicap`, last visited 2020-04-08

Pontefract

Below we show the plot of the odds trajectory of the horse *Sonja Henie*.



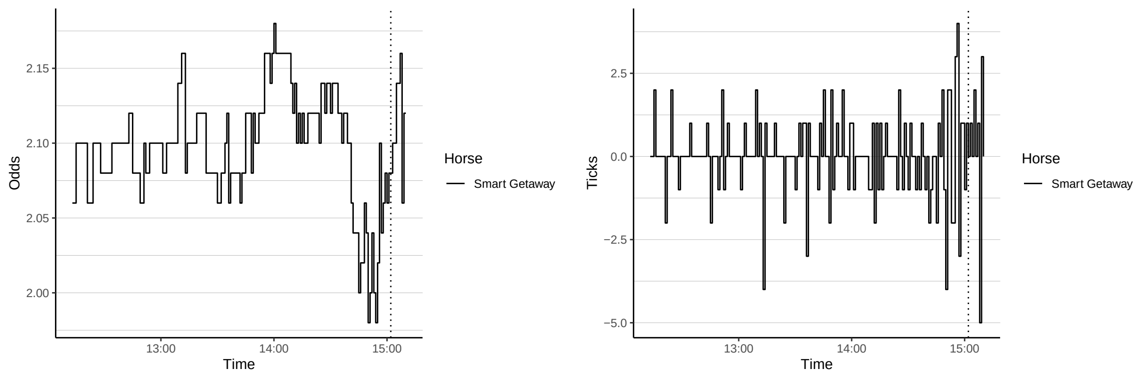
(a) (b)

Figure 6.3: The figure shows how the odds move 180 minutes before the race starts in Pontefract City. (a) Shows the odds path before the race starts. (b) Shows the tick change each minute. The dotted line illustrates the 10 minute mark before the race starts.

In this race a very clear downtrend is established. This trend suggests that the market evaluates this particular horse to be underpriced and therefore find more value in a back bet instead of a lay bet. Again, we observe how the volatility increases as the start comes closer.

Southwell

Below we shown the plot of the odds trajectory of the horse *Smart Getaway*.



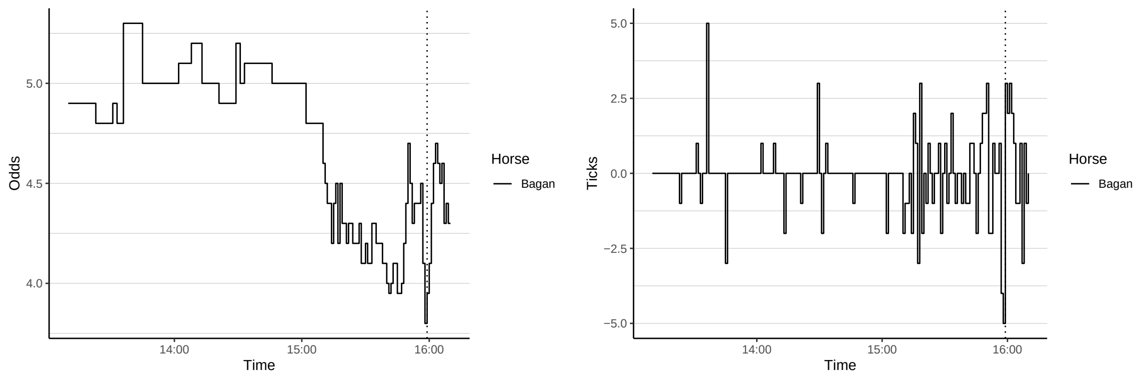
(a) (b)

Figure 6.4: The figure shows how the odds move 180 minutes before the race starts in Southwell City. (a) Shows the odds path before the race starts. (b) Shows the tick change each minute. The dotted line illustrates the 10 minute mark before the race starts.

This race seems to have a fixed trading range within all 3 hours before the race. The odds trajectory differs from the other races because it oscillates around a mean price roughly on odds of 2.08. From Figure 6.4b we observe that the volatility is almost constant throughout the time period. In this race, there is only four runners where two of them is priced at odds around 2.00 and the two remaining horses are priced at odds 17 and 51 [2]. In a race like this it is difficult to make the odds drift in either direction. Because if one of the odds adjust the remaining odds will adjust as well. In a race with two big favorites their movements tend to be negatively correlated. This means if the price of one favorite goes down then the other favorite's price moves up and creates value. When there is value the market starts backing the other favorite and thereby moves the other favorite's price down again. If the other favorite's price goes down then the first favorite's price goes up and creates value and once again the market corrects this value. We suggest that this is what happened in this race and that can be the reason for this fixed trading range.

Stratford

Lastly we present the Stratford race and illustrate the odds path of the horse *Bagan*, this odds path is shown below.



(a) (b)

Figure 6.5: The figure shows how the odds move 180 minutes before the race starts in Stratford City. (a) Shows the odds path before the race starts. (b) Shows the tick change each minute. The dotted line illustrates the 10 minute mark before the race starts.

At the beginning of this race the price moves with a relatively low jump intensity. This suggests that there is not a lot of liquidity in the market which can be justified since the race takes place on a late Saturday evening. We can only speculate about this because our data do not contain the volume of the bets. We can also observe that when the price moves it does so in a relatively volatile fashion. This is shown in Figure 6.5b where there are some large jumps, on five ticks, within one minute. Furthermore, we observe that about 1.5 hours before the race starts the price drops and the market seems to find a new price trading range.

In general we observe that the odds move more frequently as the start time comes closer. One explanation for this is how the money enters the market. In most cases the money arrives late

[2] `https://www.racingpost.com/results/61/southwell/2019-09-04/737083`, last visited 2020-04-08

into the market i.e. right before the race will begin. From the sports traders' point of view they need liquidity in order to enter and exit their trading positions. For this reason they wait until the market have enough liquidity to trade. When the liquidity is large it attracts more traders and thereby more liquidity will flow into the market. We can observe how this liquidity arrives in the market from a race in Warwick taking place *2020-02-21 15:00*. We show in Figure 6.6a the total traded volume before the race starts and also how much the volumes increase from one observation to the next in Figure 6.6b. The data is observations with two minutes apart, furthermore, it should be noted that the race is a couple of minutes delayed hence we have observations after *15:00*.

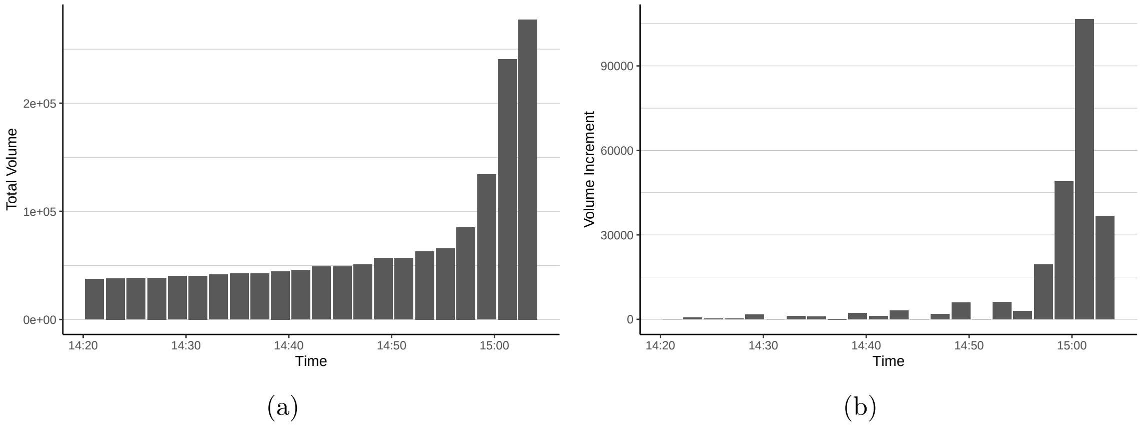


Figure 6.6: (a) Shows the total matched volume pre race at Warwick taking place *2020-02-21 15:00*. (b) Show the volume increment every *2* minutes pre race.

We will now proceed to fit the dynamic Skellam model on the presented data. In the next section we will use the filtering method from Section 5.2 and smoothing by maximizing the likelihood with the method of importance sampling described in Section 5.1.

6.2 Dynamic Skellam model Implementation

To model the presented data we will use the following dynamic Skellam model:

$$Z_t \sim Sk(\lambda_{1,t}, \lambda_{2,t})$$

$$. \begin{bmatrix} \lambda_{1,t} \\ \lambda_{2,t} \end{bmatrix} = \begin{bmatrix} \exp(\theta_{1,t}) \\ \exp(\theta_{2,t}) \end{bmatrix},$$

$$\begin{bmatrix} \theta_{1,t} \\ \theta_{2,t} \end{bmatrix} = \begin{bmatrix} c_t \\ c_t \end{bmatrix} + \begin{bmatrix} \alpha_{1,t} \\ \alpha_{2,t} \end{bmatrix},$$

$$\begin{bmatrix} \alpha_{1,(t+1)} \\ \alpha_{2,(t+1)} \end{bmatrix} = \begin{bmatrix} \phi_{1,t} & 0 \\ 0 & \phi_{2,t} \end{bmatrix} \begin{bmatrix} \alpha_{1,t} \\ \alpha_{2,t} \end{bmatrix} + \begin{bmatrix} \eta_{1,t} \\ \eta_{2,t} \end{bmatrix},$$

$$\begin{bmatrix} \eta_{1,t} \\ \eta_{2,t} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{1,t} & 0 \\ 0 & \sigma^2_{2,t} \end{bmatrix} \right).$$

We define the parameter vector as $\psi = (\sigma^2_{1,t}, \sigma^2_{2,t}, \phi_{1,t}, \phi_{2,t}, c_t)$. In the implementation of the bootstrap filter we assume that the parameters are time-varying, where in the implementation of the likelihood function we will assume the parameters are fixed throughout time.

We will now carry out our implementation of the bootstrap filter for the presented data.

Filtering Results

In this section we present our findings for using the bootstrap filter. We will illustrate how $\lambda_{1,t}, \lambda_{2,t}$ behave and how these parameters affect the mean value $\mu_t = \lambda_{1,t} - \lambda_{2,t}$ and the variance $\sigma_t^2 = \lambda_{1,t} + \lambda_{2,t}$ of the observed data. Also, we plot the cumulative sum of ticks to illustrate the odds trajectory. The data presented in this section is only the in-sample data.

Chelmsford City

In this race we observe that a large spike occurs and that the price swings are relatively volatile. We will now explore how the bootstrap filter detects these swings and spikes. Our results are shown below:

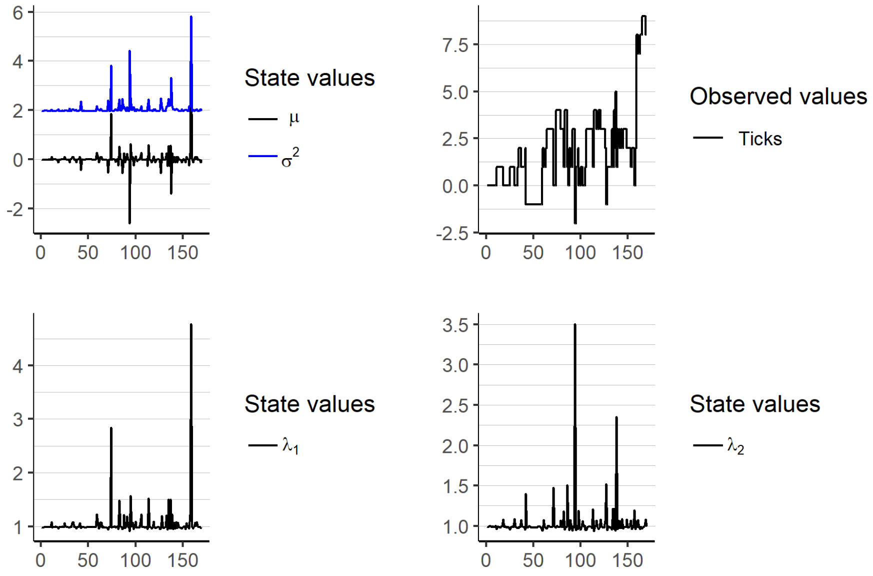


Figure 6.7: The figure shows the state values for $\lambda_{1,t}, \lambda_{2,t}, \mu_t$, and σ_t^2 as well as the cumulative sum of the observed ticks for the race in Chelmsford City.

From the results we can calculate the mean of μ_t on 0.018 per minute which indicates an upward trend. The actual trend is on average 0.047 per minute. The bootstrap filter seems to correctly estimate the trend direction.

Newcastle

The in-sample data from Newcastle have a strong upward trend for the first 3/4 of the time. In the last 1/4 the trend reverses. We will now investigate if the bootstrap filter detects these movements.

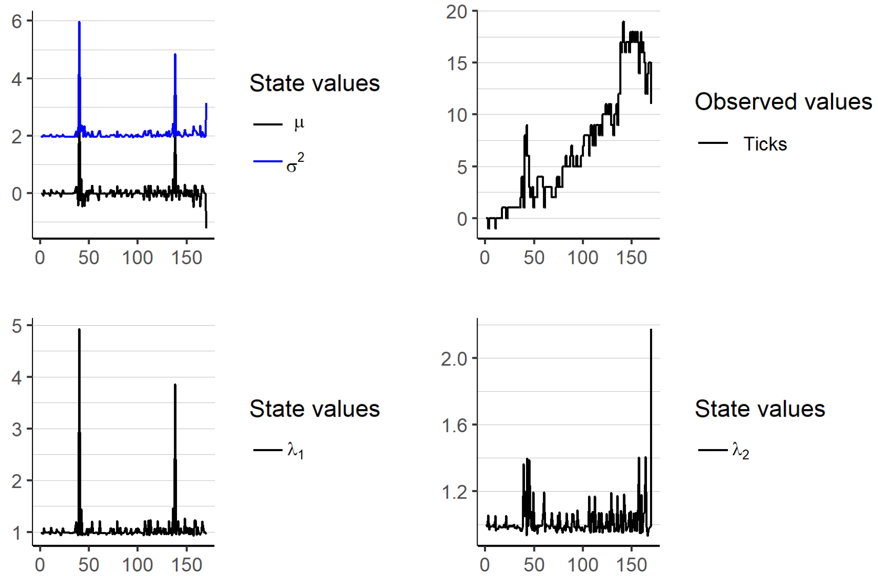


Figure 6.8: The figure shows the state values for $\lambda_{1,t}, \lambda_{2,t}, \mu_t$, and σ_t^2 as well as the cumulative sum of the observed ticks for the race in Newcastle.

First we use the 140 first values of the in-sample data set to compare the mean of the observations and the mean of $\lambda_{1,t} - \lambda_{2,t}$. The first 140 observed ticks have a mean of 0.121 ticks per minute. The state value of μ_t has an average of 0.051 for the first 140 minutes. Therefore, the bootstrap filter underestimates the true mean in the timeframe. For the observations in the period of $140 - 170$ minutes we find that the bootstrap filter overestimated the new trend. The observed mean is -0.161 and the estimated mean is -0.042. We can also observe how σ_t^2 increases at the end of the in-sample data. For the whole time period the observed mean is 0.065 and the estimated average of μ_t is 0.034. In this race we underestimate the mean value for the positive trend and overestimate the mean value for a negative trend.

Pontefract

The odds for this horse is characterized by establishing a strong negative drift in the last hour before the race begins.

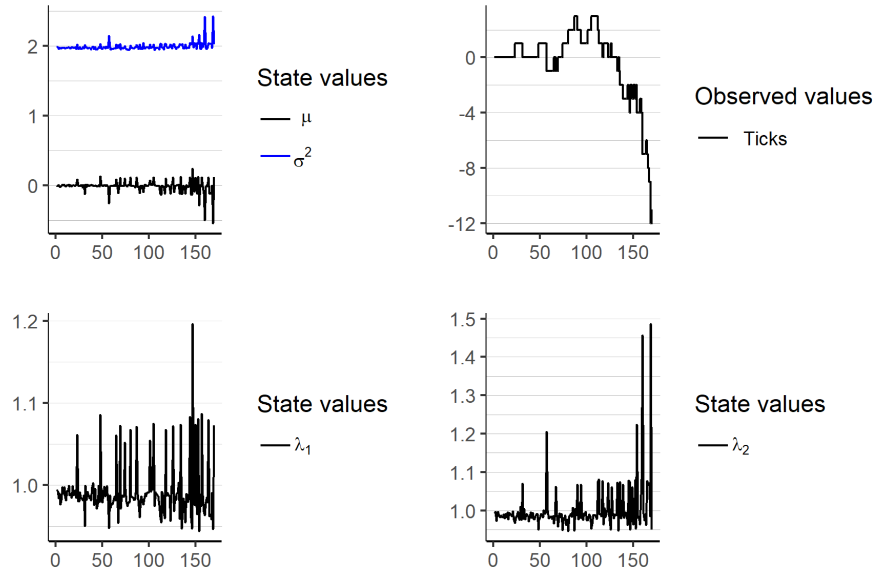


Figure 6.9: The figure shows the state values for $\lambda_{1,t}, \lambda_{2,t}, \mu_t$, and σ_t^2 as well as the cumulative sum of the observed ticks for the race in Pontefract.

The estimated state values for $\lambda_{1,t}, \lambda_{2,t}$ are fairly stable in the sense that there are no big spikes as we observed in Figure 6.8 and Figure 6.7. This also explains why the spikes in μ_t and σ_t^2 are a lot smaller. We can also observe that μ_t tends to be more negative at the end of the data set which indicates that the bootstrap filter has identified the strong negative trend.

If we split the in-sample data set after the 100 observation we can compare the observed mean and the average of μ_t. In the first part the observed mean is 0.010 and the average of μ_t is 0.001. In the second part of the sample we find the observed mean to be -0.169 and state estimate of μ_t to be -0.024. For this race we find that the bootstrap filter underestimates the negative trend. The same is true for the total of the data set, here we have an observed mean of -0.065 and an average μ_t on -0.009 per minute.

Southwell

The Southwell race is characterized by having a fixed trading range.

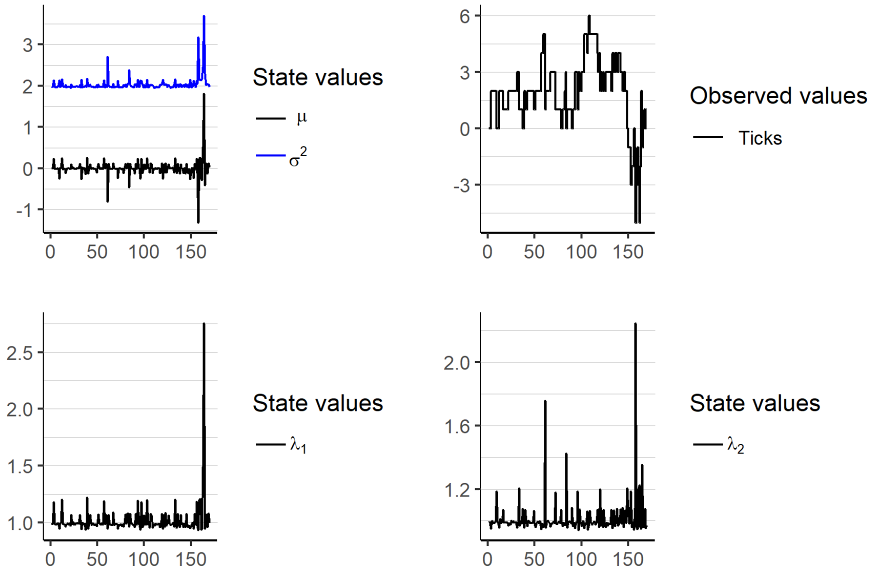


Figure 6.10: The figure shows the state values for $\lambda_{1,t}, \lambda_{2,t}, \mu_t$, and σ_t^2 as well as the cumulative sum of the observed ticks for the race in Southwell.

First, we consider $\lambda_{1,t}$ and $\lambda_{2,t}$. We observe that $\lambda_{2,t}$ is more volatile than $\lambda_{1,t}$. This suggests that when the price drops the volatility increases as well. We can observe this effect in Figure 6.10 for the state values of μ_t and σ_t^2. When μ_t is negative then σ_t^2 tend to increase.

The mean of the observed ticks is 0.006 per minute and average of the filtered μ_t is $-7 \cdot 10^{-6}$ per minute hence the bootstrap filter underestimates the trend.

Stratford

At first the Stratford race moves slowly and infrequent. As the start time comes closer the market begins to wake up and the price drops down to a lower trading range.

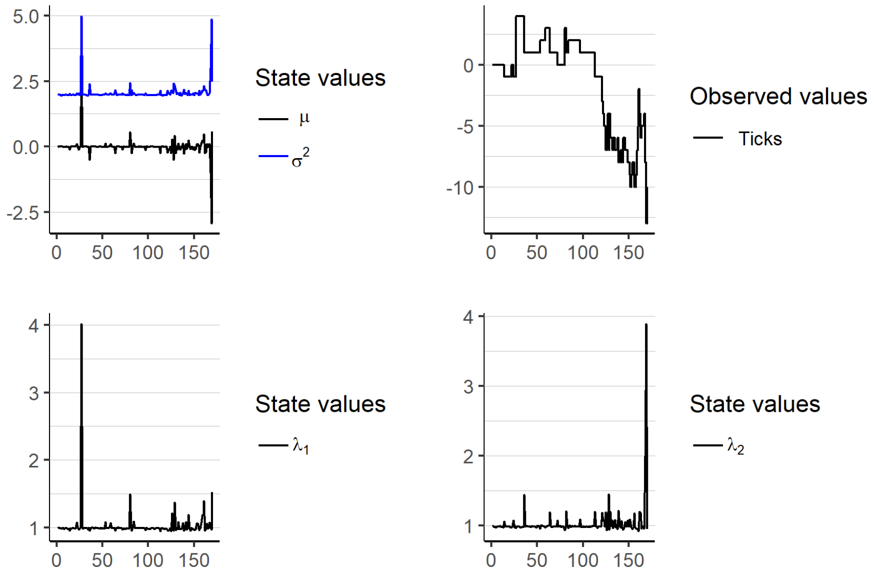


Figure 6.11: The figure shows the state values for $\lambda_{1,t}, \lambda_{2,t}, \mu_t$, and σ_t^2 as well as the cumulative sum of the observed ticks for the race in Stratford.

The state values of $\lambda_{1,t}$ and $\lambda_{2,t}$ are affected by the lack of price movements in the first two hours. They are in general very close to zero.

For this race we split the data set after the 120 observation as this is where the price drops down to a new trading range. In the first part of the training set we calculate an observed mean of -0.008 per minute and the filtered mean is 0.019 per minute on average. For the second part, the observed mean is -0.180 and the average μ_t is -0.073. For all of the 170 observations we find the observed mean to be -0.059 and the filtered mean to be -0.008.

In general we find that the absolute value of the observed mean value is greater than the filtered mean values hence the filtered values are closer to zero. However, we also find that the bootstrap filter does an adequate job of identifying the trend direction. In the next section we implement our smoothing method by maximizing the loglikelihood function of the dynamic Skellam model via importance sampling.

Smoothing Results

In this section we will maximize the loglikelihood function described in Section 5.1. By maximizing the loglikelihood function we are allowing our estimation of the parameters to be a function of the past, present, and future hence it is a smoothing method. Because our smoothing results are heavily reliant on numerical optimization we will start by conducting a sensitivity analysis of the initial values of the parameter vector. Next, we will discuss how large of a sample size that is appropriate to use when fitting the parameter values. We begin with a sensitivity analysis by investigating three initial parameter vectors:

$$\psi_{0,1} = (0.05, 0.05, 0.50, 0.50, 0.50) \quad (6.1)$$

$$\psi_{0,2} = (0.05, 0.05, 0.9, -0.9, 0.1) \quad (6.2)$$

$$\psi_{0,3} = (0.05, 0.05, 0.1, 0.5, -0.1) \quad (6.3)$$

We use these three initial parameter vectors on a sample window size of 15 minutes, 30 minutes, and 170 minutes. We begin by showing our smoothing results for the racetrack *Newcastle* with a window size of 15 minutes in the table Table 6.2.

ψ_0	σ_1^2	σ_2^2	ϕ_1	ϕ_2	c
$\psi_{0,1}$	0.748	0.104	0.249	0.372	0.480
$\psi_{0,2}$	0.830	0.222	0.520	-1.280	-0.018
$\psi_{0,3}$	0.474	0.084	-0.050	0.410	-0.226

Table 6.2: The table shows the parameter estimates for Newcastle for three different initial parameter vectors.

One concern for the estimates in Table 6.2 is for $\psi_{0,2}$, where $|\phi_2| \geq 1$. This is a concern because we know from the dynamic Skellam model that the state vector $\alpha_{2,t+1}$ is an autoregressive process formulated as: $\alpha_{2,t+1} = \phi_{2,t}\alpha_{2,t} + \eta_{2,t}$. Therefore, with $|\phi_2| \geq 1$ is not a stationary process. We find the same problem for the estimates in multiple racetracks when we used $\psi_{0,2}$ as the initial parameter vector. By expanding the window the result is still a non-stationary autoregressive process for multiple racetracks. The problem with a non-stationary autoregressive process is that the state vector can grow very large or very negative. If $\alpha_{i,t}$ is very large or very negative the link function will make $\lambda_{1,t}$ or $\lambda_{2,t}$ huge or they can become very close to zero. Such parameters do not reflect the odds movement, therefore we choose to disregard the initial parameter values of $\psi_{0,2}$.

As we concluded in Section 6.1, the market behavior changes rapidly up to the start time of a race. To capture this changing market behavior we decide to use the last 30 minutes of the in-sample data set and only use initial parameter vectors $\psi_{0,1}$ and $\psi_{0,3}$. The smoothing results for these two initial values are shown in Table 6.3 and Table 6.4.

Location	σ_1^2	σ_2^2	ϕ_1	ϕ_2	c
Chelmsford	0.5472	0.2383	0.2283	0.1878	0.5204
Newcastle	1.0499	0.2392	0.3247	0.3486	-0.0217
Pontefract	0.7258	0.1378	0.5547	0.3122	0.2304
Southwell	0.4767	0.2245	0.2729	0.5098	0.4074
Stratford	0.7712	0.1472	0.0546	0.5046	0.4194

Table 6.3: The table shows smoothing results for the last 30 minutes of the in-sample data set. The initial values for these parameters are $\psi_{0,1} = (0.05, 0.05, 0.50, 0.50, 0.50)$.

Location	σ_1^2	σ_2^2	ϕ_1	ϕ_2	c
Chelmsford	0.4961	0.1131	0.0877	0.2602	-0.2388
Newcastle	0.8286	0.4186	0.0858	0.3012	-0.5623
Pontefract	0.9164	0.2801	-0.3995	0.4203	-0.3870
Southwell	0.6210	0.0207	-0.2867	0.4106	-0.0431
Stratford	0.5585	0.0388	0.1602	0.3181	-0.1441

Table 6.4: The table shows smoothing results for the last 30 minutes of the in-sample data set. The initial values for these parameters are $\psi_{0,3} = (0.05, 0.05, 0.1, 0.5, -0.1)$.

The fact that the parameter vector estimates do not converge to the same values are not optimal. One reason for these results can be that the loglikelihood function comprises of multiple local extrema when using importance sampling hence the numerical optimization algorithm converges in different local minima. An alternative method to maximize the loglikelihood function for the dynamic Skellam model is to use *numerically accelerated importance sampling* (see Koopman, Lucas, and Scharth [2015]). This method utilizes numerical integration to determine initial parameter values. They find this method to be faster and more stable than to only rely on the importance sampling method. Unfortunately, due to time constraints, we will not proceed to investigate this method further.

When we compare the values in Table 6.3 and Table 6.4 we cannot favor one result over the other, therefore we will proceed to forecast with both sets of parameter vectors in the next section.

6.3 Forecast

In this section we present our findings for two forecasting schemes. The first scheme is to use our smoothing results discussed in the previous section. With these results we use Monte carlo simulation to forecast the out-of-sample data set. The second scheme is a combination of our filtering results and the odds trajectory model from Definition 3.7, we constructed in Chapter 3. But before we investigate how well the model can forecast the out-of-sample's odds movements we will briefly summarize the out-of-sample data set. We have plotted the odds movements as the tick paths in Figure 6.12, shown below.

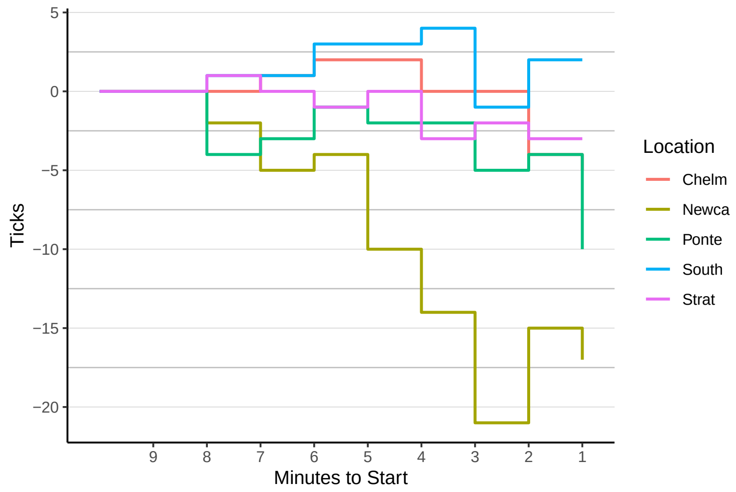


Figure 6.12: The figure shows the path for the out-of-sample data.

To give a more detailed overview we provid Table 6.5. The table shows the first odds and last observed odds in the out-of-sample set. Furthermore, we also convert the odds change into tick change. We can observe that the majority of the odds decrease in the out-of-sample data set except for Southwell.

Location	Runner	Start odds	End odds	Odds change	Tick change
Chelmsford City	Sharp Operator	6.20	5.60	-0.60	-5
Newcastle	Fard	3.50	2.86	-0.64	-17
Pontefract	Sonja Henie	5.00	4.00	-1.00	-10
Southwell	Smart Getaway	2.08	2.12	0.04	2
Stratford	Bagan	4.60	4.30	-0.30	-3

Table 6.5: The table shows the odds change as well at the tick change.

We begin by investigating how precise the dynamic Skellam model can forecast the out-of-sample data.

Forecast Results using The Dynamic Skellam Model

As we concluded in Section 6.2 we will estimate the parameter vector from the last 30 minutes of observations and use two initial parameter vectors, those initial values are:

$$\psi_{0,1} = (0.05, 0.05, 0.50, 0.50, 0.50)$$
$$\psi_{0,3} = (0.05, 0.05, 0.1, 0.5, -0.1).$$

The estimated values are shown in Table 6.3 and Table 6.4. From these parameter estimates we simulate 1,000 paths of length 10. We then collect the last values from each path and calculate three quantiles, more precisely the 2.5%, 50%, and 97.5% quantiles. The 50% quantile is our predicted value and the other two quantiles serve as a 95% prediction interval. First we present our results by using initial parameter vector $\psi_{0,3}$. The results are shown in Table 6.6 below.

Location	Forecast	P.I. 2.5%	P.I. 97.5%
Chelmsford	3	-15	28
Newcastle	4	-11	36
Pontefract	6	-13	52
Southwell	-2	-24	19
Stratford	0	-19	23

Table 6.6: The table shows the results for the forecast with initial parameters $\psi_{0,3} = (0.05, 0.05, 0.1, 0.5, -0.1)$. P.I. is short for prediction interval.

The forecasts in Table 6.6 seem to be inaccurate. The best prediction is for Stratford with an absolute error on three ticks. For all the other forecasts the sign of the total tick change is wrong hence the prediction for the trend direction is also inaccurate. Slightly improved predictions come when we use initial parameters $\psi_{0,1}$. These results are shown in Table 6.7.

Location	Forecast	P.I. 2.5 %	P.I. 97.5 %
Chelmsford	2	-8	13
Newcastle	2	-9	16
Pontefract	5	-6	25
Southwell	4	-7	18
Stratford	3	-8	15

Table 6.7: The table shows the results for the forecast with initial parameters $\psi_{0,1} = (0.05, 0.05, 0.50, 0.50, 0.50)$. P.I. is short for prediction interval.

The predictions from the table above show that all predictions are positive. The lowest absolute error is for Southwell with an absolute mean error of two ticks. Unfortunately we only generate positive forecasts which are not what we observed in the out-of-sample set seen in Table 6.5. Therefore, we conclude that this forecasting scheme insufficient.

We suggest that there are multiple reasons for these inadequate results. One of the reasons is that the method we used to maximize the loglikelihood function is insufficient as we discussed in the previous section. We will also make the argument that the assumption of fixed parameters is wrong. This argument can be supported by observing how rapidly the market can change behavior. Such an example is plotted in Figure 6.5b for the tick movements of Stratford. In this example we observe how the price suddenly dropped down and created a new trading range. Another example is the trajectory of Newcastle, Figure 6.2b. In this race the favorite establishes a clear upward trend and with seemingly no warning the trend change direction. Therefore, we suggest that in order to create more accurate predictions we need our parameters to adjust to the current state much faster than the smoothing parameters can. Our results from the bootstrap filter indicates that the state values of $\lambda_{i,t}$ for $i = 1, 2$ adjusted almost instantaneously. Using this observation we suggest an alternative forecasting scheme where we combine the filtering results and the odds trajectory model from Definition 3.7. The alternative forecasting scheme is elaborated in more detail in the next section.

Forecast using the Filter Estimations

For this next forecasting scheme we use our continuous model from Chapter 3 and it is defined as:

$$O_t = \sum_{j=1}^{N_t} Z_j, \tag{6.4}$$

where $Z_j \sim Sk(\lambda_1, \lambda_2)$ and N_t is a Poisson process with jump intensity ν. We will make λ_i for $i = 1, 2$ time-varying by replacing them with the estimates from the bootstrap filter. These parameters are calculated as an average of the latest five filtered values for $\lambda_{i,t}$ for $i = 1, 2$. Furthermore, we start by letting the expected jump intensity, ν, be 10 jumps per minute for all races. By using these parameters we simulate $1,000$ paths of length 10 from (6.4). Again, we calculate the quantiles $2.5\%, 50\%$, and 97.5% to determine the prediction for each race along with a prediction interval. We have shown these results in Table 6.8.

Location	$\widehat{\nu}$	$\widehat{\lambda}_1$	$\widehat{\lambda}_2$	Forecast	P.I. 2.5 %	P.I. 97.5 %	Tick Change
Chelmsford City	10	0.987	1.010	-2	-31	26	-5
Newcastle	10	1.042	1.211	-17	-47	13	-17
Pontefract	10	0.987	1.112	-13	-41	16	-10
Southwell	10	1.035	0.985	5	-23	33	2
Stratford	10	1.109	1.759	-65	-101	-30	-3

Table 6.8: The table shows the prediction using the odds trajectory model along with the parameters.

With this forecasting scheme we manage to predict the direction for all of the races. The predictions are within ± 3 ticks, except for Stratford. We assume that the forecast for Stratford is heavily influenced by the large spike in the state value of $\lambda_{2,t}$ seen in Figure 6.11. In general the prediction intervals are large. Also, the jump intensity, $\widehat{\nu}$, turned out to be a good guess. If we were to choose another value, such as $\widehat{\nu} = 5$ it would change our results. These results can be seen in the table below.

Location	$\widehat{\nu}$	$\widehat{\lambda}_1$	$\widehat{\lambda}_2$	Forecast	P.I. 2.5 %	P.I. 97.5 %	Tick Change
Chelmsford City	5	0.987	1.010	-1	-21	18	-5
Newcastle	5	1.042	1.211	-8	-29	12	-17
Pontefract	5	0.987	1.112	-6	-27	14	-10
Southwell	5	1.035	0.985	3	-17	22	2
Stratford	5	1.109	1.759	-33	-59	-9	-3

Table 6.9: The table shows the prediction using the odds trajectory model along with the parameters.

We can observe that the forecasts are lower, prediction intervals are more narrow, and the signs of the forecasts are still correct. This suggests that our filtering method correctly estimates the current trend for the observations. From the moments calculated in (3.8) on page 15 we know that the first moment is the expected value and is given as:

$$\xi_1 = \nu t(\lambda_1 - \lambda_2).$$

If we wish to fit ν to each race we can use the method of moments for $y_{t:t-5} = (y_t, y_{t-1}, \ldots, y_{t-5})'$ for $t = 170$. We calculate $\widehat{\xi}_1$ as the mean of $y_{t:t-5}$ and $\widehat{\lambda}_1$, $\widehat{\lambda}_2$ are the filtered values of $y_{t:t-5}$. This creates one equation with one unknown variable, namely ν, and can easily be solved. This method allows us to have more flexible jump intensities. The results are shown in Table 6.10.

Location	$\widehat{\nu}$	$\widehat{\lambda}_1$	$\widehat{\lambda}_2$	Forecast	P.I. 2.5 %	P.I. 97.5 %	Tick Change
Chelmsford City	7.052	0.987	1.010	-2	-26	22	-5
Newcastle	0.001	1.042	1.211	0	-3	3	-17
Pontefract	7.412	0.987	1.112	-9	-34	15	-10
Southwell	9.453	1.035	0.985	5	-22	32	2
Stratford	1.791	1.109	1.759	-12	-29	2	-3

Table 6.10: The table shows the prediction using the odds trajectory model along with the parameters.

By fitting individual ν to each favorite we manage to produce some good results. We can observe that the direction of the forecasts is still reliable for most of the races. However, the ν parameter for Newcastle is very close to zero and thereby it produces a very narrow prediction interval and the observed tick change is not within this interval. This result is quite curious and we cannot find a good explanation for this estimate of ν. The forecast for Stratford is improved by fitting ν to each race. We conclude that this method of forecasting produces reliable results using the minute historical data from Betfair.

Initially we started this thesis by only having access to minute data from Betfair. But due to the Coronavirus outbreak, in the spring of 2020, most European sport events have been postponed. This has given us an alternative opportunity to work with new data that have second based observations. The reason is that Betfair has released this historical data for free in the period of January to May. However, since UK races are postponed we have to use data from Australia's horse racing markets. Australia's horse racing markets have not been affected by the Coronavirus and we choose to extend our data to this market. This data does also provide us with back prices, lay prices, and volume. In the next chapter we will investigate how the spread of the prices evolve up to the start time of a race and also how the betting volume enters the market of Australian horse racing. Lastly, we will forecast the odds movements with the odds trajectory model where we use the filtered value as parameters and estimate ν with the method of moments.

The Australian Market 7

We will begin this chapter by introducing the data from the Australian racing win market. Next, we will analyze the filter values for $\lambda_{1,t}$ and $\lambda_{2,t}$ and finalize the chapter with our forecasting method and forecasting results.

7.1 Australian Data

We start by inspecting how the back and lay prices behave in the last hour before the race starts. The data show the best available back and lay prices. The timestamps for this data is in Coordinated Universal Time (UTC). We have plotted these two price paths for the favorite named *Iknow Where Youliv* in Figure 7.1.

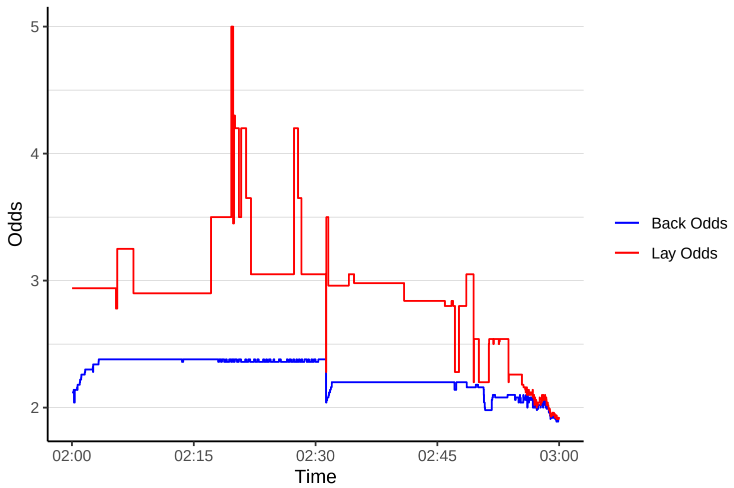


Figure 7.1: The figure shows the odds path for the horse *Iknow Where Youliv* one hour before the race starts.

The figure shows how the back and lay prices evolve up to the start of the race. We observe that most of the time there are big spreads between the back and lay prices. It is only in the last couple of minutes that the spread becomes small enough that it makes sense to trade these prices. It should be noted that these prices are not necessarily traded they are simply the best offers available in the market. If we plot the total matched volume for this horse we can observe that the volume start to increase in the last 10 minutes. This can explain why we observed such large spread in the back and lay offers.

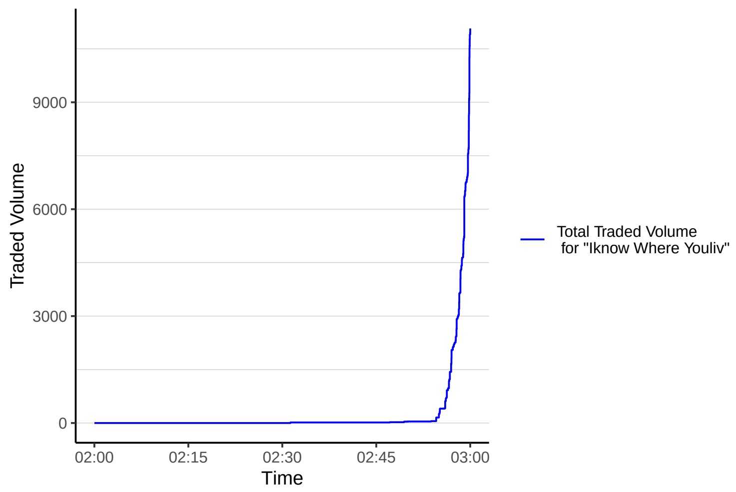


Figure 7.2: The figure shows the total traded volume for the favorite *Iknow Where Youliv* .

Due to a lack of market activity we will only use the last four minutes before the race. Furthermore, we split the data set up in two parts such that we use two minutes to forecast the next two minutes. In Figure 7.3 we show the prices in the last four minutes and the dashed vertical line indicates the border that separates the in-sample and out-of-sample data sets.

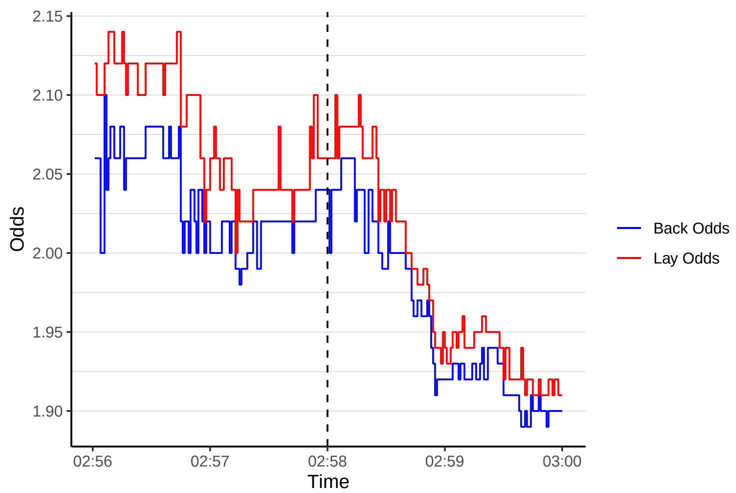


Figure 7.3: The figure shows the selected data to use from the Austrailian market. The dashed line indicates the border between the training set and the test set.

To forecast this race we decide to use the *last trade price* instead of these best available back and lay prices. We choose to do this for two reasons. Firstly, the back and lay prices are, as mentioned, only the best available prices that the market offers. This does not mean that any money is actually traded at these prices and therefore it can misrepresent the market's evaluation of the horse. Secondly, as we see in Figure 7.3 the spread is only one or two ticks apart and the last traded price can be understood as a mid price. We compare the back and lay offers to the last traded price in Figure 7.4.

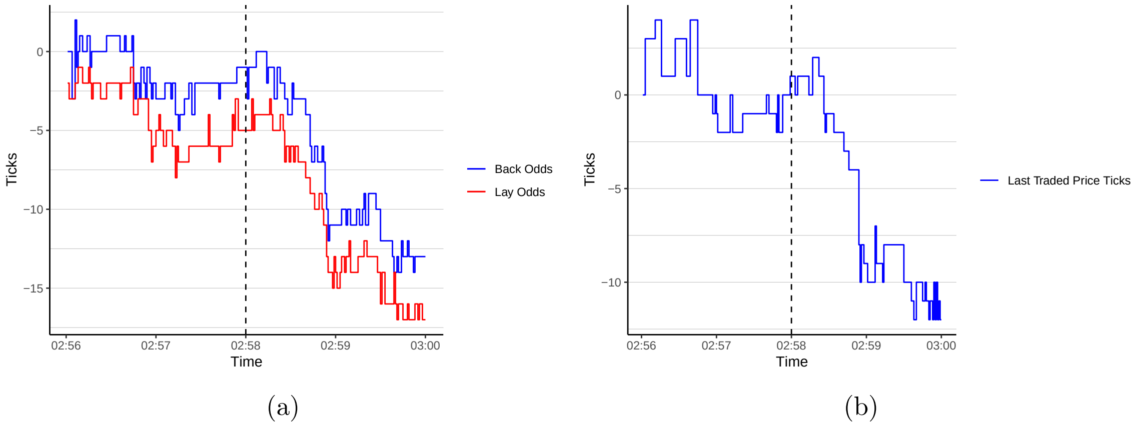


Figure 7.4: (a) This plot shows how the best available back and lay prices evolve in the last four minutes before the race starts. (b) shows how the last traded price moves in the last four minute before the race starts.

When we compare Figure 7.4a and Figure 7.4b we can observe that one disadvantage of using the last traded price is that the tick increments seams larger than the back and lay increments. However, we do not acknowledge this as a major issue. Therefore, we will now proceed to use the bootstrap filter on this selected data.

7.2 Filtering and Forecasting Results

In this section we present our findings for the Australian market by using the bootstrap filter and the forecast using the methodology of Section 6.3. We have shown the filtering results in Figure 7.5.

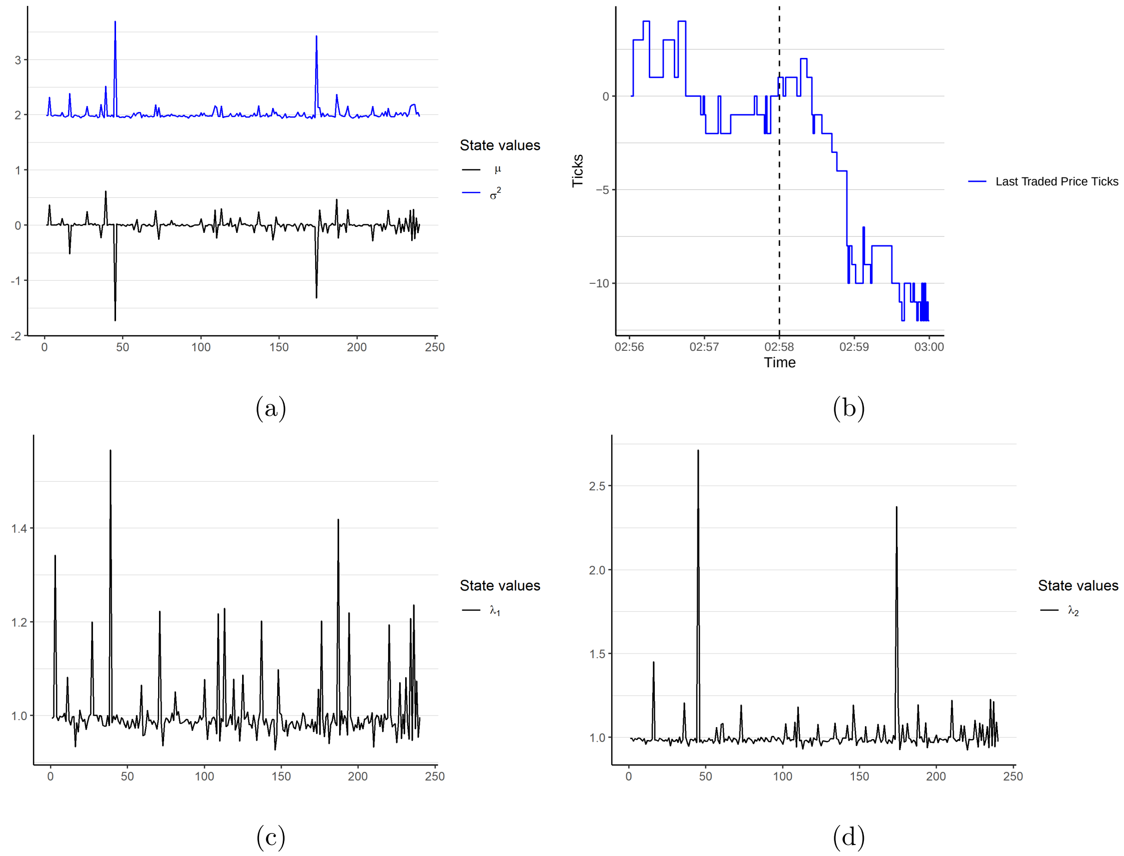


Figure 7.5: (a) This plot shows the bootstrap filter estimates of the values μ_t, and σ_t leading up to the start of the race.(b) Shows the tick increments of the last traded price. (c) Shows the state value of $\lambda_{1,t}$ and (d) shows the state value of $\lambda_{2,t}$.

If we compare the results for Figure 7.5 with the results in 6.2 we notice that the state values for $\lambda_{1,t}$ and $\lambda_{2,t}$ have fewer extreme values in the Australian data. This is as expected because the data with one second increments will move less from each observation than data with one minute increments.

To test how well this filtering method captures the trend we calculate the trend for the in-sample and out-of-sample data sets. The observed trend for the in-sample data is on average 0.0083 ticks per second and the mean of μ_t for $t = 1, \ldots, 120$ is -0.0053 per second. This finding is similar to the finding of the filtering results for the race taking place in Stratford. In the Stratford race we have similar findings when the observed trend is close to zero then the signs are opposite. For the out-of-sample we find the observed trend to be -0.1083 and the filter estimates the trend to be -0.0162. This is consistent with our findings in Section 6.2 where the absolute value of the observed mean is greater than the filtered mean value. We still find that the bootstrap filter does identify the trend direction.

We now proceed to forecast the out-of-sample. The out-of-sample have a total of -13 ticks. In order to forecast the out-of-sample data we start by calculating λ_1 and λ_2 as the mean of the filtered values if the in-sample data. By doing so, we find that $\widehat{\lambda}_1 = 1.0054$ and $\widehat{\lambda}_2 = 1.0108$. By using these values in the odds trajectory model we can estimate the jump intensity as $\nu = 59.65$

per minute. When we simulate $1,000$ paths of length 2, i.e. the length in minutes of the out-of-sample data, we forecast a tick change on -1 and a 95% prediction interval on $[-34, 31]$. We do predict the direction correctly, but the forecast is not very accurate. We assume it is due to the sudden price drop right after the test set ends. This drop do occurs as the volume is beginning to increase rapidly as seen in Figure 7.2.

To make an improved forecast we chose to discard the first of the four minutes of the data set. Furthermore, we will still use an in-sample data set of two minutes and thereby we only forecast the last minute of the race. When we investigate the traded volume for the favorite we find that the traded volume increases from \$614.39 to \$2050.13 in the discarded minute. Because the volume is so much larger in the third minute before start than in the fourth minute before start we will argue that discarding the fourth minute before start does reflect the market opinion to a greater extent. With such a big liquidity increase we assume that there are more bettors in the market and therefore we have more opinions to form the market opinion. At the last minute before start, the tick change is -3. In this new test sample, 120 to 180 seconds before the start, we find $\widehat{\lambda}_1 = 0.9991$ and $\widehat{\lambda}_2 = 1.0147$. Furthermore, we estimate $\nu = 90.83$ and we forecast a tick change in the last minute on -3 with a 95% prediction interval on $[-40, 33]$. This also indicate how fast the model can adjust to new information.

From our finding we see that the jump intensity is higher in the Australian races than in the UK races but the parameters $\lambda_{i,t}$ for $i = 1, 2$ does not differ by a noticeable amount in the two markets. There can be multiple reasons for the difference in ν. Firstly, the frequencies of the data points are different and this will affect the estimations of ν because the second based data will reflect more jumps than the minute based data. Secondly, there seems to be a big difference in how liquidity enters the two markets. We can observe that the traded volume arrives relatively late into the Australian market compared with the UK market. For this reason, our in-sample data set in the Australian is much closer to the start of the race than the UK in-sample data set. Because the market seems to become more lively up to the start of the race this can be the reason why the jump intensity is higher in the Australian in-sample data set than in the UK in-sample data set. Another explanation for this difference can be that the Australian racing market is said to be less liquid than the UK racing markets. With lower liquidity, it is much easier to move the prices. With this argument we can make the assumption that the jump intensity in general is higher in Australian horse racing than in the UK horse racing which also supports our findings. As for now we cannot give a definitive answer to how these two markets differ or how they relate, due to limited data set. We suggest that with a larger sample size of different horse racing markets it is possible to characterize each of the markets' similarities and differences. We leave this to future research.

Conclusion 8

In this thesis we built a stochastic model that could model the odds movements. We proposed a model to model the number of tick changes rather than the change in odds. The way we modeled the tick increments was with a Skellam distribution. The choice of the Skellam distribution was an ideal choice because this distribution takes on positive and negative integer values. We then derived a continuous odds model as a compound Poisson process with Skellam distributed jump sizes. Our proposed model to model the odds movements was defined as:

$$O_t = \sum_{j=1}^{N_t} Z_j,$$

where N_t is a Poisson process with parameter ν and $Z_j \sim Sk(\lambda_1, \lambda_2)$. Furthermore, we derived the moment generating function along with its first three moments. A concern about this model was that the parameters were fixed throughout time. We determined this as a major drawback for the model because we found that odds in horse racing markets could change trend direction as well as volatility as the start time of a race came closer. We also suggested that this changing market dynamic is correlated with how the traded volume arrives in the markets. In order to make the parameter time-varying we proposed a nonlinear non-Gaussian state-space model, namely a dynamic Skellam model. The dynamic Skellam model was denoted as:

$$Z_t \sim Sk(\lambda_{1,t}, \lambda_{2,t}), \quad t = 1, 2, \ldots, n.$$

The dynamics of $\lambda_{1,t}$ and $\lambda_{2,t}$ were modeled by the nonlinear transformation of the autoregressive process given as:

$$\begin{aligned} \lambda_{it} &= s_i(\theta_t), \\ \theta_t &= c_t + M_t \alpha_t, \\ \alpha_{t+1} &= d_t + T_t \alpha_t + \eta_t, \quad \eta_t \sim N(0, Q_t), \end{aligned}$$

for $i = 1, 2$ and $t = 1, \ldots, n$, where n was the length of the time series. Even though the implementation of the dynamic Skellam model made it possible to make the parameters time-varying it also gave us some challenges when it came to filtering and smoothing of this state-space model. Our smoothing method included loglikelihood maximization via importance sampling and our filtering method included the use of a bootstrap filter. We split each data set up in an in-sample and an out-of-sample data set. By having done so we attempted to forecast the last values in the out-of-sample data sets.

We used two approaches to forecast the odds movements. The first approach was conducted by using the estimated parameters from the smoothing method. With these parameters we simulated 10 steps ahead. We used the last 30 minutes of the in-sample data set to estimate the parameters for the dynamic Skellam model and with these parameters we simulated $1,000$ paths each with length 10 and used the 50% quantile as the forecast. This forecasting method did not yield satisfactory results. The second approach turned out to be much more promising. For this forecasting method we used the odds trajectory model where we estimated λ_1 and λ_2 as the average of the latest five filtered state values. Furthermore, we used the first moment to estimate the jump intensity of ν. This forecasting method showed promising results and was able to forecast the trend direction for almost all the races.

Lastly, we extended our data to the Australian horse racing market. This data was available with second based observations. We only used the odds trajectory model to forecast the tick increments for the out-of-sample data set in this market. From the data we observed that the liquidity arrived much later into this market compared to the UK races. Therefore, we only used the last three minutes until race start where the out-of-sample was the last minute until race start. The forecast in the Australian racing market also yielded good results. Furthermore, we found the jump intensity, ν, was much larger in the Australian market than the UK market but we were unable to give a definitive explanation for this large jump intensity.

Discussion/Further Research 9

Odds Trajectory Model

We constructed the odds trajectory model with a compound Poisson process and therefore the parameters were fixed. The primary challenge of this thesis was to make the jump size distribution time-varying however, we also noticed how the market activity would change as time progressed. This indicates that the jump intensity also changed with time. A relevant model extension would be to model this jump intensity as a function dependent on time. Such an extension would lead to a non-homogeneous compound Poisson process. By defining the jump intensity as a function dependent on time it could possibly lead to a better forecast and would be a relevant extension for our proposed model.

Dynamic Skellam Model

The dynamic Skellam model was a way to model the changing density for the observed tick increments. Unfortunately we did not manage to produce reliable smoothing estimates. By implementing the method of numerically accelerated importance sampling we assumed that it could lead to improved smoothing estimates. This would be an ideal way to possible improve the smoothing results.

Another improvement for the model could also be to including fundamental variables in the model e.g. ground condition, win rate, trainer, previous start odds, etc.

Market Characteristics

In the presented data we did not find many similarities for the odds paths. This could indicate that there are different market characteristics nationally as well as locally i.e. each venue might have its own market dynamics. We found that the liquidity for the UK market and the Australian market was very different. Therefore, it would be interesting to include a larger sample size with many different races for each area i.e. venues or nations, and investigate how each market behaves.

Market Extensions

In this thesis we only included data from the pre-racing win market. The models could also be extended to the place market and each way market. It could even be possible to investigate how the models would perform on the in-running market. The in-running market would probably be more reliant on fundamental variables for each horse. To implement our proposed models with a more reliable smoothing method and fundamental variables could lead to some very interesting results and practical applications.

Lastly, we would even suggest extending the models to different sports. The model could possibly model odds trajectories in in-play sports such as cricket, dart, tennis i.e. sports where the odds paths are very volatile and less deterministic. Furthermore, the Skellam distribution could also model point spreads in sports such as soccer, baseball, basketball, etc. When modeling the point spread in sports it would also make sense to have time-varying parameters and individually jump intensities. We strongly believe that the methods presented in this thesis could be a solid foundation for future research within sports betting and betting exchanges.

Bibliography

Ali, Mukhtar M (1998). "Probability models on horse-race outcomes". In: *Journal of Applied Statistics* 25.2, pp. 221–229.

Benter, William (2008). "Computer based horse race handicapping and wagering systems: A report". In: *Efficiency of racetrack betting markets*. World Scientific, pp. 183–198.

Bolton, Ruth N and Chapman, Randall G (1986). "Searching for positive returns at the track: A multinomial logit model for handicapping horse races". In: *Management Science* 32.8, pp. 1040–1060.

Bunyan, Andrew (2015). "Time Series Analysis and Forecasting of In-Play Odds on a Betting Exchange". PhD thesis. Dublin, National College of Ireland.

Durbin, James and Koopman, Siem Jan (2012). *Time series analysis by state space methods*. Oxford university press.

Gan, HL and Kolaczyk, Eric D (2018). "Approximation of the difference of two Poisson-like counts by Skellam". In: *Journal of Applied Probability* 55.2, pp. 416–430.

Gordon, Neil J, Salmond, David J, and Smith, Adrian FM (1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 2. IET, pp. 107–113.

Iacus, Stefano M (2009). *Simulation and inference for stochastic differential equations: with R examples*. Springer Science & Business Media.

Koopman, Siem Jan, Lit, Rutger, and Lucas, André (2014). "The dynamic Skellam model with applications". In:

– (2017). "Intraday stochastic volatility in discrete price changes: the dynamic Skellam model". In: *Journal of the American Statistical Association* 112.520, pp. 1490–1503.

Koopman, Siem Jan, Lucas, André, and Scharth, Marcel (2015). "Numerically accelerated importance sampling for nonlinear non-Gaussian state-space models". In: *Journal of Business & Economic Statistics* 33.1, pp. 114–127.

Ozel, Gamze (2013). "On the moment characteristics for the univariate compound poisson and bivariate compound poisson processes with applications". In: *Revista Colombiana de Estadística* 36.1, pp. 59–77.

Pedersen, Jan (2017). *Lévy Processes*. Aarhus University.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

Shumway, Robert H and Stoffer, David S (2017). *Time series analysis and its applications: with R examples*. Springer.

Snyder, Wayne W (2008). "Horse racing: Testing the efficient markets model". In: *Efficiency Of Racetrack Betting Markets*. World Scientific, pp. 269–278.

Tankov, Peter and Cont, Rama (2004). *Financial Modelling With Jump Processes*. ISBN 978-1584884132. Chapman & HALL/CRC.