# ABSTRACT

| | |
|---|---|
| Title of dissertation: | Exploring and Modeling Online Auctions<br>Using Functional Data Analysis |
| | Shanshan Wang, Doctor of Philosophy 2007 |
| Dissertation directed by: | Assistant Professor Wolfgang Jank<br>R.H.Smith School of Business<br>and<br>Associate Professor Paul J. Smith<br>Statistics Program, Department of Mathematics |

In recent years, the increasing popularity of eCommerce, and particularly online auctions has stirred a great amount of scholarly research, especially in information systems, economics, and marketing, but little or no attention has been received from statistics. ECommerce arrives with enormous amounts of rich and clean data as well as statistical challenges. eCommerce not only creates new data challenges, it also motivates the need for innovative models. While there exist many theories about economic behavior of participants in market exchanges, many of these theories have been developed before the appearance of the world wide web and often are not appropriate to be used in explaining modern economic behavior in eCommerce. This calls for new models that describe not only the evolution of a process, but also its dynamics. This research takes a different look at online auctions and proposes to study an auction's price evolution and associated price dynamics from different points of view using functional data analysis techniques.

In this dissertation, we develop novel dynamic modeling procedures applicable to online auctions. First, we develop a *dynamic forecasting system* to predict the price of an ongoing auction. By dynamic we mean that the model can predict the price of an auction "in-progress" and can update its prediction based on newly arriving information. Our dynamic forecasting model accounts for the special features of online auction data by using modern functional data analysis techniques. We also use the functional context to systematically describe the empirical regularities of auction dynamics. Second, we propose a family of *differential equation models* to capture the dynamics in online auctions. A novel multiple comparisons test is proposed to compare dynamics models of auction sub-populations. We accomplish the modeling task within the framework of principal differential analysis and functional models. Third, we propose *Model-based Functional Differential Equation Trees* to better incorporate the different characteristics of the auction, item, bidders and seller into the differential equation. We compare this new tree-method with trees either based on high-dimensional multivariate responses or functional responses. We apply our methods to a novel set of Harry Potter and Microsoft Xbox data for model validation and comparison of method.

# Exploring and Modeling Online Auctions using Functional Data Analysis

by

**Shanshan Wang**

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Commmittee:

Assistant Professor Wolfgang Jank, Co-chair/Advisor
Professor Paul J. Smith, Co-Chair/Co-advisor
Professor Benjamin Kedem
Assistant Professor Galit Shmueli
Professor Michel Wedel

To Nathan, Min and My Parents

# ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Dr. Wolfgang Jank, for his advice, encouragement, and support on research. It was him who ignited my interest in the fascinating world of functional data analysis, helped me to improve my writing and presentation, and provided me opportunities and freedom to enrich my knowledge.

I am grateful to my co-advisor, Dr. Paul J. Smith, for his advice and support on research. During my past years in Department of Mathematics, I benefited a lot from his academic guidance and teaching.

I am also grateful to Dr. Galit Shmueli, for her kind guidance and advice, especially on FDA applications to online auctions. She is another advisor in my mind.

I would like to thank the other members of my dissertation committee, Dr. Benjamin Kedem and Dr. Michel Wedel for their kind support and guidance during the past years.

I would also like to thank all the faculty and staff in Department of Mathematics, especially in Statistics Program. Many thanks to my dear friends in Department of Mathematics and Aerospace. It was they who made my life in past years so enjoyable.

My deep gratitude goes to my parents. Without their continuous encouragement and understanding, and unconditional support, it would have been impossible

for me to accomplish all of my study and research.

Finally, I would like thank my husband, Min Xue. His love and encouragement is my endless power to keep going ahead.

# TABLE OF CONTENTS

[1] Forthcoming in the Journal of Business and Economic Statistics

[2] Submitted to the Journal of the American Statistical Association

[3] Paper in preparation

## LIST OF TABLES

LIST OF FIGURES

# Chapter 1

# Introduction

## 1.1 Background: Electronic Commerce

Electronic commerce (also referred to as EC, e-commerce or eCommerce) consists primarily of the distributing, buying, selling, marketing, and servicing of products or services over electronic systems such as the internet and other computer networks. The information technology industry might see it as an electronic business application aimed at commercial transactions. The meaning of the term "electronic commerce" has changed over the last 30 years. Originally, "electronic commerce" meant the facilitation of commercial transactions electronically. Today, it encompasses a very wide range of business activities and processes, from e-banking to offshore manufacturing to e-logistics.

Electronic commerce continues to grow at an impressive pace despite widely publicized failures by prominent online retailers. According to Forrester Research, electronic commerce generated retail sales worth about US $165 billion in 2005, a 20% lift over 2004 (*www.forrester.com*).

Online auctions are one of the most successful forms of electronic commerce. On any given day there are several million items, dispersed across thousands of categories, for sale on the web behemoth *eBay*. In the full year 2005, 1.9 billion items were listed for sale on *eBay* alone, a 33 percent increase over the previous year. This generated gross merchandise sales of $ 44.3 billion, up from $34.2 billion in 2004. *eBay*'s popularity among the public can also be evidently quantified in the following numbers: In 2005 alone, the cumulative confirmed registered users totaled a record 180.6 million, which was a 33% increase over the 135.5 million users reported at the end of 2004; and *eBay* hosts approximately 383,000 stores worldwide, with approximately 212,000 stores hosted in the U.S. alone. According to the Forrester Technographics survey, close to 30% of all US households had bid in an *eBay* online auction in 2004.

The dominant auction format on *eBay* is a variant of the second price sealed-bid auction [59] with "proxy bidding". This means that individuals submit a "proxy bid", which is the maximum value that they are willing to pay for the item. The auction mechanism automates the bidding process to ensure that the person with the highest proxy bid is in the lead of the auction. The winner is the highest bidder and pays the second highest bid (plus an increment). Unlike other auctions, *eBay* has strict ending times, ranging between 1 and 10 days from the opening of the auction, as determined by the sellers. *eBay* posts information on closed auctions for a duration of at least 15 days on its web site (see *http://listings.ebay.com/pool1/listings/list /completed.html*). These publicly available postings make *eBay* an invaluable source of rich bidding data.

## 1.2 Statistical Challenges of eCommerce Data

This dissertation addresses statistical challenges associated with eCommerce research. Electronic commerce is a growing field of scholarly research especially in information systems, economics, and marketing, while surprisingly, it has only received little attention in statistics. ECommerce provides researchers an enormous amount of data and data-driven questions and problems. While eCommerce tends to generate very rich and clean data which is structurally different from offline data, it also arrives with many new data- and model-related challenges.

One of the main challenges of eCommerce data is the combination of longitudinal information (time series data) with cross-sectional information (attribute data). A sample of $n$ records in eCommerce data consists of $n$ time series and a set of $n$ attributes. Take *eBay*'s online auctions as an example. On *eBay*, each auction is characterized by a series of bids placed over time and a set of additional auction attributes such as the opening price, the item condition, the auction duration, a seller's rating, whether or not a secret reserve price has been set up, etc.

Besides the combination of longitudinal and cross-sectional information, another typical aspect of eCommerce data is the unequal spacing of events. In traditional statistics, times series are typically recorded at pre-defined and equidistant time-points at scales of days, months, quarters or years. For eCommerce data, however, this is not true. In eCommerce, different users or agents will access the web at different points in time and different geographical locations, and user behavior has a strong influence on the events that constitute the observed time series. Consequently, the resulting times when new events arrive are extremely unevenly spaced. Furthermore, the number of events in eCommerce data within a short period of time can sometimes be very sparse and other times be extremely dense due to psychological, economic or other reasons. For instance, in online auctions, data (sequences of bids) arrive in very unevenly-spaced time intervals, determined by the bidders and their bidding strategies, and bidding in these auctions often tends to be concentrated at the beginning and especially at the end. Since many traditional statistical methods assume that data arrive in evenly-spaced time intervals, irregularly spaced data is very challenging.

eCommerce not only creates new data challenges, it also motivates the need for innovative models. While there exist many theories about economic behavior of participants in market exchanges, many of these theories have been developed before the appearance of the world wide web and often are not appropriate to be used in explaining modern economic behavior in eCommerce. For instance, there exists quite a lot of empirical research that shows that online behavior deviates in many ways from offline behavior and from what is expected by economic theory. This calls for new models that describe not only the evolution of a process, but also its dynamics. Modeling dynamics is very important in the sense that they not only greatly affect the outcomes of eCommerce activities, but also show huge heterogeneity. As we can see in online auctions, while the patterns of the price evolutions could be not very illuminating, the corresponding dynamics could show tremendous heterogeneity. Changing dynamics are inherent in a fast moving environment like the online world. Fast movements and changes imply nonstationarity, which poses challenges to traditional times series modeling. Most approaches to date tend to ignore this dynamic information and treat data as cross-sectional, by aggregating over the temporal dimension. And studies investigating bidding regularities also tend to be limited to reporting summary statistics. Such approaches lead to a great loss in information. This research takes a different look at online auctions and proposes to study an auction's price evolution and associated price dynamics, and investigate the empirical regularities in eBay's auction dynamics.

Lastly, eCommerce typically arrives with huge databases which can put a computational burden on users' storage and processing facilities. This burden is often intensified by the complicated structure of eCommerce data. The high-dimensional feature of the data usually makes multivariate techniques perform in a way that is very clumsy and unsuccessful. In this research, we investigate several new and innovative approaches to overcome these challenges.

## 1.3 Contributions of this Dissertation

In the following we discuss the research problems and methodological innovations addressed in this dissertation.

- Dynamic price forecasts for online auctions: On any given day, there are many different auctions for the same or similar item available on eBay. One of the problems associated with such an information flood is how bidders and buyers can make informed decisions: when to bid and how much to bid, or from the seller's point of view, when to set up an auction and how to set up an auction. A useful tool in this context could be a forecasting method that, at any point in time, accurately predicts the price of an auction. Forecasting price in online auctions can thus have benefits to different auction parties, and forecasting an auction in operation can be even more interesting and has more tangible benefits. As we addressed in the previous section, eCommerce data is generally characterized by the combination of longitudinal and cross-sectional information. As a typical form of eCommerce data, online auction data also carries such features. Due to the difficulty caused by these features, most of the studies to date focusing on forecasting online auction price are static in nature and does not account for information that becomes available after the start of the auction. In other words, these studies only use the cross-sectional information to forecast the final prices of online auctions, ignoring the longitudinal information when an auction is in progress. The present research successfully overcomes this data challenge with the great aid of functional data analysis. We interpret longitudinal process data as functional observations (i.e., continuous curves) and interpret cross-sectional data as functional attributes, thereby wedding both types of information and making both of them contribute to forecasting prices. This dissertation develops a dynamic forecasting system which can predict the price of an auction "in-progress" and can update its prediction based on newly arriving information. The dynamic nature of our forecasting approach is founded within the framework of functional data analysis (FDA). This is discussed in Chapter 4. This work has been accepted by the Journal of Business and Economic Statistics [105].

- Modeling dynamics of online auctions: Recent research provides more and more evidence that the price of an auction is not only determined by the a priori calculations of all bidders, but it is also affected by what happens *during* the auction. While we are not able to observe many of the underlying factors that drive bidders' behavior, auction dynamics capture many of their effects. Meanwhile, there is evidence that dynamics vary from auction to auction. Therefore the need of characterizing price process with respect to auction-related characteristics is motivated. For that reason, we develop a formal machinery to capture and model online auction dynamics and to characterize the price process, namely using differential equation models. We propose a family of linear differential equations to directly model online auction dynamics, and also propose a novel test to compare multiple dynamic models for several sub-populations of online auctions. This work has been submitted to the Journal of the American Statistical Association and is under review; see [106] or *http://www.smith.umd.edu/faculty/wjank/Wang-Jank-Shmueli-Smith-PDA_of_Online_Auctions.pdf.*

- Functional differential equation trees: Our previous work has shown that dynamics in online auctions differ a lot based on auction sub-populations. In order to capture these differences, we propose a new functional differential equation tree to incorporate covariate information into functional differential equations. There is a large set of auction-related characteristics that are important factors related to the price dynamics, but incorporating covariate information into differential equations is not obvious. In order to tackle the problem, we propose an elegant partitioning-based approach. A manuscript based on this work is currently in preparation and it is expected to be submitted by the end of May.

## 1.4 Organization of the Dissertation

The organization of this dissertation is as follows:

In Chapter 2, we review previous work on online auction research in economics, information systems and statistics. The statistical review includes areas such as functional data analysis techniques, differential equation models, and tree-structured models.

In Chapter 3, we give a description of the data used in this research: the data availability, how we obtained the data, and a discussion of the data details.

In Chapter 4, we present our initial systematic study of the empirical regularities in online bidding dynamics. We develop a dynamic forecasting system for the price curves of on-going online auctions. We apply the method to our eBay data and compare with the results from traditional methods. A sensitivity analysis is presented to study robustness of the model to changes in knot allocation and with respect to the choice of smoothing parameter.

Chapter 5 focuses on the differential equation models. We model the price curves and capture the dynamics in online auctions using differential equation models. A novel test is presented to compare multiple dynamic models for auction sub-populations. We accomplish the modeling task within the framework of principal differential analysis and functional data analysis.

In Chapter 6, we give a brief overview of tree techniques and associated terminologies for univariate, multivariate, and functional responses. A brief introduction is also given to model-based recursive partitioning method from which our method extend. We employ a tree-structured model to better embed the influences of cross-sectional factors. A functional-tree framework based on differential equation models is presented and applied to our eBay data. The results are compared and contrasted with those from the methods mentioned above.

Chapter 7 concludes this thesis and discusses future work.

# Chapter 2

# Literature Review

## 2.1 Empirical Studies on eCommerce and Online Auctions

Electronic commerce, and in particular online auctions, have received an extreme surge of popularity in recent years. While a large amount of extensive research on classical auction theory (see [66] and [56] for an introduction and overview) has been conducted, most research has been focused on a game-theory perspective, and the involvement of statisticians in the field is scarce. Empirical research in this area can be seen for example in [38]. The lack of previous statistical research in the field is most likely due to the absence of widely available data. However, the recent surge of online auctions and the capability of collecting data conveniently over the internet have made more and more bidding data become available. The popularity of online auctions and the easy data availability online have stirred a great number of empirical studies in economics and information systems. For instance, [11] uses clustering analysis and finds that significant heterogeneity exists in the users of electronic markets like eBay and develops a stable taxonomy of bidding behavior in online auctions. The determinants of bidder and seller behavior are also explored by [9] using regression models. Moreover, an interesting phenomenon is the winner's curse. With the existence of a common value, the winner's curse occurs when bidders are not aware that they will only win the auction when they have the highest evaluation of the product and as a consequence, inexperienced bidders frequently overpay. This is seen as manifestation of informational asymmetry in electronic markets [10]. A structural econometric model of bidding to measure the extent of the winner's curse is used in [9]. Feedback mechanisms are a popular feature of online auctions and can decrease the informational asymmetries between buyers and sellers. Proper feedback mechanisms can induce trust and trust can reduce information asymmetry by reducing transaction-specific risks [8]. A detailed discussion of important differences between internet-based feedback mechanisms and traditional "word-of-mouth" networks can be found in [22]. That author also surveys important issues related to design, evaluation and use of online feedback mechanisms. In an investigation of the determining factors of price, [88] finds that, via regression models, a seller's feedback rating has a measurable effect on auction prices, with a few negative ratings having a much greater impact than many positive ratings. They also find that the magnitude of the opening bid and the use of secret reserve prices tend to have a positive effect on the final auction price. Other empirical work observes the prevalence of "bid sniping" in eBay's auctions (see [90], [10], [88]). Bidders hold back their bids as long as possible, resulting in a huge amount of bids placed in the last moments of the auction. Last-moment bidding may be a response by rational bidders against naive bidders or a form of "tacit collusion" by the bidders against the seller. Generally, bidders feel that they increase their chances of winning by revealing their valuation as late as possible during the auction. Despite the prevalence of bid sniping, "early bidding" also exists. People may bid early to establish their time priority on multiunit auction sites like Ubid.com or perhaps to assess their competition [11]. In either case, both “late bidding” and “early bidding” indicate that the dynamics change tremendously over the course of an online auction.

While online auctions experience an increasing amount of interest in the economics and information systems literature, relatively little work, with a few recent exceptions, has been done from a statistical point of view. To deal with the overwhelming amount of data found on auction sites like eBay.com, [95] introduces graphical methods such as profile plots and statistical-zooming to visualize online auctions in an informative way. Their visualizations allow for a straightforward inspection of bidding heterogeneity, manifested in “early bidding” and “sniping”. Modeling bid arrivals during an auction, [96] introduces a class of 3-stage nonhomogenous Poisson processes to describe the heterogeneous stages of bid arrivals within a finite time period. Furthermore, recent study by [50] proposes the use of modern statistical methods, in particular functional data analysis, to investigate the dynamics of the price process rather than just looking at the auction statically. The authors utilize functional cluster analysis and find that the price-dynamics, like the price-velocity and price-acceleration, can be quite different for different auctions. In an extension of that work, [93] employs functional regression to investigate the effect of covariates like the opening bid on the dynamics of the auction. Interestingly, it was found that during the beginning of the auction, high opening bids are associated with faster acceleration in the bidding process while towards the auction end, high opening prices are associated with a slow-down of the price-dynamics. All of these previous observations have been made using a modern statistical methodology, called Functional Data Analysis (FDA). In the following, we review the basics of FDA in detail.

## 2.2 Functional Data Analysis

Methodological and applied research related to the analysis of functional data is currently receiving a tremendous amount of interest in the statistics literature. Functional data analysis (FDA) is a tool set that, although based on the ideas of classical statistics, differs from it (and, in a sense, generalizes it), especially with respect to the type of data structures that it encompasses. While the underlying ideas for FDA have been around for a longer time, the surge in associated research can be attributed to the monographs of [85, 86]. In FDA, the interest centers around a set of curves, shapes, images, or, more generally, a set of *functional objects*. There is a number of recent studies devoted to the generalization of standard statistical methodology to the context of functional observations. For instance, [31] develops a measure of centrality for a given functional observation within a group of curves. A principal component approach for a set of sparsely-sampled curves is developed in [43] (see also [70]). Other exploratory tools have been developed such as curve-clustering (see [1, 47, 103]) and curve-classification (see [36, 44]). Classical statistical methods have also been generalized to functional canonical correlation analysis [37], functional ANOVA [28, 34], functional regression [29, 107, 20, 77], and functional generalized linear models [46, 76]. Differential equation models are fitted to data of functional form in [83] (see also [84]). This list is only a small part of the current methodological efforts in this emerging field.

Functional data analysis has been applied to many areas, such as the agricultural sciences [73], the behavioral sciences [89] as well as medical research [74]. The method has been applied to a wide range of areas from analyzing the dynamics of seasonally-varying production indices [84] to predicting El Niño [14]. However, while there exist many more applications in which functional data methods have been fruitful, it appears that this set of tools has not yet been explored extensively to analyse price behavior in online auctions or data originating from electronic commerce. One exception is the recent work of [50], which applies functional clustering to bid histories of eBay auctions to differentiate main clusters. Along that same stream of research, [52] uses functional regression analysis to explore process dynamics in eCommerce like eBay online auctions (see also [87] for examples of exploring bid dynamics in auctions for modern Indian art via FDA). State-of-the-art functional data methodology is proposed in [51] for directly modeling temporal bidding information and its dynamic change. These examples prove that FDA is slowly finding its way into the empirical exploration and modeling of online auction data. More discussion on the versatility of this tool set in the broader context of electronic commerce research can be found in [51]. In this dissertation, we set out to create a formal platform for investigating eCommerce data using FDA. In particular, we set out to create a platform for investigating dynamics of eCommerce transactions using FDA.

# Chapter 3

# Online Auction Data and Their Pre-Processing

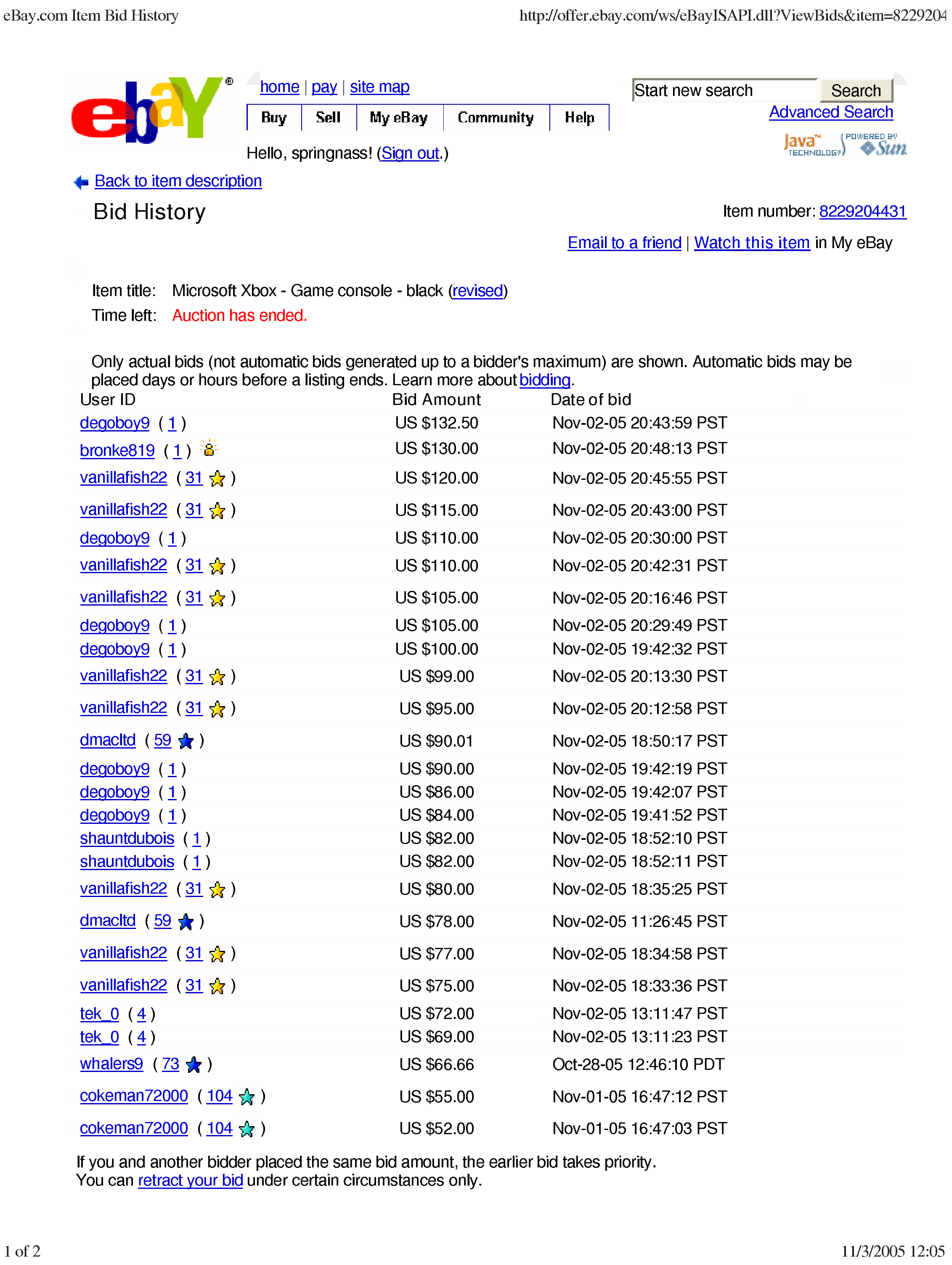
## 3.1 Data Availability

As we mentioned in Chapter 1, *eBay* posts information of closed auctions for a duration of at least 15 days on its web site. These publicly available postings make *eBay* an invaluable source of rich bidding data.

A typical bid history for a closed auction (see e.g., Figure 3.1) includes information about the magnitude and time when a bid was placed. Additional information that is made available includes information about the seller and the bidders (e.g., username, feedback ratings), information about the item sold (e.g., name, description), and information about the the auction format (e.g., auction duration, magnitude of the opening bid). In the following, we will give a brief introduction to collect online transaction data.

## 3.2 Data Collection

Every day on eBay, there are several million items for sale, which means that large amounts of data are available. While such data could at least in principle be collected "manually" by simply browsing through individual web pages, in practice this can be very time consuming, and therefore data are often collected automatically



eBay.com Item Bid History http://offer.ebay.com/ws/eBayISAPI.dll?ViewBids&item=8229204

home | pay | site map

Buy | Sell | My eBay | Community | Help

Start new search | Search

Advanced Search

Hello, springnass! (Sign out.)

Back to item description

Bid History

Item number: 8229204431

Email to a friend | Watch this item in My eBay

Item title: Microsoft Xbox - Game console - black (revised)

Time left: Auction has ended.

Only actual bids (not automatic bids generated up to a bidder's maximum) are shown. Automatic bids may be placed days or hours before a listing ends. Learn more about bidding.

| User ID | Bid Amount | Date of bid |
|---|---|---|
| degoboy9 ( 1 ) | US $132.50 | Nov-02-05 20:43:59 PST |
| bronke819 ( 1 ) | US $130.00 | Nov-02-05 20:48:13 PST |
| vanillafish22 ( 31 ☆ ) | US $120.00 | Nov-02-05 20:45:55 PST |
| vanillafish22 ( 31 ☆ ) | US $115.00 | Nov-02-05 20:43:00 PST |
| degoboy9 ( 1 ) | US $110.00 | Nov-02-05 20:30:00 PST |
| vanillafish22 ( 31 ☆ ) | US $110.00 | Nov-02-05 20:42:31 PST |
| vanillafish22 ( 31 ☆ ) | US $105.00 | Nov-02-05 20:16:46 PST |
| degoboy9 ( 1 ) | US $105.00 | Nov-02-05 20:29:49 PST |
| degoboy9 ( 1 ) | US $100.00 | Nov-02-05 19:42:32 PST |
| vanillafish22 ( 31 ☆ ) | US $99.00 | Nov-02-05 20:13:30 PST |
| vanillafish22 ( 31 ☆ ) | US $95.00 | Nov-02-05 20:12:58 PST |
| dmacltd ( 59 ★ ) | US $90.01 | Nov-02-05 18:50:17 PST |
| degoboy9 ( 1 ) | US $90.00 | Nov-02-05 19:42:19 PST |
| degoboy9 ( 1 ) | US $86.00 | Nov-02-05 19:42:07 PST |
| degoboy9 ( 1 ) | US $84.00 | Nov-02-05 19:41:52 PST |
| shauntdubois ( 1 ) | US $82.00 | Nov-02-05 18:52:10 PST |
| shauntdubois ( 1 ) | US $82.00 | Nov-02-05 18:52:11 PST |
| vanillafish22 ( 31 ☆ ) | US $80.00 | Nov-02-05 18:35:25 PST |
| dmacltd ( 59 ★ ) | US $78.00 | Nov-02-05 11:26:45 PST |
| vanillafish22 ( 31 ☆ ) | US $77.00 | Nov-02-05 18:34:58 PST |
| vanillafish22 ( 31 ☆ ) | US $75.00 | Nov-02-05 18:33:36 PST |
| tek_0 ( 4 ) | US $72.00 | Nov-02-05 13:11:47 PST |
| tek_0 ( 4 ) | US $69.00 | Nov-02-05 13:11:23 PST |
| whalers9 ( 73 ★ ) | US $66.66 | Oct-28-05 12:46:10 PDT |
| cokeman72000 ( 104 ☆ ) | US $55.00 | Nov-01-05 16:47:12 PST |
| cokeman72000 ( 104 ☆ ) | US $52.00 | Nov-01-05 16:47:03 PST |

If you and another bidder placed the same bid amount, the earlier bid takes priority.
You can retract your bid under certain circumstances only.

1 of 2 11/3/2005 12:05

Figure 3.1: Partial bid-history for an eBay Palm-515 auction. On the left-most side of the table we can see a bidder's username, followed by the bidder's rating. The stars indicate that this eBay member has achieved 10 or more feedback points. The amount and time of the bids appear on the right.

using so-called web agents or web crawlers.

Web crawlers are software programs that visit a number of pages automatically and extract (or "parse") the required information. That way, high quality information on a large number of auctions can be gathered in a short period of time. In general, crawlers start with a list of URLs to visit, called the seeds. And the list of URLs will be recursively visited based on a certain schedule. In our context, we focus on the bid information of one specific item. Thus, we will be crawling a URL instead of a list. Since it is often difficult to retrace the bid history after the end of the auction, we have to make our crawler recursively visit the destination before the auction ends. Of course, it is good practice to keep the frequency as low as possible in order to avoid overloading the opposite server. With this in the back of our minds and using a basic crawling package created by Dr. Gove N. Allen (see [2] and http://www.gove.net for details.), we wrote two short programs. Sample scripts used for our eBay data collection are provided in Appendix A. The scripts collect eBay bid information and bid histories during 2005.

## 3.3 Data Used in this Study

The data used in this study are 190 7-day auctions of *Microsoft Xbox* gaming systems and *Harry Potter and the Half-Blood Prince* books. The data were obtained via the web crawler described previously during the months of August and September of 2005. Xbox systems are popular items on eBay and had a market price of $179.98 (based on Amazon.com). Harry Potter books are also very popular items and sold for about \$27.99 on Amazon.com. We can thus consider Xbox systems high-valued items and can compare results to the lower-valued Harry Potter books.

For each auction in our dataset we collect the bid history which reveals the temporal order and magnitude of bids, and which forms the basis of our models and the cross-sectional data. Figure 3.2 shows a scatterplot of the bid history for a typical auction. We can see that bids arrive at very irregularly spaced time intervals. While the number of incoming bids is sparse during some periods of the 7-day auction (especially in the middle), it can be very dense at other times such as at the very beginning and especially at the auction-end. Figure 3.3 shows the scatterplot of bids, aggregated over all of our 190 auctions. Note that most of the bids arrive in the last minutes of the auction, which, as we have pointed out earlier, is a typical feature of eBay's auctions.

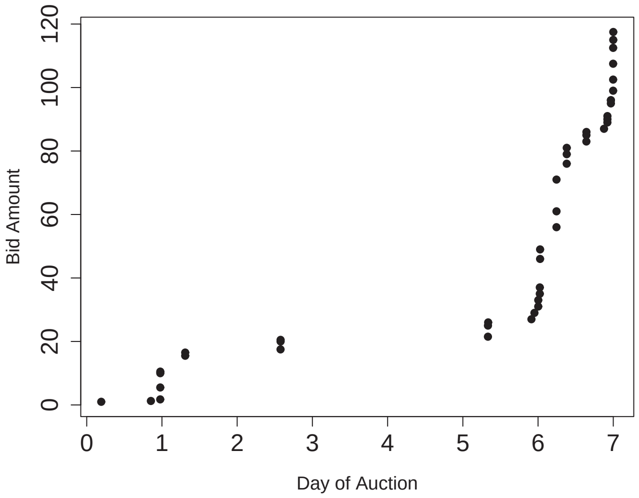


Figure 3.2: The bids placed in auction number 75 of a Microsoft Xbox auction. The horizontal axis denotes time (in days); the vertical axis denotes bid amount (in \$).

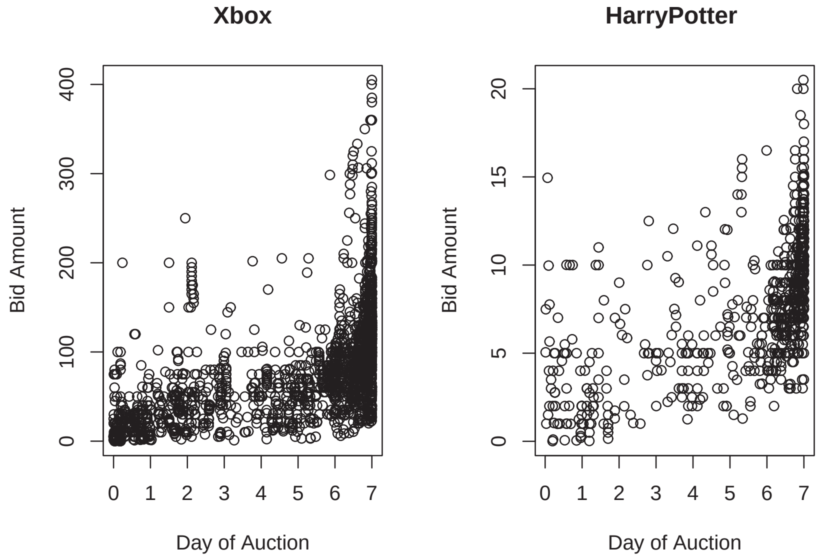


Figure 3.3: Data for the 190 7-day auctions: The graph on the left shows the amount of the bid vs the time of the bid, aggregated across all Microsoft Xbox gaming systems auctions. The graph on the right shows the amount of the bid vs the time of the bid, aggregated across all Harry Potter books auctions.

Every auction in our data resulted in a sale. In addition to the bid history, we also collected information on a wide variety of other auction characteristics such as the opening bid and the final price, the number of bids, and the seller and bidder ratings (see top of Table 3.1 for a summary of these continuous variables). We also recorded item condition (used vs. new), whether or not the seller sets a secret reserve price, and whether or not the auction exhibited early bidding or jump bidding (see bottom of Table 3.1 for these categorical variables).

From Table 3.1, we can see that auctions vary considerably. For instance, while some auctions only received 2 bids, others received as many as 75 bids. We also see considerable variation in final prices which is not surprising since we are considering items of different value. Unsurprisingly, the high-valued items (Xbox) have, on average, a higher opening bid and a higher final auction price. However, it is noteworthy that the high-valued items see, on average, more competition (i.e., a larger number of bids), but feature auction participants with lower average bidder and seller ratings. The variation in the opening bid is more intriguing since it has been found to have a direct and indirect effect on price [13]. The opening bid directly influences final prices in that higher valued items often see higher opening bids. However, its indirect influence has the opposite direction: lower opening bids attract more bidders and the increased competition often results in a higher price. We can also see that the seller rating varies between 0 and almost 10,000. On eBay, a seller's rating is often associated with trust, and higher rated sellers often extract price premiums associated with this higher level of trust. Similarly, bidder ratings, which vary between 0 and almost 800 in our data, are often taken as a measure of "experience," and one typically hypothesizes that more experienced bidders make smarter bidding decisions. More interestingly, most of the high-valued items are used (over 90%), compared to only 46% of the used Harry Potter books.

Table 3.1 allows for additional insight into the data. We can see that for only 2% of all auctions the seller set a secret reserve price. Secret reserve prices act as an insurance for the seller in that s/he is not obligated to sell if the price stays below that level. The magnitude of a secret reserve prices is not known to the bidders and it has been found that imposing a reserve price on the one hand leads to increased revenue (in the event that the object is sold) but on the other hand it also lowers the likelihood of selling the object (see [88, 72]).

More directly related to bidding dynamics are the phenomena of early bidding and jump bidding. Table 3.1 also shows the distribution for the two variables *"early bidding"* and *"jump bidding"*. Note that these two variables are not directly observed but are derived from the bid history. We comment on how we derived these two variables next.

*Early bidding:* The timing of a bid plays an important role in bidders' strategic decision making. For example, [90] finds evidence that many bidders place their bids very late in the auction, resulting in what is often called "bid-sniping". According to [96], an auction often consists of 3 relatively distinct parts: an early part with *some* bidding activity, a middle part with *very little* bidding and a final part with *intense* bidding. In particular, they find that the early bidding part of the auction typically extends until about day 1.5 of a 7-day auction. In [11], bidders' strategies are characterized by, among other things, the timing of their bids. Bidders of the "early evaluators" type place their first bid on average on day 1.4 of a 7-day auction. Following this empirical evidence, we define an auction as characterized by *early bidding* if the first bid is placed within the first 1.5 days. Table 3.1 shows the distribution of auctions with early bidding. We see that among the high-valued auctions (Xbox), over 50% exhibit early bidding while this number is much lower (28%) for the low-valued items (Harry Potter). It may well be that bidders for high-valued items are more inclined to bid early in order to establish a time priority, since, in the case of two bidders with identical bids, the bidder with the earlier bid wins the auction.

*Jump bidding:* We also include information about jump bidding. To that end, one has to define what exactly determines a "jump bids," that is, what magnitude of difference between two consecutive bids constitutes an unusually high increase in the bidding process. The unusual nature of a jump bid is that it increases the auction by much more than what would be required by the prescribed bid increment. There are different theories as to why a bidder may employ jump bidding. One possible reason is to emphasize their determination for winning this auction and, as a consequence, to deter competing bidders. There exists little prior investigation on that topic. For instance, [24] studies jump bidding as a strategy in ascending auctions and define jump bids as bid increments that are larger than the minimum increment required by the auctioneer (see also [42, 21]). Bid increments larger than the minimum increment are relatively common on eBay (see Figure 3.4). We therefore focus here on increments that result in a very unusual "jump". In order to define "unusual", we take the following approach. For all auctions in our data set, we first examine all differences in bid magnitudes between pairs of consecutive bids. The difference in consecutive bids leads to a step function of bid increments. Figure 3.4 shows this step function for all Xbox and Harry Potter auctions. We can see that most auctions are characterized by only very small bid increments (i.e., only very small steps). But we can also see that the relevance of a jump depends on the scale (i.e., value of the item, which is the final price of the item.) and should be considered *relative* to this value.

The distribution of the *relative* jumps, relative to the average final price, is displayed in Figure 3.5. We see that the distribution for both high- and low-valued items is very skewed. In addition, the great majority of relative jumps, regardless of item-value, are smaller than 30% (see right graph in Figure 3.5). We therefore

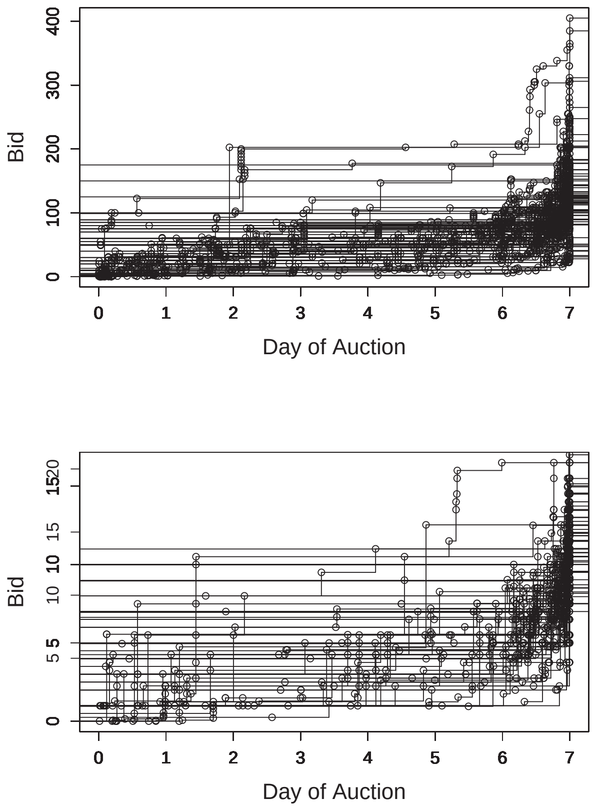


Figure 3.4: Step function of bid-increments for Microsoft Xbox systems (upper panel) and Harry Potter books (lower panel).

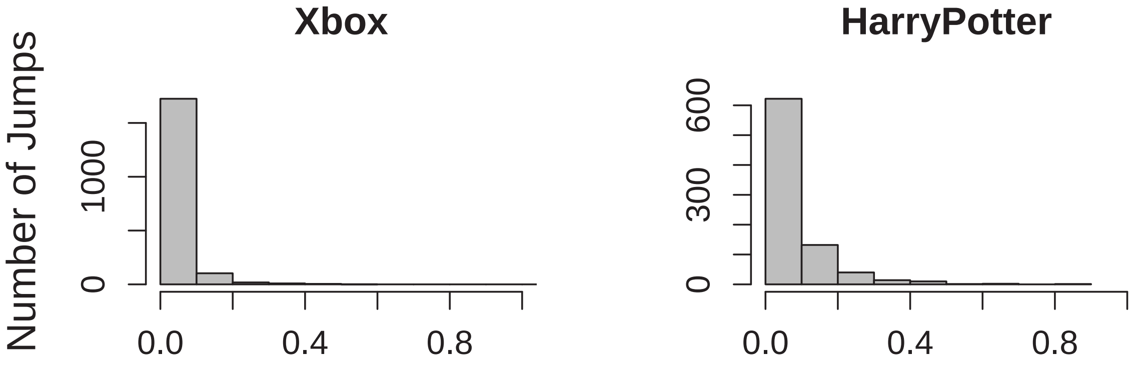


Figure 3.5: Distribution of relative jumps for Xbox alone (left) and Harry Potter alone (right).

define a *jump bid* as a bid that is at least 30% higher than the previous bid. We define a corresponding indicator variable for auctions that have at least one jump bid (i.e. the variable "Jump Bidding" in Table 3.1 equals one if and only if the auction has at least one jump bid). Table 3.1 shows that over 25% of the low-valued auctions (Harry Potter) see jump bidding, while this number is only about 9% for the high-valued auctions.

The data described in the previous section correspond to the cross-sectional attribute data that accompany every auction. However, our main focus in this dissertation is on the functional data produced by online auctions. That is, our focus is on the time series of bids placed over time. We will think of this time series of bids as the price evolution or the price path. In order to arrive at a smooth representation of the price path, we employ smoothing techniques. This is described next.

## 3.4 Creation of Smooth Functional Objects

Functional data consist of a collection of continuous functional objects such as the price which increases in an online auction. Despite their continuous nature, limitations in human perception and measurement capabilities allow us to observe these curves only at discrete time points. Moreover, the presence of error results in discrete observations that are noisy realizations of the underlying continuous curve. In the case of online auctions, we observe only bids at discrete times which can be thought of as realizations from an underlying continuous price curve. Thus, the first step in every functional data analysis is to recover, from the observed data, the underlying continuous functional object. This is typically done using smoothing techniques. Before using smoothing methods, our data has to be pre-processed suitably, which we will describe next.

### 3.4.1 Data Pre-Processing

In this subsection, we give a brief review of some techniques for converting raw functional data into true functional form. The recovery stage is often initiated by some data pre-processing steps (e.g., [83]). The bid data that are displayed on eBay's website are the so-called "proxy bids." Proxy bids are the highest bids that bidders are willing to pay for an item. eBay's automated bidding system records a proxy bid but displays, during the live auction, only an increment above the second highest proxy bid. According to eBay, there are several advantages to the proxy bidding system: On the one hand, bidders do not constantly have to monitor the auction site; another reason is that the winner only pays (an increment above) the second highest bid (see *http://pages.ebay.com/help/buy/proxy-bidding.html*). From a conceptual point of view, proxy bids are *not* what the bidders see and react to during the live auction. For that reason, we first reconstruct the live bids from the proxy bids by using eBay's bid increment table (*http://pages.ebay.com/help/buy/bid-increments.html*).

Figure 3.6 shows the difference between live and proxy bids for a sample auction. The "+"'s denotes the proxy bids, the solid circles denote the corresponding live bids, and the dashed line shows the live bid function which is a step function with steps at the time of a new bid. R code for transforming proxy bids into a live bid step function is available online at *www.smith.umd.edu/ceme/statistics/code.html.*

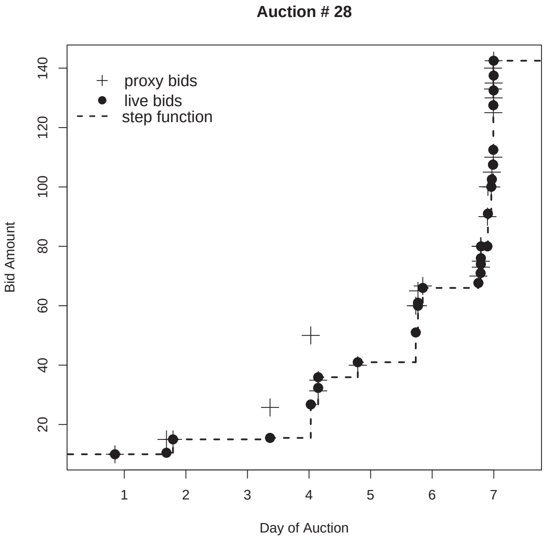


Figure 3.6: "Proxy bids", "live bids" and the corresponding step function for a sample auction.

We denote the time that the $i$th bid was placed, $i = 1, \ldots, N_j$, in auction $j$ ($j = 1, \ldots, N$) by $t_{ij}$. Note that due to the irregular spacing of the bids, the $t_{ij}$'s vary for each auction. In our data, $N = 190$ and $0 < t_{ij} < 7$. Let $y_i^{(j)}$ denote the bid placed at time $t_{ij}$. To better capture the bidding activity, especially at the end of the auction, we transform bids into log-scores. In order to account for the irregular spacing, we linearly interpolate the raw data and sample it at a common set of time points $t_i$, $0 \leqslant t_i \leqslant 7$, $i = 1, \ldots, G$. Then we can represent each auction by a vector of equal length

$$y^{(j)} = (y_1^{(j)}, \ldots, y_G^{(j)}), \tag{3.1}$$

where $y_i^{(j)} = y^{(j)}(t_i)$ denotes the value of the interpolated bid sampled at time $t_i$.

Once the preprocessing step is done, we convert the raw functional data into true functional form using various smoothing techniques, which we describe in the next section.

### 3.4.2 Representing Functional Data as Smooth Functions

The basic philosophy of functional data analysis is that we should think of observed data functions as single entities, rather than merely a sequence of individual observations. The term *functional* refers to the intrinsic structure of the data rather than to their explicit form. But in practice, functional data are usually observed and recorded discretely as we showed in previous sections. Since some observational noise is part of most data, the functional representation of raw data usually involves some smoothing.

There are three basic approaches for approximating discrete data by a smooth function. The first, which is one of the most common smoothing procedures, is using a set of basis functions; a second approach is via local expansion smoothing techniques; and a third approach is via the roughness penalty approach. We explore different smoothing approaches in this dissertation for different applications. We describe these approaches next.

The basis approach involves representing the function by a linear combination of $K$ known basis functions $\phi_k$ (see [86, 35] for details). As pointed out in [86], basis expansions work well if the basis functions have the same essential characteristics as the process generating the data. The disadvantage is that basis expansions have clumsy discontinuous control over the degree of smoothing and can be expensive to compute if the basis exhibits neither orthogonality nor local support.

Local expansion smoothing techniques, i.e., standard kernel smoothing and local polynomial fitting techniques, are based on appealing, efficient and easily understood algorithms that are fairly simple modifications of classic statistical techniques (see [86, 35] for details). They offer continuous control of the smoothness of the approximation, but they are seldom optimal solutions to an explicit statistical problem, such as minimizing a measure of total squared error, and their rather heuristic character makes extending them to other smoothing situations difficult.

In this work, we use the third approach since the roughness penalty or regularization approach retains the advantages of the basis function and local expansion smoothing techniques, but circumvents some of their limitations. Like the basis expansion approach, roughness penalty methods are based on an explicit statement of what a smooth representation of the data is trying to do, but the need to have a smooth representation is expressed explicitly at the level of the criterion being optimized. More importantly, they can be applied to a much wider range of smoothing problems than simply estimating a curve $y$ from observations $y(t_i)$ at certain points $t_i$. The book [35] discusses a variety of statistical problems that can be approached using roughness penalties, including those where the data's dependence on the underlying curve is akin to the dependence on parameters in generalized linear models. The scope of roughness penalty methods is extended still further in [86] by discussing various functional data analysis contexts where roughness penalties are an elegant way to introduce smoothing into the analysis.

In the following, we give a brief introduction to the roughness penalty approaches that we use in our work.

## Penalized Spline Smoothing

In this part, we briefly introduce the penalized spline smoothing method which we use in Chapter 4 (see [98, 86] for more details).

Consider a polynomial spline of degree $p$

$$f(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \ldots + \sum_{l=1}^{L} \beta_{pl}(t - \tau_l)_+^p, \tag{3.2}$$

where $\tau_1, \ldots, \tau_L$ is a set of knots and $u_+$ denotes the positive part of a function $u$. The choices of $L$ and $p$ strongly influence the departure of $f$ from a straight line. The degree of departure can be measured by the roughness penalty $PEN_m = \int D^m f(s)^2 ds$. The penalized smoothing spline minimizes the penalized residual sum of squares. That is the $j$th smoothing spline $f^{(j)}$ satisfies

$$PENSS_{\lambda,m}^{(j)} = \sum_{i=1}^{n} (y_i^{(j)} - f^{(j)}(t_i))^2 + \lambda PEN_m^{(j)}, \tag{3.3}$$

where the smoothing parameter $\lambda$ controls the trade-off between data fit and smoothness of $f^{(j)}$.

We base the selection of the knots on the bid arrival distribution. Consider again Figure 3.3, which shows that over 60% of the bids arrive during the last day of the auction. Moreover, the phenomenon of bid sniping [90] suggests that auctions should be sampled more frequently at their later stages. Also, in [96], it was found that the bid-intensity changes significantly during the last 6 hours. Motivated by this empirical evidence, we place a total of 14 knots and distribute the first 50% equally over the first 6 auction days. Then, we increase the intensity by placing the next 3 knots at every 6 hours, between day 6 and day 6.75. We again increase the intensity over the final auction moments by placing the remaining 4 knots every 3 hours, up to the end of the auction. This results in a total set of smoothing spline knots given by $\Upsilon = \{0, 1, 2, 3, 4, 5, 6, 6.25, 6.5, 6.75, 6.8125, 6.875, 6.9375, 7\}$.

We use smoothing splines of order $m = 5$ since this choice allows for a reliable estimation of at least the first three smooth derivatives of $f$ [86]. Note that our results are robust to changes in the knot-allocation and with respect to the choice of $\lambda$ (see Appendix B for a sensitivity study of analyses in Chapter 4).

Figure 3.7 shows the recovered functional object for a typical auction. The plot in the upper left panel shows the curve pertaining to the price evolution $f(t)$ on the log-scale (solid line), together with the actual bids (crosses), and the remaining plots show the first, second and third derivatives of $f(t)$, respectively. The price evolution shows that price, as expected from an auction, increases monotonically towards the end. However, the rate of increase does not remain constant. While the price evolution resembles almost a straight line, the finer differences in the change of price increases can be seen in the price velocity $f'(t)$ (the first derivative of $f(t)$) or in the price acceleration $f''(t)$ (its second derivative). For instance, while the price velocity increases at the beginning of the auction, it stalls after day three and remains low until the end of day five, only to rise again and to sharply increase towards the auction-end. Acceleration precedes velocity and we can see that a price deceleration over the first day is followed by a decline in price velocity after day one. In a similar fashion, the third derivative ($f'''(t)$) measures the change in the second derivative. The third derivative is also referred to as the "jerk," and we can see that the jerk increases steadily over the entire auction duration, indicating that price acceleration is constantly experiencing new forces that influence the dynamics of the auction. Similar changes in auction dynamics have also been noted in [50].

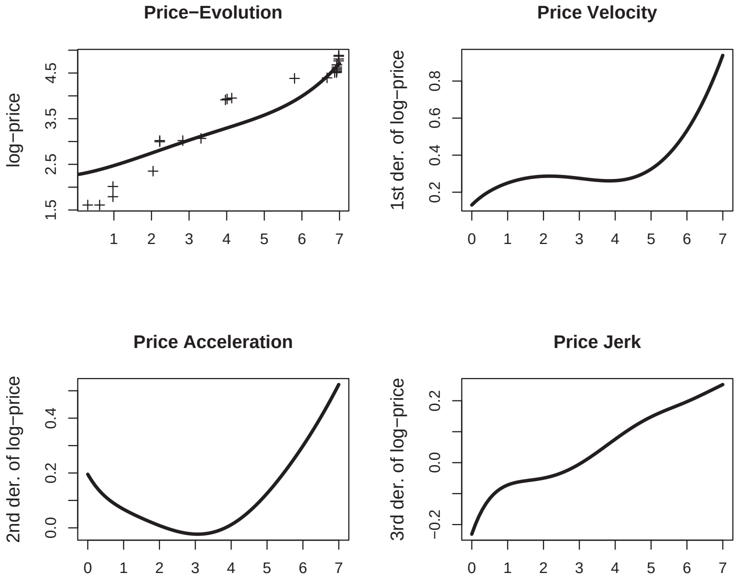


Figure 3.7: Price dynamics for Xbox auction number 10.

## The Regularized Basis Approach

In Chapters 5 and 6, we use a regularized basis approach, which is a more general approach of which spline smoothing is a special case. In the standard basis expansion method, the function $y$ is forced to lie in a relatively low dimensional space, defined in terms of a suitable basis. In local expansion smoothing, it is not assumed that the whole function is in the span of a particular basis, but rather a local basis expansion is considered at any given point. In the regularized basis approach, the function is allowed to have a higher dimensional basis expansion, but a roughness penalty is used in fitting the function to the observed data (see [86] for details).

In the regularized basis approach, a set of basis functions, $\phi_k, k = 1, \ldots, K$ is used to represent the function.

In order to arrive at a smooth representation, we approximate $y_i$ by a linear combination of basis functions. Write

$$y_i(t) = f_i(t) + \epsilon_i(t) \tag{3.4}$$

where the error term $\epsilon_i(t)$ is assumed to be the only cause of roughness for an otherwise smooth object. Using an appropriate basis functions expansion, we can represent $f_i(t)$ as

$$f_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t) \tag{3.5}$$

for a set of known basis functions $\phi = (\phi_1(t), \ldots, \phi_K(t))$ and a coefficient vector $c_i = (c_{i1}, \ldots, c_{iK})^T$. Then the $K \times N$ estimated coefficient matrix $\hat{\mathbf{c}} = (\hat{c}_1, \ldots, \hat{c}_N)$

minimizes the penalized sum of squares

$$PENSSE_\lambda(c) = \sum_{i=1}^{N}\sum_{j=1}^{n}(y_i(t_j) - f_i(t_j))^2 + \lambda \int (L_f(t))^2 dt. \tag{3.6}$$

In [35], it is shown that for $L_f(t) = f^{''}(t)$, $\hat{c}$ is given by

$$\hat{c} = (B^T B + \lambda H)^{-1} B^T Y(t), \tag{3.7}$$

where $B$ is the $n \times K$ basis matrix, $Y(t)$ is the $n \times N$ matrix of responses, and $\lambda$ is a smoothing parameter that controls the trade off between data fit and the smoothness. Note that the elements of H are given by $H_{kl} = \int c_k^{''}(t)c_l^{''}(t)dt$ (see also [86]). In Chapter 4, we use p-splines and in Chapters 5 and 6 we use B-splines. In Chpater 6, we use B-splines of order 6 to allow for a reliable estimation of at least the first three smooth derivatives of $f$ [86]. B-splines of order 6 are equivalent to P-splines of order 5. The R package *"fda"* contributed by Ramsay and Wickham, uses the regularized basis approach to convert functional data as functional data objects and to perform different types of analyses based on those objects. In particular, we employ principal differential analysis in Chapters 5 and 6. In order to represent our data using the specific form of functional representation in the *"fda"* package, we choose to use B-splines in those two chapters. The selection of the knots and smoothing parameter are driven by visual inspection of the resulting functional objects. The knots for the B-splines can be chosen to satisfy a criterion on the fit of the approximation, or knots can be placed on a fixed grid based on information about the behavior of the functional curve. For example, our selection of the knots is based on the bid arrival distribution. As explained earlier, we use the knots $\{0, 1, 2, 3, 4, 5, 6, 6.25, 6.5, 6.75, 6.8125, 6.875, 6.9375, 7\}$.A sensitivity analysis (see Appendix C) shows that our results for Chapter 5 are stable across different parameter choices.

The left panel in Figure 3.8 shows the functional objects for the 190 auctions in our data. In the following, we also refer to these objects simply as the "price curves". One advantage of having smooth functional objects is that one can readily obtain estimates for their dynamics via their derivatives. First and second derivatives of the price curve correspond to the price-velocity and -acceleration, respectively. The middle panel in Figure 3.8 shows that most price velocities are close to zero, especially during mid-auction. A near-zero price-velocity implies a price process that is in linear motion. Conversely, while velocities are low during the middle of the auction, they can be very high at the auction-start and especially at the end. Yet, Figure 3.8 also shows that although there are overall trends in the data, on an individual level, variation is quite large. For instance, while for some auctions price-acceleration is positive (and increases) towards the end, it is negative for others. A negative acceleration (=deceleration) indicates auctions for which the price movement slows down significantly. In the following we use *phase-plane plots* to investigate differences among auction dynamics more carefully. The insight drawn from this investigation will also be our starting point for our subsequent modeling work.

In the following chapters, we use the functional objects derived in this section

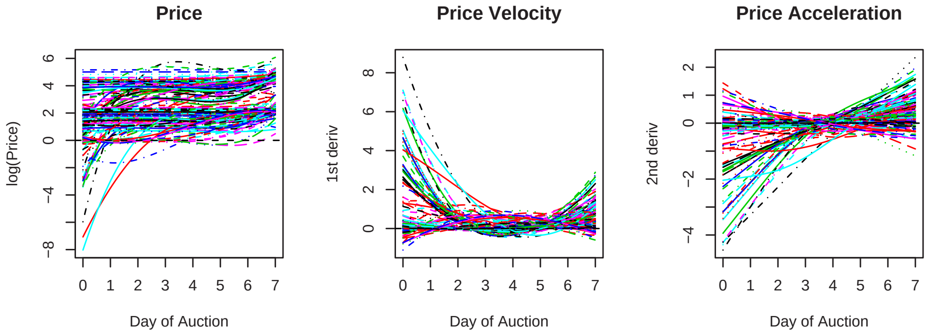


Figure 3.8: Price curves for the 190 7-day auctions on Xbox play stations and Harry Potter books, together with their estimated first two derivatives.

in a variety of ways. In Chapter 4, we develop a method to forecast a partially observed functional object. In Chapter 5, we propose several new tests to study differences between dynamic models based on functional data. And in Chapter 6, we incorporate dynamics into tree models for functional data.

| Variable | Item | Count | Mean | Median | Min | Max | StDev. |
|---|---|---|---|---|---|---|---|
| Opening Bid | Xbox | 93 | 36.22 | 24.99 | 0.01 | 175.00 | 37.96 |
| | Harry Potter | 97 | 4.13 | 4.00 | 0.01 | 10.99 | 3.26 |
| Final Price | Xbox | 93 | 134.58 | 125.00 | 28.00 | 405.00 | 66.03 |
| | Harry Potter | 97 | 11.56 | 11.50 | 7.00 | 20.50 | 2.40 |
| Number of bids | Xbox | 93 | 20.01 | 19.00 | 2.00 | 75.00 | 12.76 |
| | Harry Potter | 97 | 8.47 | 8.00 | 2.00 | 24.00 | 4.30 |
| Seller Rating | Xbox | 93 | 232.04 | 49.00 | 0.00 | 4604.00 | 613.07 |
| | Harry Potter | 97 | 325.99 | 126.00 | 0.00 | 9519.00 | 995.78 |
| Bidder Rating | Xbox | 93 | 30.33 | 4.00 | -1.00 | 2736.00 | 135.06 |
| | Harry Potter | 97 | 83.21 | 14.00 | -1.00 | 2258.00 | 226.21 |

| Variable | Item | Case | Count | Proportion |
|---|---|---|---|---|
| Reserve Price | Xbox | Yes | 4 | 4.3% |
| | | No | 89 | 95.7% |
| | Harry Potter | Yes | 1 | 1.0% |
| | | No | 96 | 99.0% |
| Condition | Xbox | New | 8 | 8.6% |
| | | Used | 85 | 91.4% |
| | Harry Potter | New | 52 | 53.6% |
| | | Used | 45 | 46.4% |
| Early Bidding | Xbox | Yes | 53 | 57.0% |
| | | No | 40 | 43.0% |
| | Harry Potter | Yes | 28 | 28.9% |
| | | No | 69 | 71.1% |
| Jump Bidding | Xbox | Yes | 9 | 9.7% |
| | | No | 84 | 90.3% |
| | Harry Potter | Yes | 25 | 25.8% |
| | | No | 72 | 74.2% |

Table 3.1: Summary statistics for all categorical variables. “Case” is the category for the particular variable.

# Chapter 4

# Explaining and Forecasting Online Auction Prices and their Dynamics using Functional Data Analysis[1]

## 4.1 Introduction

In this Chapter we develop a dynamic forecasting model to predict price in online auctions. By dynamic we mean a model that operates during the live auction and forecasts price at a future time point of the ongoing auction, and, as a by-product, also at the end of the auction. This is in contrast to static forecasting models that predict only the final price, and that take into consideration only information available at the start of the auction. Such information may involve the length of the auction, its opening price, product characteristics or the seller's reputation, and may be modelled using standard least-squares regression analysis. However, a static approach cannot account for information that becomes available after the start of the auction, e.g. the amount of competition or current price level, and it cannot incorporate such information "on the fly." As we explain throughout this essay, we find functional data analysis a very suitable tool for developing dynamic price predictions.

Forecasting price in online auctions can have benefits to different auction parties. For instance, price forecasts can be used to dynamically score auctions for the

[1] Forthcoming in the Journal of Business and Economic Statistics

same (or similar) item by their predicted price. On any given day, there are several hundred, or even thousand, open auctions available, especially for very popular items such as Apple iPods or Microsoft Xboxes. Dynamic price scoring can lead to a ranking of auctions with the lowest expected price. Such a ranking could help bidders focus their time and energy on only a few select auctions, namely those which promise the lowest price. Auction forecasting can also be beneficial to the seller or the auction house. For instance, the auction house can use price forecasts to offer insurance to the seller. This is related to the idea of [33], which suggests offering the seller an insurance that guarantees a minimum selling price. In order to do so though, it is important to correctly forecast the price, at least on average. While Ghani and Simmons' method is static in nature, our dynamic forecasting approach could potentially allow more flexible features like an "Insure-It-Now" option, which would allow sellers to purchase an insurance either at the beginning of the auction, or during the live-auction (with a time-varying premium). Price forecasts can also be used by eBay-driven businesses that provide service to buyers or sellers. Recently the authors were contacted by a company that provides brokerage services for eBay sellers, about using the dynamic forecasting system to create a secondary market for eBay-based derivatives.

While there has been some work related to price forecasting in online auctions, our approach is novel particularly because of its dynamic nature (see also [40] for related work on the dynamics of seller reputation ). As pointed out earlier, [33], using data-mining methods, also predicts the end-price of online auctions, however that method is static and cannot account for newly arriving information in the live-auction. Structural models to recover the bid distribution [9], while able to more explicitly account for mechanism design, are also focused on the final price. The dynamic nature of our forecasting approach is founded within the framework of functional data analysis (FDA).

Our forecasting approach presents several methodological additions to this stream of literature. First, to the best of our knowledge, forecasting functional data is a topic that has not been sufficiently addressed in the FDA literature to date. In fact, the use of functional data analysis presents several practical and conceptual advantages for online auction data. Traditional methods for forecasting time series, such as exponential smoothing or moving averages, cannot be applied in the auction context, at least not directly, due to the special data structure. Traditional forecasting methods assume that data arrive in evenly-spaced time intervals such as every quarter or every month. In such a setting, one trains the model on data up to the current time period $t$, and then uses this model to predict at time $t+1$. Implied in this process is the important assumption that the distance between two adjacent time periods is equal, which is the case for quarterly or monthly data. Now consider the case of online auctions. Bids arrive in very unevenly-spaced time intervals, determined by the bidders and their bidding strategies, and the number of bids within a short period of time can sometimes be very sparse and other times be extremely dense. In this setting, the interval between bids can sometimes be more than a day, and at other times only seconds. And secondly, online auctions, even for the same product, can experience price paths with very heterogeneous *price dynamics*. By price dynamics we mean the speed at which price increases during the auction and the rate at which this speed changes. Traditional models do not account for instantaneous change and its effect on the price forecast. This calls for new methods that can measure and incorporate this important dynamic information.

Another appeal of the functional data framework is the observation that the price dynamics change quite significantly over the course of an auction [50]. By treating auction price as a functional object and recovering the underlying price curve, we obtain reliable estimates of the price dynamics via derivatives of the smooth functional object, and we can consequently incorporate these dynamics into the forecasting model. This results in a novel and potentially very powerful forecasting system. While one may also approximate dynamics differently, e.g. by using the first forward difference or the central difference, such an approach is likely to be much less accurate, especially for applications with very unevenly spaced data (as in the case of online auction), and even more so for approximating higher order derivatives.

The Chapter is organized as follows. In Section 4.2 we provide a systematic description of the empirical regularities in online bidding dynamics. Section 4.3 develops the forecasting model and we apply the method to our data in Section 4.4. Section 4.5 concludes with final remarks.

## 4.2 Functional Regression and Auction Dynamics

In order to understand the motivation for our forecasting model, it is useful to first take a closer look at eBay auction data. We have pointed out earlier that the data are characterized by rapidly changing price dynamics. We illustrate this phenomenon in this section by investigating the relationship between eBay's auction dynamics and other auction-related information. This will also lay the ground for the forecasting model which we describe in the next section.

We investigate the empirical regularities in eBay's auction dynamics using *functional regression analysis*. Functional regression analysis is similar to classical regression in that it relates a response variable to a set of predictors. However, in contrast to classical regression where the response and the predictors are vector-valued, functional regression operates on *functional objects* which can be a set of curves, shapes, or objects. In our application, we refer to the continuous curve that describes the price evolution between the start and end of the auction as the functional object. More details on functional regression can be found in [86].

Functional regression analysis involves two basic steps. In the first step, the functional object is recovered from the observed data. This has been described in Chapter 3. After recovering the functional object, we model the relationship between a response-object and a predictor-object in a way that is conceptually very similar to classical regression. We describe that step in Subsection 4.2.1.

### 4.2.1 The Mechanics of Functional Regression Models

In this section we briefly review the general mechanics of functional regression models for a functional response variable. For a more detailed description see Chapter 11 of [86].

Our starting point is an $N \times 1$ vector of functional objects $\underline{y}(t) = [y_1(t), \ldots, y_N(t)]$ where $N$ denotes the sample size, the total number of auctions in this case. We use the symbol $y_j(t)$ in a rather generic way to model the price evolution of an auction by setting $y_j(t) \equiv f_j(t)$. However, one of the advantages of functional data is that we also have estimates of the dynamics. For instance, to model an auction's price acceleration we set $y_j(t) \equiv f_j''(t)$, and so forth. Classical regression models the response as a function of one (or more) predictor variables and that is no different in functional regression. Let $\underline{x}_i = [x_{i1}, \ldots, x_{iN}]$ denote a vector of $p$ predictor variables, $i = 1, \ldots, p$. $x_{ij}$ can represent the value of the opening bid in the $j$th auction or, alternatively, its seller rating. Time-varying predictors can also be accommodated in this setting. For instance, $x_{ij}(t)$ can represent the number of bids in the $j$th auction *at time* $t$, which we refer to as *the current number of bids at* $t$. Operationally, one can include such a time-varying predictor into the regression model by discretizing it over a finite grid. Let $x_{ijt}$ denote $x_{ij}(t)$ evaluated at $t$, for a suitable grid $t = t_1, \ldots, t_G$. We collect all predictors (time-varying and time-constant) into the matrix $X$. Typically, this matrix has a first column of ones for the intercept. Also, we could write $X = X(t)$ to emphasize the possibility of time-varying predictors but we avoid it for ease of notation. We then obtain the functional regression model

$$\underline{y}(t) = X^T \underline{\beta}(t) + \underline{\varepsilon}(t) \tag{4.1}$$

where the regression coefficient $\underline{\beta}(t)$ is time-dependent, reflecting the potentially varying effect of a predictor at varying stages of the auction. In this setting, $\underline{\beta}$

is defined for the finite grid $t = t_1, \ldots, t_G$, while it is also defined for continuous time $t$. The residual function $\underline{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_N)^T$ is the unexplained variation with independent components and each component is specific to each functional response.

Estimating the model (4.1) can be done in different ways (see [86] for a description of different estimation approaches). We choose a pointwise approach, that is, we apply ordinary least squares to (4.1) for a fixed $t = t^*$, and repeat that process for all $t$ on a grid, $t = t_1, \ldots, t_G$. By smoothing the resulting sequence of parameter estimates $\hat{\beta}(t_1), \ldots, \hat{\beta}(t_G)$, we obtain the time-varying estimate $\hat{\beta}(t)$.

While functional regression is, at least in principle, very similar to classical least-squares regression, attention has to be paid to the interpretation of the estimate $\hat{\beta}(t)$. We reemphasize that since the response is a continuous curve, so is $\hat{\beta}(t)$. This makes reporting and interpreting the results different from classical regression and slightly more challenging. We show how this is done in the next subsection.

### 4.2.2 Empirical Application and Results

We fit the functional regression model (4.1) to our data and investigate two different models: The first model investigates the effect of different predictor variables on the *price evolution*; that is, we set $y_j(t) \equiv f_j(t)$. The results are shown in Figure 4.1. The second model investigates the effect of the same set of predictors on the *price velocity*, that is $y_j(t) \equiv f_j'(t)$. Those results are shown in Figure 4.2. For both models, we use the nine predictor variables described in Table 3.1. Figures 4.1 and 4.2 show the estimated parameter curves $\hat{\beta}(t)$ (solid lines) together with 95%

confidence bounds (dotted lines) indicating the significance of the individual effects. The confidence bounds are computed pointwise by adding plus/minus 2 standard errors at each point of the parameter curve ([86]).

Interpretation of the parameter curves has to be done with care. At any time point $t$, $\hat{\beta}(t)$ evaluated at $t$ indicates the sign and strength of the relationship between the response (i.e. price in Figure 4.1, and velocity in Figure 4.2) and the corresponding predictor variable. The time-varying curve underlines the time-varying nature of this relationship. The confidence bounds help in assessing the statistical significance of that relationship.

The insight from Figures 4.1 and 4.2 is summarized below:

**Mechanism Design** We see that the choices that a seller makes regarding the opening bid and inclusion of a secret reserve price affects price according to what auction theory predicts: higher opening bids and inclusion of a secret reserve price are associated with higher price, at any time during the auction (see Figure 4.1). What has not been shown in previous studies though is the fact that this relationship, for both predictors, holds throughout the auction, rather than only at the end. Even more interesting is the observation that high opening bids and usage of a reserve price influence the price dynamics negatively towards the auction end by depressing the price velocity (see the negative coefficients in Figure 4.2). In both cases, this is most likely because price has already been inflated by the high opening bid and/or the driving bidding-force of the unobserved reserve price. We describe each of these two

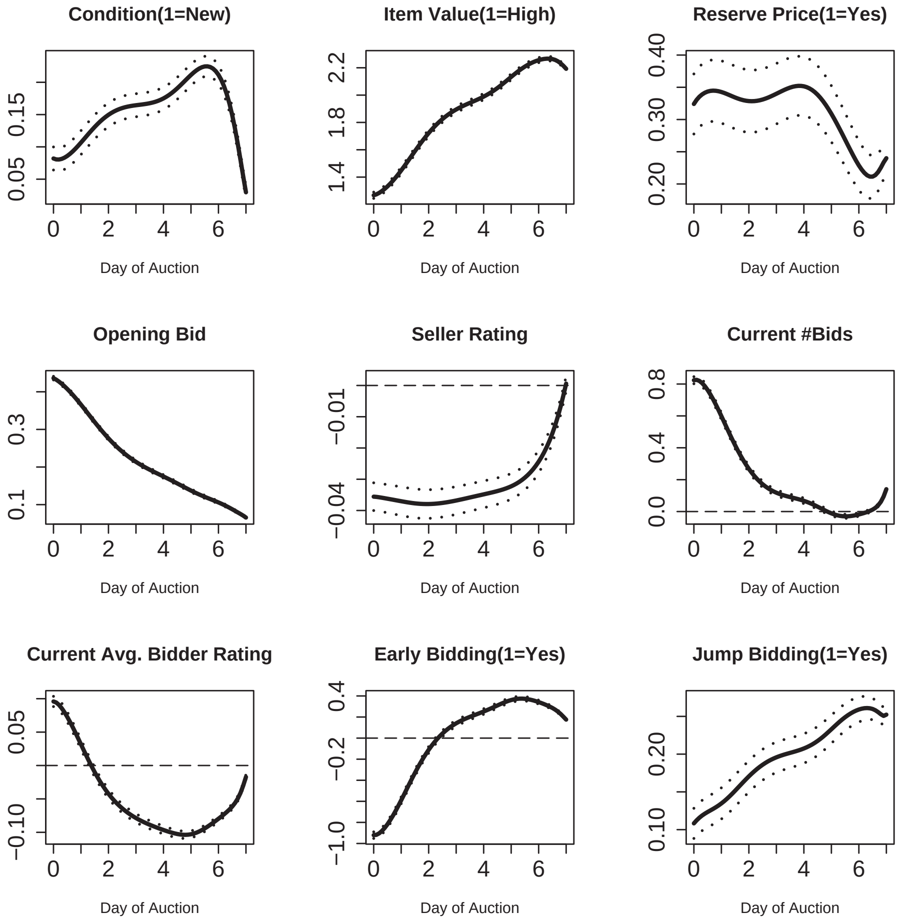


Figure 4.1: Estimated parameter curves based on functional regression on the price evolution. The x-axis denotes the time of the 7-day auction. The dotted lines correspond to 95% pointwise confidence bounds.

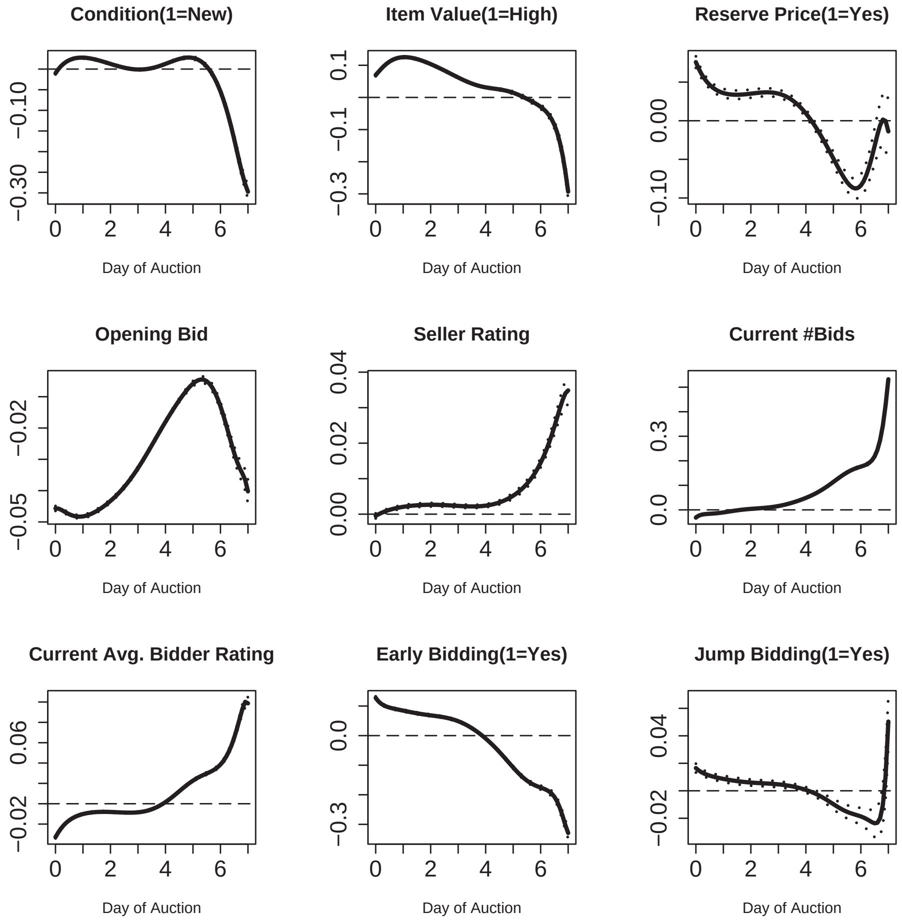


Figure 4.2: Estimated parameter curves based on functional regression on the price velocity. The x-axis denotes the time of the 7-day auction. The dotted lines correspond to 95% pointwise confidence bounds.

effects in more detail below.

Opening bid: The coefficient for opening bid in the regression on price evolution curves is shown in the middle-left panel in Figure 4.1. Throughout the entire auction the coefficient is positive, indicating a positive relationship between the opening bid and price at any time during the auction. However, the coefficient does decrease towards the auction-end, indicating that while the positive relationship between opening bid and price is strong at the auction start, it weakens as the auction progresses. One possible explanation is that at the auction-start, in the absence of other bids, auction participants derive a lot of information from the opening bid about their own valuation. As the auction progresses this source of information decreases in importance and participants increasingly look to other sources (e.g. number of competitors, number of bids and their magnitude, communication with the seller, etc.) for decision-making. In addition, the coefficient for opening bid in the regression on price velocity (Figure 4.2) is negative throughout the auction and strongest at the start and end of the auction. This indicates that higher opening bids depress the rate of price increase, especially at the start and end of the auction. Thus, although higher opening bids are generally associated with higher prices at any time in the auction (Figure 4.1), the auction dynamics are slowed down by high opening bids. In some sense, higher opening bids leave a smaller gap between the current price and a bidder's valuation, and therefore less incentive to bid.

Reserve price: Using a similar rationale as above, auctions with a secret reserve price tend to have a higher price throughout the auction, but a reserve price, similar to the high opening bid, appear to negatively influence price dynamics.

**Seller Characteristics** The anonymity of the internet makes it hard to establish trust. A seller's rating is typically the only sign that bidders look to in order to evaluate a seller's trustworthiness [22].

Seller rating: Empirical research has shown that higher seller ratings are associated with higher final prices (e.g. [88, 8]). Figures 4.1 and 4.2 though show that higher seller ratings are associated with lower prices during the entire auction, except for the auction end. Moreover, higher seller ratings are associated with faster price increases, but again only towards the auction-end.

**Item Characteristics** The items in our dataset are characterized by condition (used vs. new) and by value (high for Xbox, and low for Harry Potter).

Used/new Condition: Overall, new items achieve higher prices, which may not be surprising. However, the relationship between item condition and price velocity is negative. The price of new items appears to increase faster than used items *earlier*, but slows down *later* in the auction when used item prices increase at a faster pace. Perhaps the uncertainty associated with used items leads bidders to search for more information (such as contacting the seller or waiting for other bids to be placed) thereby leading to delays in the price spurts.

Item value: As one might expect, high-value items see higher prices than low-value items throughout the auction, and this gap increases as the action proceeds. More interestingly, the price dynamics are very similar in both low- and high-value items until about day 6, but then price increases much faster for low-value items. This is indicative of later bidding on low-value auctions, a phenomenon that we observed in the exploratory analysis. Bidders are more likely to bid early on high-value items, perhaps to establish their time priority.

**Bidding Characteristics** Our data contains four variables that capture effects of bidders and bidding, namely the current number of bids as a measure of level of competition, the current average bidder rating as a measure of bidder experience, and early and jump bidding as a measure of different bidding strategies. All variables share the feature that their impact changes sometime during the auction, thereby creating two phases in how they affect the price evolution and the price velocity. In some cases, different strategies (such as early vs. late bidding) lead to direct impacts on the price, but to more subtle effects on the price dynamics. For instance, the current number of bids affects the price evolution directly during the first part of the auction, while during the second part of the auction it affects price only through the price dynamics. The opposite phenomenon occurs with early bidding. Thus, functional regression reveals that (1) bidding appears to have two phases, and (2) price can be affected either directly or indirectly by the bidding process.

Current number of bids: This factor influences the price evolution during the first part of the auction, with more bids resulting in higher prices. However, this effect decreases towards the end of the auction where it only influences price through increasing price dynamics.

Early bidding: The effect of this factor switches its direction between the first and second part of the auction: At first, auctions with early bidding have higher dynamics but lower price evolution, but later this effect reverses. This means that early bidding manifests itself as early increased price dynamics, which later turn into higher price curves.

Jump bidding: Auctions with jump bidding tend to have generally higher price curves, and especially high price dynamics close to the auction end. The jump bidding obviously causes the price curve to jump and the price velocity to peak at the time of the jump bid. When averaging over the entire set of auctions, the effect of a jump bid has its highest impact on price at the auction end. This is not necessarily in contrast to [24], which examines the timing of jump bidding (rather than the time of its highest impact). It was found that jump bidding is more prevalent early in the auction, which is explained by the strategic value of jump bidding for bidders.

Current average bidder rating: It appears that higher rated bidders are more likely to bid when the price at the start of the auction is high, compared to lower rated bidders (as reflected by the positive coefficient during the first day). But then they are able to keep the price lower throughout the auction (the coefficient turns negative). Towards the end of the auction, though, participation of high-rated bidders leads to faster price increases, which reduces the final price gap due to bidder rating.

## 4.3 Dynamic Auction Forecasting via Functional Data Analysis

We now describe our dynamic forecasting model. We have shown in the previous section how unequally spaced data can be overcome by moving into the functional context, and also that online auctions are characterized by changing price dynamics. Our forecasting model consists of four basic components that capture price dynamics, price lags, and information related to sellers, bidders, and auction design. First we describe the general forecasting model which is based on the availability of price dynamics. Then, we describe how to obtain forecasts for the price dynamics themselves.

### 4.3.1 The General Forecasting Model

Our model combines all information that is relevant to price. We group this information into four major components: a) static predictor variables; b) time-varying predictor variables; c) price dynamics; and d) price lags.

Static predictor variables are related to information that does not change over the course of the auction. This includes the opening bid, the presence of a secret reserve price, seller rating, and item characteristics. Note that these variables are known at the start of the auction and remain unchanged over the auction-duration. Time-varying predictor variables are different in nature. In contrast to static predictors, time-varying predictors *do* change during the auction. Examples of time-varying predictors are the number of bids at time $t$, or the number of bidders and their average bidder-rating at time $t$. Price dynamics can be measured by the price velocity, the price acceleration, or both. And finally, price lags also carry important information about the price development. Price lags can reach back to price at times $t-1$, $t-2$, and so on. This corresponds to lags of order 1, 2, etc.

We obtain the following dynamic forecasting model based on the smoothed functional data. Let $y(t|t-1)$ denote the price at time $t$, given all information observed until $t-1$. For ease of notation, we write $y(t) \equiv y(t|t-1)$. Our forecasting model can then be formalized as

$$y(t) = \alpha + \sum_{i=1}^{Q} \beta_i x_i(t) + \sum_{j=1}^{J} \gamma_j D^{(j)} y(t) + \sum_{l=1}^{L} \eta_l y(t-l) \qquad (4.2)$$

where $x_1(t), \ldots, x_Q(t)$ is the set of static and time-varying predictors, $D^{(j)}y(t)$ denotes the $j$th derivative of price at time $t$, and $y(t-l)$ is the $l$th price lag. The resulting $h$-step ahead prediction, given information up to time T, is then

$$\tilde{y}(T+h|T) = \hat{\alpha} + \sum_{i=1}^{Q} \hat{\beta}_i x_i(T+h|T) + \sum_{j=1}^{J} \hat{\gamma}_j \tilde{D}^{(j)} y(T+h|T) + \sum_{l=1}^{L} \hat{\eta}_l \tilde{y}(T+h-1|T). \quad (4.3)$$

The model (4.3) has two practical challenges: (1) price dynamics appear as coincident indicators and must therefore be forecasted *before* forecasting $\tilde{y}(T+h|T)$; (2) the static predictor variables among the $x_i$'s do not change their value over the course of the auction and must therefore be adapted to represent time-varying information. We explain these two challenges in more detail below and present some solutions.

### 4.3.2 Forecasting Price Dynamics

The price dynamics $D^{(j)}y(t)$ enter (4.3) as coincident indicators. This means that the forecasting model for price at time $t$ uses the dynamics from the same time period! However, since we assume that the observed information extends only until $t-1$, we must obtain forecasts of the price dynamics before forecasting price. This process is described next.

We model $D^{(j)}y(t)$ as a polynomial in $t$ with autoregressive (AR) residuals. We also allow for covariates $x_i$. The rationale for these covariates is that dynamics are strongly influenced by certain auction-related variables such as the opening bid (see again Figure 4.2). This results in the following model for the price dynamics

$$D^{(j)}y(t) = \sum_{k=0}^{K} a_k t^k + \sum_{i=1}^{P} b_i x_i(t) + u(t), \quad t = 1, \ldots, T, \tag{4.4}$$

where $u(t)$ follows an autoregressive model of order $R$ :

$$u(t) = \sum_{i=1}^{R} \phi_i u(t-i) + \varepsilon(t), \quad \varepsilon(t) \sim iid\ N(0, \sigma^2). \tag{4.5}$$

To forecast $D^{(j)}y(t)$ based on (4.4), we first estimate the parameters $a_0, a_1, \ldots, a_K$, $b_1, \ldots, b_P$ and estimate the residuals. Then, using the estimated residuals $\hat{u}(t)$, we estimate $\phi_1, \ldots, \phi_R$. This results in a 2-step forecasting procedure: Given information until time $T$, we first forecast the next residual via

$$\tilde{u}(T+1|T) = \sum_{i=1}^{R} \tilde{\phi}_i u(T-i+1), \tag{4.6}$$

and then use this forecast to predict the corresponding price derivative

$$D^{(j)}\tilde{y}(T+1|T) = \sum_{k=0}^{K} \hat{a}_k (T+1)^k + \sum_{i=1}^{P} \hat{b_i} x_i(T+1|T) + \tilde{u}(T+1|T). \tag{4.7}$$

In a similar fashion, we can predict $D^{(j)}y(t)$ $h$ steps ahead:

$$D^{(j)}\tilde{y}(T+h|T) = \sum_{k=0}^{K} \hat{a}_k (T+h)^k + \sum_{i=1}^{P} \hat{b_i} x_i(T+h|T) + \tilde{u}(T+h|T). \tag{4.8}$$

### 4.3.3 Integrating Static Auction Information

The second structural challenge that we face is related to the incorporation of static predictors into the forecasting model. Take, for instance, the opening bid. The opening bid is static in the sense that its value is the same throughout the auction, that is $x(t) \equiv x, \forall t$. Ignoring all other variables, model (4.2) becomes

$$y(t) = \alpha + \beta x. \tag{4.9}$$

Because the right hand side of (4.9) does not depend on $t$ the least-squares estimates of $\alpha$ and $\beta$ are confounded!

The problem outlined above is relatively uncommon in traditional time series analysis since it is usually only meaningful to include a predictor variable into an econometric model if the predictor variable itself carries time-varying information. However, the situation is different in the context of forecasting online auctions and may merit the inclusion of certain static information. The opening bid, for instance, may in fact carry valuable information for predicting price in the ongoing auction. Economic theory suggests that sometimes bidders derive information from the opening bid about their own valuation, but the impact of this information decreases as the auction progresses. What this suggests is that the opening bid can influence bidders' valuations and therefore also influence price. What this also suggests is that the opening bid's impact on price does not remain constant but should be discounted gradually throughout the auction.

One way of discounting the impact of a static variable $x$ is via its influence on the price evolution. That is, if $x$ has a stronger influence on price at the beginning of the auction, then it should be discounted less during that period. On the other hand, if $x$ only barely influences price at the end of the auction, then its discounting should be larger at the auction end. One way of measuring the influence of a static variable on the price curve is via functional regression analysis, as described in Section 4.2.1. Let $\tilde{\beta}(t)$ denote the slope-coefficient from the functional regression model $y(t) = \alpha(t)+\beta(t)x+\varepsilon$, similar to (4.1), and thus $\tilde{\beta}(t)$ quantifies the influence of $x$ on $y(t)$ at any time $t$. We combine $x$ and $\tilde{\beta}(t)$ and compute the *influence-weighted*

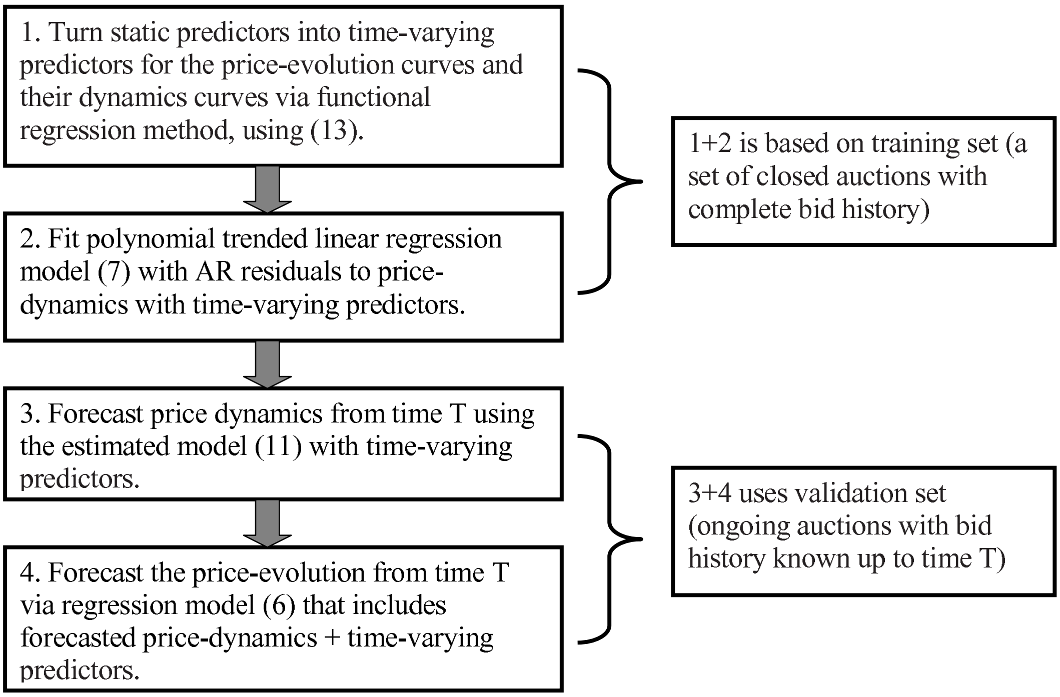

1. Turn static predictors into time-varying predictors for the price-evolution curves and their dynamics curves via functional regression method, using (13).
2. Fit polynomial trended linear regression model (7) with AR residuals to price-dynamics with time-varying predictors.
3. Forecast price dynamics from time T using the estimated model (11) with time-varying predictors.
4. Forecast the price-evolution from time T via regression model (6) that includes forecasted price-dynamics + time-varying predictors.
1+2 is based on training set (a set of closed auctions with complete bid history)
3+4 uses validation set (ongoing auctions with bid history known up to time T)

Figure 4.3: Flow-chart of dynamic forecasting model.

version of the static variable $x$ as

$$\tilde{x}(t) = x\tilde{\beta}(t). \tag{4.10}$$

$\tilde{x}(t)$ now carries time-varying information and can consequently be included as time-varying predictor variable.

As pointed out earlier, our dynamic forecasting model consists of two basic parts: one part forecasts the price dynamics, and the other part uses these forecasted dynamics as input into the price forecaster. A flowchart of our algorithm is shown in Figure 4.3.

## 4.4 Empirical Application and Forecasting Comparison

We apply the forecasting methodology to our dataset of 190 eBay auctions. Model fitting and prediction are implemented using modules of the R software package. We randomly partition our data into a training set (70% or 130 auctions) and a validation set (30% or 60 auctions). We use the training set to estimate the model, and test the method on the validation set. For testing, we first remove all price information from the last auction day, and then compare our results with the true price.

### 4.4.1 Model Estimation

Estimation of the model is done in two steps. We first estimate model (4.4) and then use the forecasted dynamics as inputs into model (4.3).

### Modeling Price Dynamics

Model (4.4) is fitted iteratively. This leads to a best-fitting model with a quadratic trend ($K = 2$) and three predictors ($P = 3$), where $x_1, x_2$ and $x_3$ are the influence-weighted variants of the opening bid, the item value and jump-bidding, respectively. The resulting residuals are AR(1), that is $R = 1$ in (4.5). Figure 4.4 shows the significance of $x_1, x_2, x_3$ over the last auction day in the form of *significance curves*. Since we use $x_1, x_2$ and $x_3$ to predict price dynamics for all time points between day 6 and 7, the significance of individual predictors may be different at different time points. Indeed, Figure 4.4 shows that while jump bidding (line #3 in the graph) is insignificant during the beginning of day 6 (with a huge spike around day 6.3), it turns significant towards the auction-end. The opposite is true for the item value, which becomes insignificant at the auction end. On the other hand, the opening bid remains significant throughout the last day. This change in significance suggests that the "burden of prediction" does not remain equally distributed over all three predictors. In fact, the burden is heavier on item value at the beginning of the auction and then shifts to jump bidding at the auction-end. Meanwhile, the opening bid carries the same prediction burden throughout all of the last day.

Figure 4.5 illustrates the forecasting performance on the holdout sample. We chose four representative auctions and compared the true price velocity over the last day (solid line) with its prediction based on model (4.4) (broken line). We see that the model captures the true price dynamics very well.

### Modeling Price

We estimate model (4.2) using the following 11 predictor variables (grouped by their type)

Influence-Weighted Static Predictors: Opening bid, Reserve price, Seller rating, Item condition, Item value, Early bidding, Jump bidding

Time-Varying Predictors: Current number of bids, Current average bidder rating

Price-Dynamics: Price velocity

Price-Lags: Price at time $t-1$

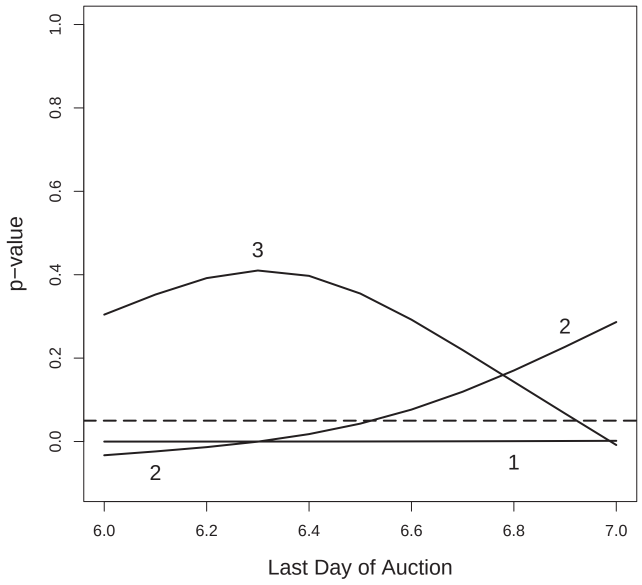


Figure 4.4: P-value curves for $x_1, x_2$ and $x_3$ over the last auction day. Consistent with the three predictors, we denote 1=opening bid; 2=item value; 3=jump bidding. The dotted horizontal line marks the 5% significance level.

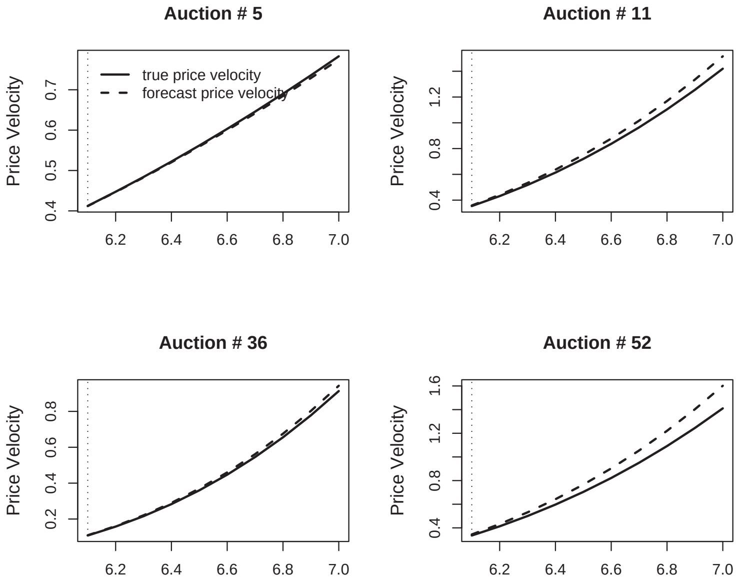


Figure 4.5: Forecasting performance of model (4.4) over the last auction day for four sample auctions.

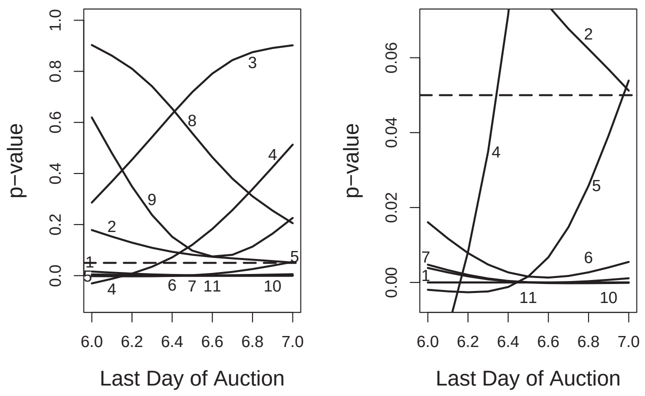


Figure 4.6: P-value curves for all 11 predictors over the last auction day. 1=opening bid; 2=reserve price; 3=seller rating; 4=item condition; 5=item value; 6=early bidding; 7=jump bidding; 8=current number of bids; 9=current avg.bidder rating; 10=price velocity; 11=price at time $t-1$. The dotted horizontal line marks the 5% significance level. The right panel shows the information from the left panel "zoomed-in" for p-values between 0 and 0.08.

Figure 4.6 shows the significance curves for all 11 predictors. Interestingly, reserve price, seller rating, current number of bids and current average bidder rating are insignificant at the auction start. While the significance of the latter two increases towards the auction end, seller rating turns even more insignificant. On the other hand, while reserve price becomes highly significant at the end, item condition, which is significant at the start, becomes insignificant at the end. All remaining predictors remain at (or below) the 5% significance mark throughout the entire auction.

### 4.4.2 Price Forecasting

After estimating the model using the training set, we apply it to the validation set to obtain forecasts for the price on the last day. Since we removed all price information from the last auction day, we can measure prediction accuracy by comparing the true price with our forecast.

Figure 4.7 illustrates the forecasting method for 4 sample auctions. Each of the 4 graphs in Figure 4.7 contains three separate pieces of information: a) the actual current auction price (a step function); b) the functional price curve; and c) the forecasted price curve. The actual current auction price is the price observed during the live auction. The functional price curve is the smoothed functional object based on the observed prices. And, the forecasted price curve is our forecast based on model (4.2).

Note that Figure 4.7 reveals two levels of "truth." The first is on the functional level, which compares true and forecasted *curves*. Our forecasting method operates on the functional objects and predicts the price curves. In that sense, the closer the forecasted curve is to the functional price curve, the better its functional prediction performance. Indeed, Figure 4.7 shows that the functional and forecasted curves are generally very close. However, the functional price curve is merely an approximation of the live auction price. Therefore, a second level of truth is revealed by comparing the forecasted price curve with the actual current auction price. On this level, the discrepancy is larger, which is not surprising: the quality of the forecasting output is only as good as its input. If the quality of the input is poor (i.e. functional objects that do not approximate the current auction price well), then not much can be expected of the forecasted output. This underlines the importance of generating high-quality functional objects. The most reliable way of checking the quality of the functional objects is via visualization. Several ways of inspecting functional data visually are proposed in [53]. Another way of guaranteeing the quality of the results is via sensitivity studies with respect to the allocation of knots and the choice of the smoothing parameter (see Appendix B).

### Forecast Accuracy

We measure forecast accuracy on the validation set using the mean-absolute-percentage-error (MAPE). We compute the MAPE in two different ways, similar to Figure 4.7, once between the forecasted curve and the true functional curve ($\text{MAPE}_1$), and then between the forecasted curve and the actual current auction price ($\text{MAPE}_2$). The result is shown in Figure 4.8.

Naturally, $\text{MAPE}_2$ is higher than $\text{MAPE}_1$, because it is harder to reach the second level of "truth" compared with the first level. $\text{MAPE}_1$ is, at least on average, less than +5% for the entire prediction period (i.e. over the last day), implying that our model has a very high forecasting accuracy. $\text{MAPE}_2$ is a bit larger in magnitude due to the inevitable variation in fitting smoothing splines to the observed data. The width of the confidence bounds underline the heterogeneity across all auctions in our data set.

### Forecast Accuracy by Auction Characteristics

Forecast accuracy can lead to new insight about the empirical regularities of bidding when breaking it up by different auction characteristics. We therefore compare forecast accuracy for different levels of the opening bid, secret reserve

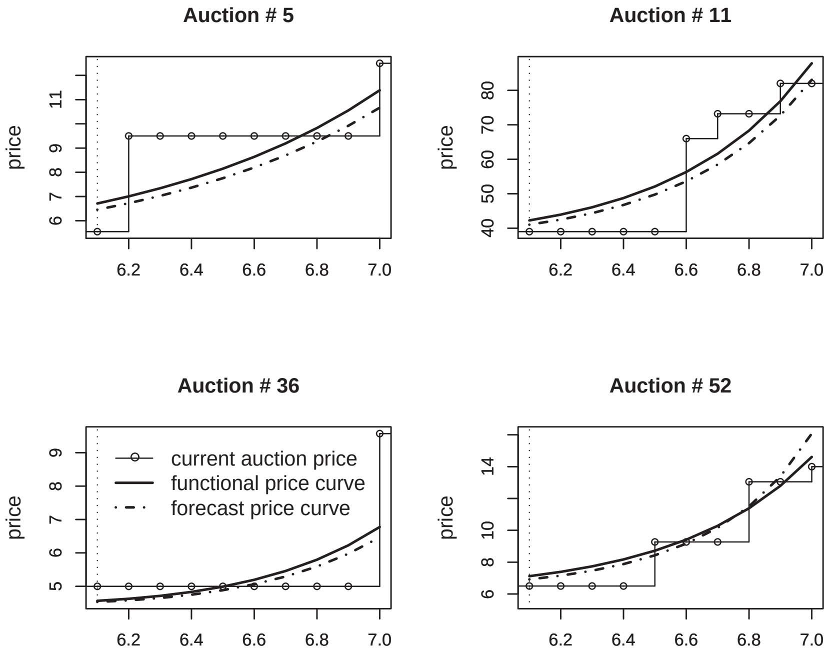


Figure 4.7: Dynamic forecasting results of last day price for 4 sample auctions. $x$-axes represent time of auctions and $y$-axes represent amounts of prices. Auctions #5, #36 and #52 are all auctions on Harry Potter Books, and auction #11 is an auction on Microsoft Xbox gaming system.

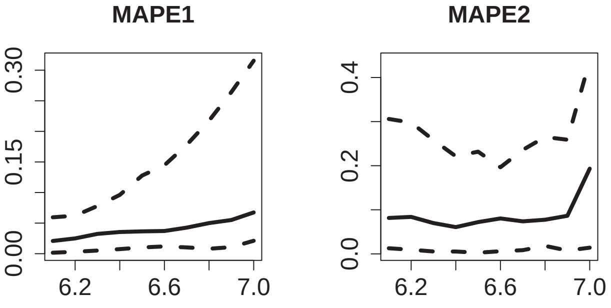


Figure 4.8: Mean Absolute Percentage Errors (MAPEs). MAPE$_1$ is the error between the forecasted price curve and the true functional price curve; MAPE$_2$ is the error between the forecasted price curve and the actual current auction price. The dotted lines correspond to the 5th and 95th percentiles.

price, item condition and value, seller reputation, bidder experience, competition, and early and jump bidding. Table 4.1 shows the results. We find that the error is generally relatively small, no larger than 20% of the true functional price curve, and no larger than 36% of the actual final auction price. But there are subtle differences across the different variables: the error is larger when forecasting new items as compared to used items. High value items, on the other hand, have a smaller error than low value items, which could be attributed to the fact that, when the stakes are higher, bidders spend more time researching the item and thus price dispersion is lower. Not surprisingly, auctions with a high opening bid have a smaller forecasting error since, when the opening bid is high and the item's value is relatively well-known as in our situation, then there is less uncertainty about the possible outcomes of the auction. Lower seller reputation results in more accurate forecasts. This may be due to the fact that higher seller ratings often elicit price-premiums [88], thereby increasing the price-variance. Bidder experience has a similar impact on forecasting accuracy. As for bidding competition (captured by the number of bids), higher competition results in larger variation in the forecast errors. It is also interesting to note that early bidding has barely any effect on the predictability of an auction; this again is different for jump bidding.

Table 4.1: Mean absolute percentage errors (MAPEs) broken up by different variables. MAPE$_1$ is the error between the forecasted final price and the functional final price; MAPE$_2$ is the error between the forecasted final price and the actual final price. The standard error of reserve price is "NA" since there is only one auction with a reserve price in the validation set.

| Variable | Case | MAPE$_1$ | | MAPE$_2$ | |
|---|---|---|---|---|---|
| | | Mean | Std.Err. | Mean | Std.Err. |
| Reserve Policy | Yes | 0.08 | NA | 0.16 | NA |
| | No | 0.12 | 0.02 | 0.23 | 0.02 |
| Condition | New | 0.17 | 0.05 | 0.31 | 0.05 |
| | Used | 0.09 | 0.01 | 0.19 | 0.02 |
| Item Value | High | 0.07 | 0.01 | 0.14 | 0.01 |
| | Low | 0.16 | 0.04 | 0.31 | 0.04 |
| Opening Bid | High | 0.06 | 0.01 | 0.14 | 0.02 |
| | Low | 0.20 | 0.04 | 0.36 | 0.04 |
| Seller Rating | High | 0.14 | 0.04 | 0.26 | 0.04 |
| | Low | 0.09 | 0.02 | 0.20 | 0.03 |
| Avg. Bidder Rating | High | 0.15 | 0.04 | 0.30 | 0.04 |
| | Low | 0.09 | 0.02 | 0.17 | 0.02 |
| Number of Bids | High | 0.13 | 0.03 | 0.24 | 0.03 |
| | Low | 0.10 | 0.02 | 0.22 | 0.03 |
| Early Bidding | Yes | 0.11 | 0.02 | 0.22 | 0.03 |
| | No | 0.12 | 0.03 | 0.24 | 0.04 |
| Jump | Yes | 0.09 | 0.01 | 0.27 | 0.03 |
| | No | 0.13 | 0.03 | 0.21 | 0.03 |

### Comparison with Exponential Smoothing designed for stationary processes

To benchmark the performance of our method, we compare it to Double Exponential Smoothing. Double Exponential Smoothing is a popular short term forecasting method which assigns exponentially decreasing weights as the observation become less recent and also takes into account a possible (changing) trend in the

Table 4.2: Comparison of forecasting accuracy between our dynamic forecasting model and exponential smoothing. The forecasting accuracy is measured by mean absolute percentage error (MAPE).

| Method | $\text{MAPE}_1$ | | $\text{MAPE}_2$ | |
|---|---|---|---|---|
| | Mean | Std.Err. | Mean | Std.Err. |
| Dynamic Forecasting | 0.12 | 0.02 | 0.23 | 0.02 |
| Exp. Smoothing | 0.42 | 0.03 | 0.49 | 0.03 |

data. Following are two equations associated with Double Exponential Smoothing:

$$\begin{aligned} S_t &= \alpha y_t + (1-\alpha)(S_{t-1} + b_{t-1}) \quad 0 \leqslant \alpha \leqslant 1 \\ b_t &= \gamma(S_t - S_{t-1}) + (1-\gamma)b_{t-1} \quad 0 \leqslant \gamma \leqslant 1. \end{aligned} \tag{4.11}$$

This method cannot be applied directly to the raw bid data due to its uneven spacing. Functional objects once again come to the rescue. We apply double exponential smoothing to a grid of evenly-spaced values from the functional curve. The dashed lines in Figure 4.9 show the performance of exponential smoothing for the same four auctions as in Figure 4.7. We see that the predictions based on exponential smoothing are very far from the true auction price and even far from the true functional price curve. Table 4.2 compares our forecasting system with exponential smoothing in terms of MAPE. We find that the forecast error of exponential smoothing is more than twice the error of our forecasting system.

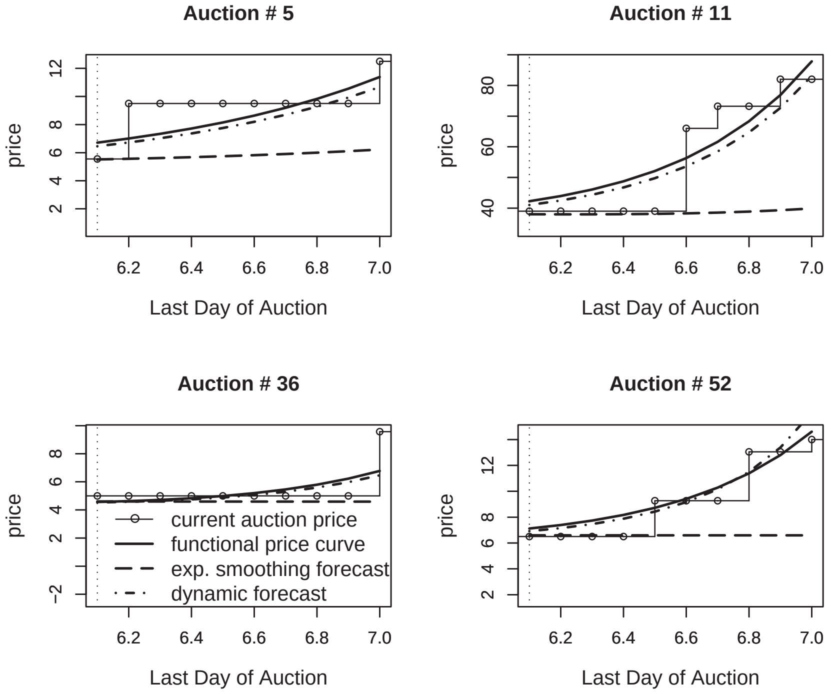


Figure 4.9: Comparison of forecasting results of last day price-evolution of individuals auctions between using Exponential Smoothing method and our dynamic forecasting method.

## 4.5 Conclusions and Future Directions

In this Chapter we propose a dynamic forecasting model for price in online auctions. We set up the forecasting problem in the context of functional data analysis by treating the price-evolution in an auction as a functional object. This leads to a novel use of FDA for forecasting which has not been considered in the literature to date. It is also new in that it allows dynamic forecasting of an ongoing auction. The functional setup allows us (1) to represent the extremely unevenly spaced series of bids in a compact form, (2) to estimate price dynamics via the derivatives of the smooth functional objects, and integrate this dynamic information into the forecaster, and (3) to incorporate both static and time-varying information about the auction into the forecasting system. Combining the dynamics with the static and time-varying information enables forecasting the price in ongoing live-auctions for different types of products. The functional approach allows us also to investigate regularities of the bidding dynamics as a function of relevant auction dimensions.

We apply our forecasting system to real data from eBay on a diverse set of auctions and find that the combination of static and time-varying information creates a powerful forecasting system. The model produces forecasts with low errors, and it outperforms standard forecasting methods like double exponential smoothing which severely under-predicts the price-evolution. This also shows that online auction forecasting is not an easy task. While traditional methods are hard to apply, they are also inaccurate since they do not take into account the dramatic change in auction dynamics. Our model, on the other hand, achieves high forecasting accuracy and accommodates the changing price-dynamics well.

This work can be extended in several ways. In this work, we focus on auctions of the same duration. The lessons learned from this work can be used to extend the model to auctions of different length. Combining auctions of different durations is challenging since it involves registration of misaligned curves (see e.g. [86] or [48]). However, in the auction context the misaligned curves are of different length which poses additional difficulties. Another extension is to incorporate a concurrency component. In online auctions, bidders have the option to inspect and follow multiple auctions at the same time. This places new challenges for modeling, especially in the functional framework. In a related series of papers (see [52, 41]) we propose some solutions via visualization of concurrent functional objects and modeling of concurrent final prices. Finally, further research is required to better understand the exact role of price dynamics and their impact on economic theory. One possible avenue is the exploration of functional differential equation models in the auction context (see [50]).

# Chapter 5

# Estimating Price Dynamics in Online Auctions Using Differential Equation Models[2]

## 5.1 Introduction

Online auctions have become a major player in providing electronic commerce services. EBay (*www.eBay.com*), the largest consumer-to-consumer auction site, enables a global community of buyers and sellers to interact and trade with one another. After less than ten years in existence, it already sees over $24 billion worth of transactions annually (*http://investor.ebay.com/ annuals.cfm*). Online auctions are different from their offline counterparts in their duration (typically several days), anonymity of participants (bidders and sellers do not know each other's identity), low barriers of entry (all it takes to place a bid on eBay is a valid credit card or a verified *Paypal* account), global reach and round-the-clock availability.

While online auctions have become a serious competitor to offline auctions, they also create new and previously unknown phenomena that depart from and cannot be explained by classical auction theory. These phenomena are related to the bidding process and lead to a drastic variability in the bidding dynamics. Classic auction theory, in a nutshell, says that the final price of an auction is determined by the a priori calculations of all bidders. There is more and more evidence though that

[2] Submitted to the Journal of the American Statistical Association

what happens *during* the auction also matters. For instance, [24] finds that jump bidding is an effective strategy in winning an auction. Related studies on the role of the starting bid and the bid increment (e.g., [12, 67]) find that both have a significant effect on the final price. Similarly, bid-timing and auction-entry matter [15], and [58] finds that the information revealed during the auction has a significant effect on its outcome. Moreover, there is an increasing notion that the auction process itself has social value for its participants (e.g. entertainment, competition) [99]. One prominent example of changing auction dynamics is the prevalence of "last-minute bidding" or "bid sniping" identified by several researchers [71, 90, 9, 10]. One can think of bid sniping as a "burst" of energy transpiring from one or more bidding parties in an attempt to "steal-away" the auctioned item in a last moment effort. The result is a drastic change in the auction dynamics, that is, in the speed at which the price moves and the rate at which new bids arrive.

These and other documented observed patterns suggest that online auctions experience and exhibit new behaviors that are not explained by classical auction theory. In particular, it also means that what happens during the auction process matters. While some effects of this process can be observed directly (such as jump bidding and bid sniping), we do not observe what motivates or causes these effects. For instance, we cannot observe *why* bidders engage in bid sniping. Moreover, there are additional effects (e.g., entertainment value, competitiveness) that we will never be able to directly observe. For example, bidders' competitiveness might result in "auction fever" with one outbidding the other over and over again. While we are not able to observe the factors that motivate the two bidders to act in this competitive way, we *are* able to observe the result: an increased speed of price movement and bid placement. In other words, while we are not able to observe many of the underlying factors that drive bidders' behavior, auction dynamics capture many of their effects. In that sense, we set out to develop a formal machinery to capture and model online auction dynamics.

There is recent evidence that dynamics exist, that they vary from auction to auction, and that they matter. Examples include [93], which finds that different levels of the opening bid are associated with different price dynamics. Moreover, [94] claims that price processes, even for auctions for an identical good, fall into one of three groups each of which exhibits different dynamics (see also [87] for similar results in auctions for modern Indian art). When considering the dynamics of bid timings, [96] shows that the bid arrival process changes during an auction with three stages of varying arrival-intensity. Additional evidence for a change in the bid arrival process has been pointed out in [90] which observes a change in last minute bidding activity. several types of bidder strategies that affect the number and timings of bids that bidders place in an auction are described in [11]. And finally, it is shown in [105] that the price dynamics, when incorporated into a forecasting model, can lead to real-time predictions of ongoing auctions and can improve upon the accuracy compared to classical forecasting methods.

In order to capture the dynamics of an online auction we employ, similar to some of the aforementioned authors, a functional data analysis framework. In that framework, we assume that the observed bids are realizations from an underlying continuous price process. Functional data analysis (FDA) has become popular in recent years, particularly with the monographs of [85, 86]. In FDA, the interest centers around a set of curves, shapes, images, or, more generally, a set of *functional objects*. In the auction context our object of interest is the price curve. To illustrate this, consider Figure 5.1 which shows price curves from 190 7-day online auctions for *Xbox* play stations and *Harry Potter* books (details of the data can be found in Chapter 3). We think of these 190 curves as a sample from a much larger population of online auction price processes. The goal is to model the population dynamics and to compare processes across different sub-populations. To do this, we directly model

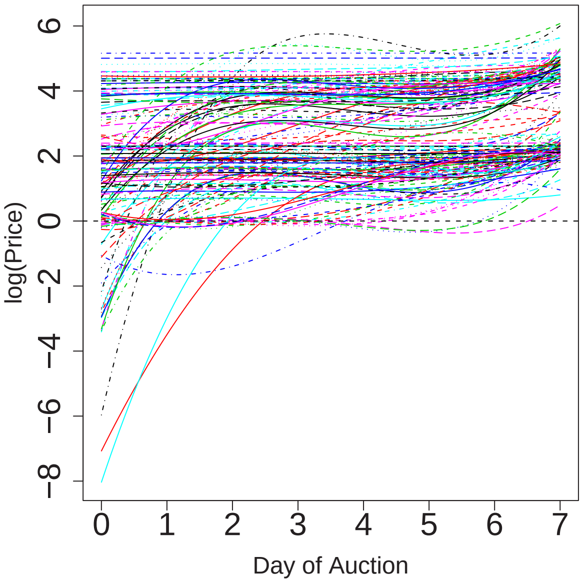


Figure 5.1: Price curves (on the log-scale) for 190 closed 7-day auctions of Xbox play stations and Harry Potter books.

price dynamics. By dynamics we mean the speed of price increases and the rate at which this speed changes.

In this Chapter we propose the use of differential equations for modeling online auction dynamics. Differential equations are a common tool in physics and engineering for modeling the dynamics of a closed system. They are commonly used for describing processes such as population growth, mixing problems, mechanics, and electrical circuits [19]. Applications of differential equations are also found in economics and finance. The Solow Model [100], for example, utilizes differential equations to model the long-run evolution of the economy (see also [64]). In finance, partial differential equations are used for pricing financial derivatives [54, 30, 4]. In contrast, the use of differential equations in classical statistics is rather little. In this Chapter we focus on the functional version of differential equations called *principal differential analysis* (PDA) and introduced in [82]. The basics of PDA are described in detail in the monograph [86]. We show that price dynamics in online auctions can be captured well by a single family of differential equation models. In doing so, we also propose a new test for multiple comparisons of differential equation models. Our test shows that auction sub-populations can be quite heterogeneous especially when considering different product-, auction- seller- or bidder-characteristics.

There are several practical implications of our work. As pointed out earlier, dynamics capture many of the otherwise unobservable effects of an online auction. In that sense, our work pioneers the formal modeling of dynamic bidding phenomena such as “auction fever” or “bidding frenzy”. Moreover, knowledge about auctions with different dynamics allows bidders to make more informed bidding decisions, e.g., by choosing to participate in auctions that have low anticipated end-dynamics. Knowledge of what drives dynamics can also help the seller in designing better auctions, and it can help the auctioneer to make adjustments that change the auction-experience (e.g., by controlling bid increment policies which could avoid e.g. the commonly experienced "bidding draughts" in the auction middle).

The Chapter is organized as follows. Section 5.2 provides an exploratory analysis using phase plane plots and motivates the employment of differential equation models we use in the following study. In Section 5.3, we describe the differential equation model, model-estimation and model-validation. We also introduce a new multiple-comparison test for principal differential analysis. Section 5.4 shows the results of fitting differential equations to online auction data and discusses insights and implications. We conclude with further remarks in Section 5.5.

## 5.2 Exploratory Analysis via Phase Plane Plots

The study of this Chapter is based on the same data set, 190 closed 7-day auctions for two different products, *Microsoft Xbox* gaming systems and *Harry Porter and the Half-Blood Prince* books, which was used in the previous work in Chapter 4 and which was described in detail in Chapter 3. At the heart of differential equations are models that relate the function and its derivatives to one another. As a preliminary step towards arriving at a suitable differential equation model, one often studies graphs (similar to scatter plots) that plot the derivative of one order versus the derivative of another order. These plots are often referred to as phase plane plots (PPPs). In the functional context, where one has repeat observations at each derivative level, one typically graphs the *averages* versus one another; for instance, the average acceleration versus the average velocity (see [86]).

Figure 5.2 shows a PPP for the average second derivative of price (or price-acceleration) versus the average first derivative (price-velocity). The numbers along the curve indicate the day of the auction (for 7-day auctions). We can see that the price velocity is high at the beginning of the auction (days 0-1): At the auction-start, it takes an instantaneous "burst of energy" to overcome the opening bid (which can be considerably high). After this initial burst, the dynamics slow down: the price acceleration is negative and since changes in acceleration precede changes in velocity, we observe a consequent slow-down in the price-speed. This slow-down continues until about day 4, after which the dynamics reverse: acceleration turns positive and causes the velocity to increase. In fact, it increases quite rapidly until the auction-end.

There are several interesting aspects that appear in Figure 5.2. First, the "C"-shape of the PPP is typical of a an online auction: a phase of decrease in dynamics followed by a transitional phase of change, and finally a phase of increase in dynamics. Our subsequent analyses will show that these three phases are typical of online auctions in general. However, we will also show that the magnitude/importance of each phase varies quite significantly, depending on different auction characteristics.

We now consider a series of *conditional* PPPs (see Figure 5.3), where the average derivatives are conditional on the auction characteristics described in Table 3.1. Note that we make a series of *pairwise* comparisons in the sense that we compare a conditional PPP *with* a certain feature to one *without* that feature. For instance, the two leftmost graphs in the second row contrast PPPs for auctions with and without a secret reserve price. We see that, as pointed out earlier, the general "C"-

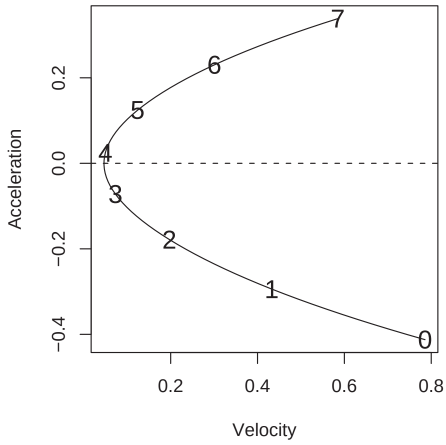


Figure 5.2: Phase Plane Plot for the average price curve of the data: the second derivative (acceleration) versus the first derivative (velocity).

shape of both PPPs is equivalent. However, the *magnitude* of the dynamics is very different: For auctions with no reserve price, the *range* of dynamics is significantly smaller, especially at the auction start. In particular, the price velocity is smaller at the auction-start and, as a consequence, does not decrease as fast as in auctions with a reserve price. Interestingly, towards mid-auction (at day 4) the dynamics of both types of auctions turn identical, but then again diverge towards the end of the auction, with reserve-price auctions exhibiting larger acceleration. The different size of the “C”-shapes also indicates that the magnitude of the *relationship* between velocity and acceleration differs: While in reserve-price auctions small changes in acceleration have a large effect on velocity, this effect is much more depressed in auctions without reserve prices.

In the following, we summarize the most important features we learn from Figure 5.3. For item value, high-valued items appear to have a larger range of

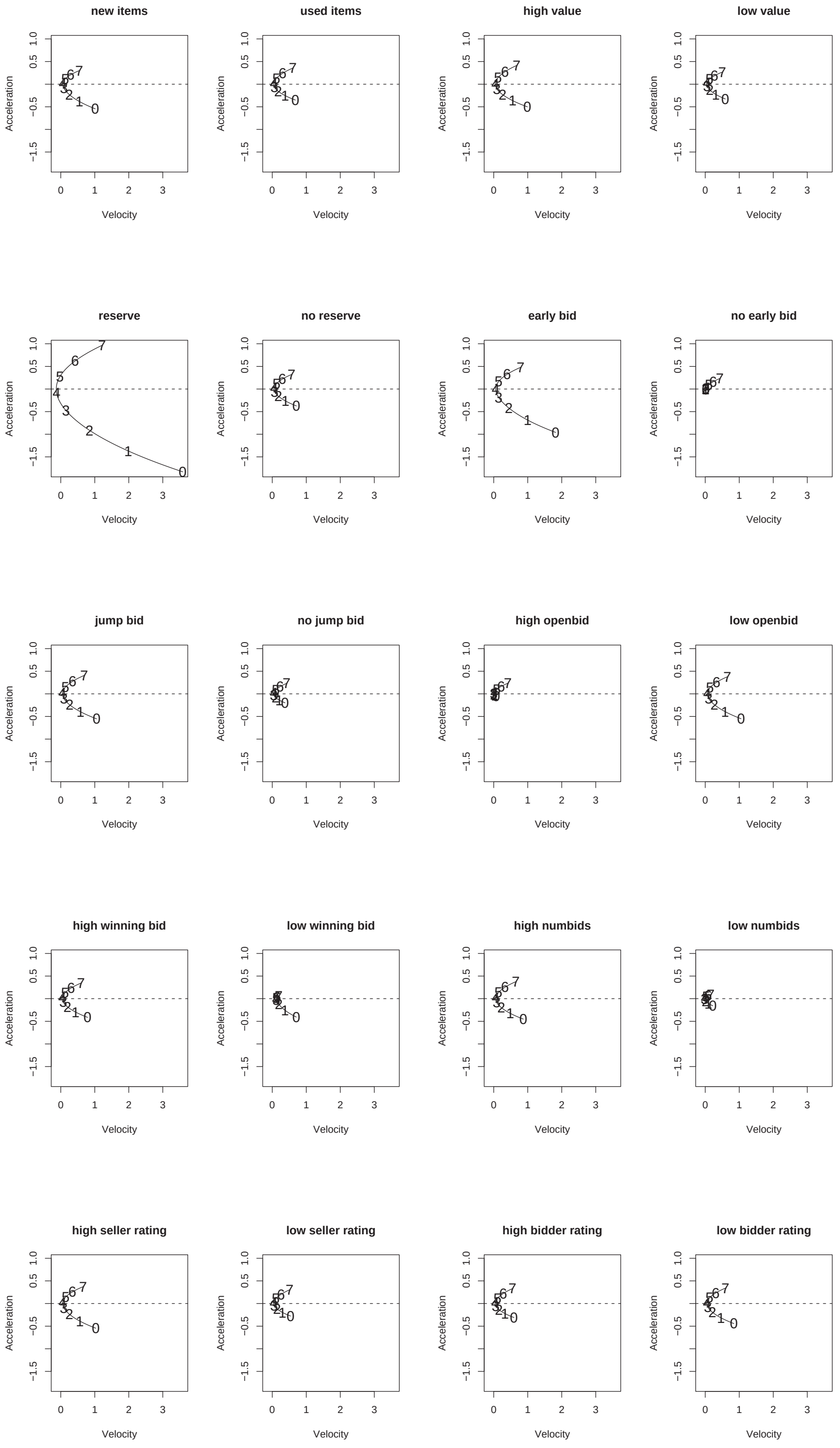


Figure 5.3: Conditional Phase Plane Plots (PPP) for the average price curve of the data, conditional on 10 auction characteristics from Table 1.

dynamics compared to lower-valued items. Early bidding seems to have a large effect on the dynamics not only at the auction-start but throughout the entire auction. For jump-bidding, we see that auctions that experience jump bids have a different relationship between velocity and acceleration compared to auctions without jump bids. Additional observations are that the opening bid and the number of bidders both have an impact on the dynamics. Interestingly, while different bidder ratings do not appear to make much of a difference, seller ratings do. We will revisit these findings from a more formal statistical angle in Section 4.

The exploratory analysis shown in this section indicates that dynamics exist, that they matter and that they are quite different from one auction to another. Moreover, some of the variation in dynamics appears to be driven by characteristics that are observable, such as characteristics of the product, the auction, the seller or the bidder. We also find that while dynamics vary, the general functional relationship between acceleration and velocity is the same "C"-shape for all auctions. The difference lies in the magnitude of that relationship. We take this as evidence that dynamics in online auctions can be captured using a single family of models. In the following, we derive dynamic models based on principal differential analysis and discuss a particular class of models suitable for online auctions dynamics.

## 5.3 The Differential Equation Model

Differential equations are widely used in the areas of engineering and physics. A differential equation describes a process with changing dynamics by finding relationships among the function and its derivatives. Many complex mechanical systems can be described in terms of differential equations. In the context of online auctions we view the price process as a dynamic system with many observed and unobserved factors acting upon it. We thus set out to find a differential equation model that can capture online auction dynamics.

Due to their relative novelty in statistical modeling, we open with a summary of how differential equation models are formulated, estimated, and evaluated in the functional setting (Sections 5.3.1-5.3.2). More details can be found in [86] (Chapter 13, 14). We then move to discussing a particular model that is very suitable for the auction context (Section 5.3.3). After that, we propose a new test for comparing models of auction sub-populations (Section 5.3.4).

### 5.3.1 Model Formulation and Estimation

Let $y_i$ be the price function for auction $i$ ($i = 1, \ldots, N$), recovered from the observed bid data, and let $D^m y_i$ be the $m^{th}$ derivative of $y_i$. Our goal is the identification of a linear differential operator (LDO) of the form

$$L(t) = \omega_0(t)I + \omega_1(t)D + \cdots + \omega_{m-1}(t)D^{m-1} + D^m \tag{5.1}$$

that satisfies the homogeneous linear differential equation $Ly_i(t) = 0$ for each observation $y_i(t)$. In other words, we seek a linear differential equation model so that our data satisfy

$$D^m y_i(t) = -\omega_0(t) - \omega_1(t) D y_i(t) - \cdots - \omega_{m-1}(t) D^{m-1} y_i(t). \tag{5.2}$$

An important motivation for finding the operator $L(t)$ is substantive: applications in the physical sciences, engineering, biology and elsewhere often make extensive use of differential equation models of the form

$$L y_i(t) = f_i(t). \tag{5.3}$$

The function $f_i(t)$ is often called a *forcing* or *impulse* function, and it indicates the influence of exogenous agents on the system defined by $Ly(t) = 0$. Returning to the online auction setting, we can reason that variation in price is due to variation in the forces resulting from bid placement and bid timing, and that these forces have a direct or proportional impact on the acceleration of the price process.

In practice, due to the prevalence of noise, it will be virtually impossible to find a model that satisfies (5.2) *exactly.* Hence, principal differential analysis adopts a least squares approach to the fitting of the differential equation model. The fitting criterion is to minimize the sum of squared norms

$$SSE_{PDA}(L) = \sum_{i=1}^{N} \int [L y_i(t)]^2 dt \tag{5.4}$$

over all possible models $L$. Notice that identifying $L$ is equivalent to identifying the $m$ weight functions $\omega_i$ that define the linear differential equation in (5.1).

Linear differential operators $L$ of degree $m$ (of the form (5.1)) have $m$ linearly independent solutions $u_j(t)$ of the homogeneous equation $Lu_j(t) = 0$. Although there is no unique way of choosing these $m$ functions $u_j(t)$, any choice is related by a linear transformation to any other choice. Therefore, any function $y(t)$ that satisfies $Ly(t) = 0$ can be expressed as a linear combination of the $u_j$'s, and since $L(t)$ is chosen to minimize the $Ly_i(t)$'s we expect to obtain a good approximation of the $y_i(t)$'s by expanding them in terms of the $u_j(t)$'s.

There are generally two approaches for estimating the weight functions $\omega_j(t)$. The first, pointwise minimization, yields pointwise estimates of the weight functions $\omega_j(t)$ by minimizing the (pointwise) fitting criterion

$$PSSE_L(t) = N^{-1}\sum_i (Ly_i)^2(t) = N^{-1}\sum_i [\sum_{j=0}^{m} \omega_j(t)(D^j y_i)(t)]^2, \tag{5.5}$$

where $\omega_m(t) = 1$ for all $t$ .

The pointwise approach can pose problems, especially if the $\omega_j(t)$'s are estimated at a fine level of detail. An alternative approach, which is computationally more efficient, is to use basis expansions. In the basis expansion approach, the weight functions $\omega_j$ are approximated by a fixed set of basis functions $\phi_k, k = 1, \ldots, K$. Let $\phi$ denote a $K$-dimensional vector of basis functions $(\phi_1, \ldots, \phi_K)'$. Then we assume that

$$\omega_j \approx \sum_k c_{jk}\phi_k, \tag{5.6}$$

where $c$ denotes the $(mK)$-vector of all basis function coefficients $c_{jk}$.

Using this approximation for the weight functions, we can now approximate $SSE_{PDA}(L)$ in (5.4) as a quadratic form in $c$, $\hat{F}(c|y)$, that can be minimized by standard numerical algebraic techniques. Specifically, we get

$$\hat{F}(c|y) = C + c'Rc + 2c's, \tag{5.7}$$

where the constant $C$ does not depend on $c$, and hence the estimate $\hat{c}$ is given by the solution of the equation $Rc = -s$. Moreover, the symmetric matrix $R$ is of order $mK$, and consists of an $m \times m$ array of $K \times K$ sub-matrices $R_{jk}$ of the form

$$R_{jk} = N^{-1} \int \phi(t)\phi(t)' \sum_i D^j y_i(t) D^k y_i(t) dt, \tag{5.8}$$

for $j = 0, \ldots, m-1$. The integrals involved in these expressions often have to be evaluated numerically (e.g., using the trapezoidal rule) over a fine mesh of equally-spaced values of $t$. For more details on the estimation of differential equation models, see [86] or [80].

### 5.3.2 Model Fit

An initial impression of the model fit can be obtained via visualization. If the model represents the data well, then the identified differential operator $L(t)$ should be effective at annihilating variation in the $y_i(t)$, and this can be visualized by plotting the *empirical forcing functions* $Ly_i(t)$. If the plotted $Ly_i(t)$'s are small and mainly noise-like, then the model provides good data-fit. As a point of reference for the magnitude of the $Ly_i(t)$'s we use the size of the $D^m y_i(t)$'s, since these are the empirical forcing functions corresponding to $\omega_0(t) = \ldots = \omega_{m-1}(t) = 0$.

To confirm visual impression, the quality of fit can also be gauged by more quantitative statistics. In the differential equation context, this can be done via the point-wise error sum of squares $PSSE_L(t)$ in (5.5). A logical baseline against which to compare $PSSE_L$ is the error sum of squares defined by a theoretical baseline model and its associated weight functions $\omega_j$:

$$PSSE_0(t) = \sum_i \left[ \sum_{j=0}^{m-1} \omega_j(t)(D^j y_i)(t) + (D^m y_i)(t) \right]^2 . \tag{5.9}$$

Since in this case there is no one particular model that forms the most reasonable baseline, we set $\omega_j(t) = 0$ so that the comparison is simply with the sum of squares of the $D^m y_i(t)$, which is analogous to the classical sum of squares in ANOVA. Thus, we can assess the model fit of the differential equation by examining the pointwise squared multiple correlation function

$$RSQ(t) = \frac{PSSE_0(t) - PSSE_L(t)}{PSSE_0(t)} \tag{5.10}$$

and the pointwise F-ratio

$$FRATIO(t) = \frac{(PSSE_0(t) - PSSE_L(t))/m}{PSSE_0(t)/(N-m)}. \tag{5.11}$$

### 5.3.3 A Second-order Linear Differential Equation Model

We now discuss in further detail a special case of the above general differential equation model: the second order linear differential equation. We focus particularly on second-order differential equations since our exploratory analyses in Section 5.2 indicated varying relationships between the first and second derivatives of price. Moreover, from a model-parsimony point of view, differential equation models of lower order are, a priori, preferred over models of higher order. Consider the general second-order differential equation

$$Ly_i = \omega_0 y_i + \omega_1 Dy_i + D^2 y_i = 0. \tag{5.12}$$

Setting $\omega_0 = 0$, we get

$$Ly_i = \omega_1 Dy_i + D^2 y_i = 0, \tag{5.13}$$

where $\omega_1(t)$ is a Lebesque square integrable function, and which describes a strictly monotone, twice-differentiable function $f$ [81]. The class of monotone functions discussed in this Chapter consists of those functions $f$ for which $ln(Df)$ is differentiable and $Dln(Df) = D^2 f/Df$ is Lebesque square integrable. Given that the live bid is monotonically increasing, equation (5.13) appears to be a reasonable candidate for online auctions. From here on out, for ease of notation, we write $\omega = \omega_1$ and $\omega^* = -\omega$.

## Data Simulation

In order to investigate the appropriateness of this class of models for the auction context, we simulate data from the second-order linear differential equation (5.13) and compare it to observed auction data. Simulating data from this model is done by generating solutions evaluated at an evenly spaced grid over the interval $[0, 7]$. We simulate the time-varying weight function $\omega^*(t)$ with overall linear trend according to a straight line with intercept $a = 1.647$ and slope $b = -0.407$, and local deviations from this trend using a linear combination of 4 Fourier basis functions. This results in the weight function depicted in the left panel of Figure 5.4. We then add Gaussian noise with mean zero and standard deviation 0.01. The resulting 190 simulated curves are shown in the middle panel of Figure 5.4. Also, the resulting PPP of the average acceleration versus the average velocity for these simulated curves is displayed in the right panel of Figure 5.4. We see that the simulated PPP strongly resembles the observed "C"-shapes from Section 2. This further supports the appropriateness of the class of differential equation models in (5.13) for modeling auction dynamics.

## Model Interpretation

A few comments on the implications of model (5.13) are in order. The coefficient function $\omega^* = -\omega = D^2y/Dy$ measures the relative curvature of the monotone function in the sense that it assesses the size of the curvature of $D^2y$ relative to the slope $Dy$. The special case of $\omega^* = -\alpha$ leads to $Y(t) = C_0 + C_1 \exp(\alpha t)$, whose ex-

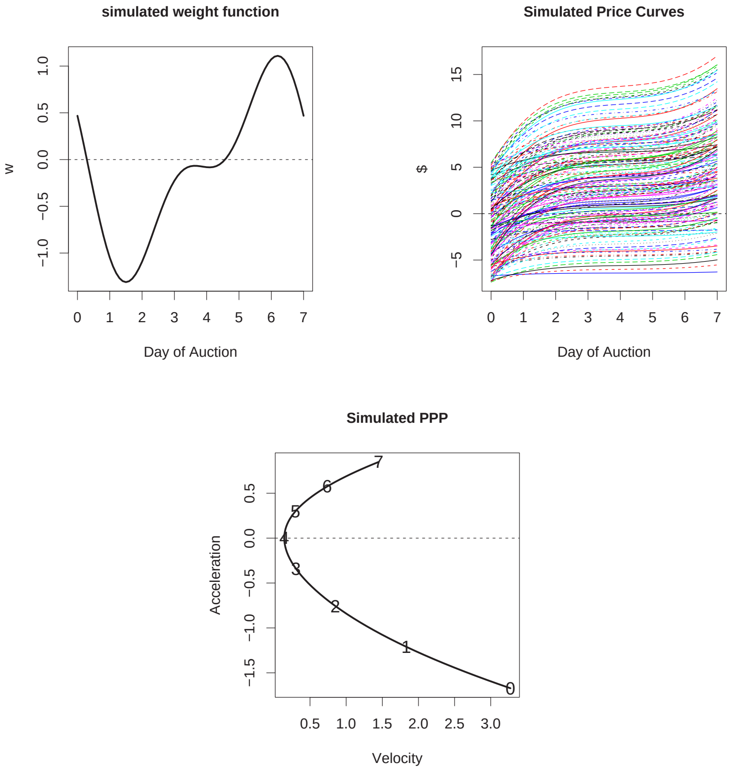


Figure 5.4: Simulation results for the monotone 2nd-order linear differential equation (5.13). Left: weight function $\omega^*(t)$ used to simulate data; Middle: 190 simulated curves; Right: resulting phase-plane plot.

ponent has constant curvature relative to $\alpha$, while $\omega^* = 0$ defines a linear function. Thus, small or zero values of $\omega^*(t)$ correspond to locally linear functions, whereas very large values correspond to regions of sharp curvature. In mechanical systems, the latter type is generally caused by internal or external frictional forces or viscosity. In the context of online auctions, sharp curvature in the price process can be related to jump bids caused by bidders attempting to apply external force ("determent of other bidders") to the bidding process. On the other hand, $\omega^* = 0$ indicates a very slowly moving price process which is often observed during the middle of the auction ("bidding drought").

### 5.3.4 Multiple Comparisons for Differential Equation Models

One of the goals of our study is to investigate whether factors that are related to the characteristics of the auction, item, seller, and bidders are associated with different dynamics. We therefore define $J$ auction sub-populations with dynamic models $D_1, D_2, \ldots, D_J$, that correspond to groupings according to the above mentioned characteristics. For example, groupings can be defined by item value: high-valued items vs. low-valued items. Another example is auctions where jump-bidding is present versus absent. We are therefore interested in testing whether the differential equation models of each of the $J$ groups are different. Although it is somewhat similar to multiple comparisons in the classical ANOVA setting, in our case we want to test whether $J$ population *models* are significantly different rather than $J$ population means. Our null hypothesis is therefore $H_0 : D_1 = \ldots = D_J$ vs.

the alternative $H_a$: at least one of the $D_i$'s is different. The problem that arises is that within each group, we have an associated single differential equation model and therefore there are no replications from which to estimate variances.

One possible approach is to use the functional shape test proposed by [49], which tests whether a functional object is equivalent to a given underlying curve. Rather than operating on mean functions as done in [49], we might compare the differential equations' coefficient functions. One important limitation of this test in our dynamic context is that it can only answer the general question "Do two differential equation models differ?". If differences between $J$ models exist, then the magnitude of their difference might change over different areas of the parameter space. For instance, for certain parameter values two models may be identical while they might differ substantially for other values. Evidence that this phenomenon exists in the online auction domain can be seen in Figure 5.3, where the relationship between acceleration and velocity (and thus, the corresponding differential equation models) is similar for some time periods, but very different for others.

Another difference between our context and the [49] setup is that we have a multiple-sample curve comparison whereas they deal with testing the fit of a single curve to a hypothesized curve.

We therefore propose a new multiple comparison test that is directly intended for comparing multiple functional objects (such as differential equation models) both locally and globally. The test captures not just overall curve difference but also differences at local areas, thereby enabling to answer questions such as "where do the curves differ and does the difference change over the range of the model parameters?". Our multiple comparison functional test is inspired by the work of [101] which considers multiple comparisons of several linear regression models. To date, most of the work on simultaneous inference and multiple comparisons has focused on comparing the means of $K$ ($\geqslant 3$) populations. An exception is the work of [101] and [61]. Spurrier [101] considers multiple comparisons of several simple linear regression lines and derives sets of simultaneous confidence bands for all possible contrasts between several simple linear regression lines over the entire range $(-\infty, \infty)$, assuming that the design matrices are the same. Liu et al [61] extend Spurrier's work to comparing multiple linear regression models that can have several explanatory variables and different design matrices. In the following we extend the work of [101] and [61] to multiple comparisons for differential equation models. We derive simultaneous confidence bounds for several PDA models and propose a way to implement the method in practice.

## A Multiple Comparison Test for Functional Differential Equations

Suppose there are $J$ groups in the population, and let $i$ ($i = 1, \ldots, J$) denote the index of $i$th group. Suppose further that each group can be described by a differential equation model of the form

$$-D^m Y_i(t) = \omega_0^i(t) Y_i(t) + \omega_1^i(t) DY_i(t) + \ldots + \omega_{m-1}^i(t) D^{m-1} Y_i(t) + e_i(t), \quad (5.14)$$

where $Y_i(t)^T = (y_{i1}, \ldots, y_{in_i})$, $D^p$ ($p = 0, \ldots, m$) is the $p-$th differentiation operator, and $e_i(t)^T = (\varepsilon_{i1}, \ldots, \varepsilon_{in_i})$ has components $\varepsilon_{ij}, j = 1, \ldots, n_i, i = 1, \ldots, J$, that are assumed *iid* $N(0, \sigma^2)$.

For $i = 1, \ldots, J$, let $H_i(t) = -(D^m Y_i)(t)$, let $D_i(t)$ be an $n_i \times m$ full column rank matrix with the $p$th ($p = 0, \ldots, m-1$) column given by $(D^p y_{i1}(t), \ldots, D^p y_{in_i}(t))^T$ defined at every single time point $t \in \mathcal{T}$, and let $\omega_i(t)^T = (\omega_0^i(t), \ldots, \omega_{m-1}^i(t))$. Then (5.14) can be expressed in matrix form as

$$H_i(t) = D_i(t)\omega_i(t) + e_i(t) \qquad i = 1, \ldots, J. \tag{5.15}$$

Holding $t$ fixed, the classical least squares estimate of $\omega_i(t)$ is

$$\hat{\omega}_i(t) = [D_i(t)^T D_i(t)]^{-1} D_i(t)^T H_i(t) \qquad i = 1, \ldots, J. \tag{5.16}$$

Let $\hat{\sigma}^2$ denote the pooled mean squared error estimate of $\sigma^2$ with degrees of freedom $\nu = \sum_{i=1}^{J}(n_i - m)$. Note that using classical linear models arguments, $\hat{\sigma}^2$ is independent of $\hat{\omega}$.

The goal is to construct a set of simultaneous confidence bands for

$$z^T\omega_i(t) - z^T\omega_j(t) = (z_0, \ldots, z_{m-1})\omega_i(t) - (z_0, \ldots, z_{m-1})\omega_j(t) \qquad (i, j) \in \Lambda, \tag{5.17}$$

where $\Lambda$ is an index set that determines the comparison of interest. Denote $\Delta_{ij} = (D_i^T D_i)^{-1} + (D_j^T D_j)^{-1}$. Then $\mathrm{Var}(z^T\omega_i - z^T\omega_j) = \sigma^2 z^T \Delta_{ij} z$, and the simultaneous confidence bands can be constructed as follows:

$$z^T\omega_i(t) - z^T\omega_j(t) \in z^T\hat{\omega}_i - z^T\hat{\omega}_j \pm c\hat{\sigma}\sqrt{z^T\Delta_{ij}z} \qquad \forall (i, j) \in \Lambda \tag{5.18}$$

where $c$ is the critical constant such that the confidence level is equal to $1 - \alpha$. We can compute $c$ via the relation $P(T < c)$, where

$$T = \sup_{(i,j)\in\Lambda} \frac{|z^T[(\hat{\omega}_i(t) - \omega_i(t)) - (\hat{\omega}_j(t) - \omega_j(t))]|}{\hat{\sigma}\sqrt{z^T \Delta_{ij}(t) z}}. \tag{5.19}$$

Finding an analytical representation for the distribution of $T$ is involved. In the following, we suggest a way of approximating it via simulation.

Let $P_{ij}$ be a $m \times m$ nonsingular matrix such that

$$P_{ij}^T P_{ij} = (D_i^T D_i)^{-1} + (D_j^T D_j)^{-1} \qquad \forall 1 \leqslant i \neq j \leqslant J. \tag{5.20}$$

Let $Z_i$ be independent normal random vectors distributed as $N(0, (D_i^T D_i)^{-1})$, $i = 1, \ldots, J$, independent of $\hat{\sigma}$. Let $Z_{ij} = (P_{ij}^T)^{-1}(Z_i - Z_j)$. Then the distribution of $T$ is the same as that of

$$\sup_{(i,j)\in\Lambda} \frac{|(P_{ij}z)^T Z_{ij}|}{(\hat{\sigma}/\sigma)\sqrt{(P_{ij}z)^T(P_{ij}z)}}. \tag{5.21}$$

We can then simulate a realization of the random variable $T$ as follows:

1. Calculate $P_{ij}, 1 \leqslant i \neq j \leqslant J$.

2. Simulate independently

$$Z_i \sim N(0, (D_i^T D_i)^{-1}) \qquad i = 1, \ldots, J \tag{5.22}$$

and

$$\hat{\sigma}/\sigma \sim \sqrt{\chi^2_\nu/\nu} \tag{5.23}$$

3. Calculate $Z_{ij} = (P_{ij}^T)^{-1}(Z_i - Z_j)$.

4. Compute $T$ via (5.21).

Repeat steps 1-4 $B$ times to obtain $B$ *iid* replications of the random variable $T$, $T_1, \cdots, T_B$. Calculate the estimate $\hat{c}$ of the critical constant $c$ as the $(1-\alpha)B$th largest simulated value of the $T_i$'s; that is, $\hat{c} = T_{((1-\alpha)B)}$, where $T_{(i)}$ denotes the ordered value of $T_i$.

Equation (5.18) defines simultaneous confidence bands for the difference between the coefficient functions of different differential equation models weighted by the vector $z$ with respect to a specific time $t$. Thus, for variable time, we should have a set of time-varying simultaneous confidence bands. That is to say, when time changes, the critical value $c$ in Equation 5.18 should be a function of time $t$. Repeat the procedure described above over a fine grid $t \in [0, 7]$, to obtain a vector of point-wise critical values $\hat{c}(t) = (\hat{c}_1, \ldots, \hat{c}_n)$, where $t = (t_1, \ldots, t_n)$ and $\hat{c}_j = \hat{c}(t_j)$.

Using $\hat{c}(t)$, we obtain the simultaneous confidence bands (5.18) as follows:

$$z^T\omega_i(t) - z^T\omega_j(t) \in z^T\hat{\omega}_i - z^T\hat{\omega}_j \pm \hat{c}(t)\hat{\sigma}\sqrt{z^T\Delta_{ij}z} \qquad \forall (i,j) \in \Lambda. \quad (5.24)$$

We consequently reject the null hypothesis that a set of models is identical if the confidence bands do not include zero.

## 5.4 Modeling eBay's Online Auction Data

We now return to our auction dataset. Our first goal is to evaluate whether auction dynamics can be captured by a differential equation. We therefore first estimate a model and evaluate its goodness-of-fit. Once such a model is established, we fit differential equations to sub-populations of the data and compare their dynamics by testing the difference between the models, using the proposed multiple comparison test. We describe each of these steps and the results next.

### 5.4.1 eBay Dynamics

We start by fitting a differential equation to the pre-processed data described in Chapter 3. We initially estimate model (5.13) using the entire set of auctions. The estimated coefficient function $\omega^* = -\omega$ is displayed in Figure 5.5. We can see that $\omega^*$ has three phases of values: negative, zero, and finally positive. These correspond to the three bidding phases during an auction: early activity, little mid-auction activity, and high late activity. The typical bidding behavior during an eBay auction consists of some early bidding, where bidders establish their time priority (when the two highest bids are tied, the earliest bidder is the winner). Then comes a period of "bidding drought", where there are hardly any bids placed (one possible reason is that bidders avoid revealing their willingness to pay too early to avoid that the price increases too much), and finally, during the last hours of the auction, bidding picks up again and dramatically peaks during the last auction minutes. This last moment bidding is called "sniping" and there are various explanations as to why bidders engage in it. One of these is to avoid bidding wars, because last moment bidding does not allow other bidders to respond. Returning to the shape of the estimated $\omega^*$ curve, recall that a value of zero indicates linear motion of the price process (i.e., no dynamics), whereas large positive or negative values are indicative of changes in the dynamics (oppressing them or increasing them, respectively). The first phase (up to day 3) is characterized by a negative $\omega^*$, with a dip on day 2. This negative dip marks the change from early bidding to "bidding draught", when velocity decreases. Then, we have $\omega^* = 0$ during the bidding draught, until bidding starts to increase again with a peak on day 6, in transition to high-intensity last moment bidding.

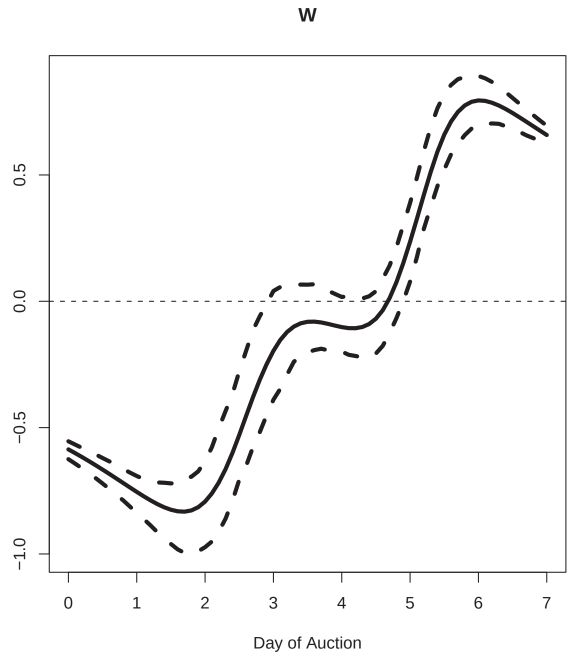


Figure 5.5: Estimated coefficient function of the monotone 2nd order differential equation fitted to online auction data.

The fit of the differential equation model can be gauged from Figure 5.6 which shows the equivalence of residual analysis and goodness-of-fit. The top panels show the observed price accelerations (left) and the estimated forcing functions (right) of the differential equation in (5.13). These two should be identical under perfect model-fit (similar to the observed and fitted observations in regression). We can see that although the fit is not perfect, the range of the forcing functions is identical to that of the observed accelerations during most of the auction. The fit is especially close in the middle and end of the auction; only the auction-start does not seem to be captured as well by the differential equation model.

As mentioned in Section 5.3.1, once the operator $L$ has been computed by estimating its weight functions $\omega_j$ by either point-wise minimization or basis expansion, we can compute a set of $m$ ($m = 2$ in this case) linearly independent basis functions $u_j$ satisfying $Lu_j = 0$. The plots in the second row of Figure 5.6 show the two solutions to the homogeneous differential equation $Lu = 0$ (left) and their corresponding derivatives (right). Recall that the price function is a linear combination of the solutions. One solution (dashed line) is simply a constant, which captures the overall monotone increasing nature of the price process. (Recall that price is transformed to the log-scale.) The second solution and its derivative (solid line) closely resemble the average bidding process and the average price velocity in Figure 3.8.

Finally, another quantification of model fit is given by the point-wise R-squared (RSQ) and F-ratio from (5.10) and (5.11). These are shown in the bottom panels of Figure 5.6. The point-wise RSQ is larger than 0.99 throughout the entire auction, indicating a very good global fit of the monotone second-order linear differential

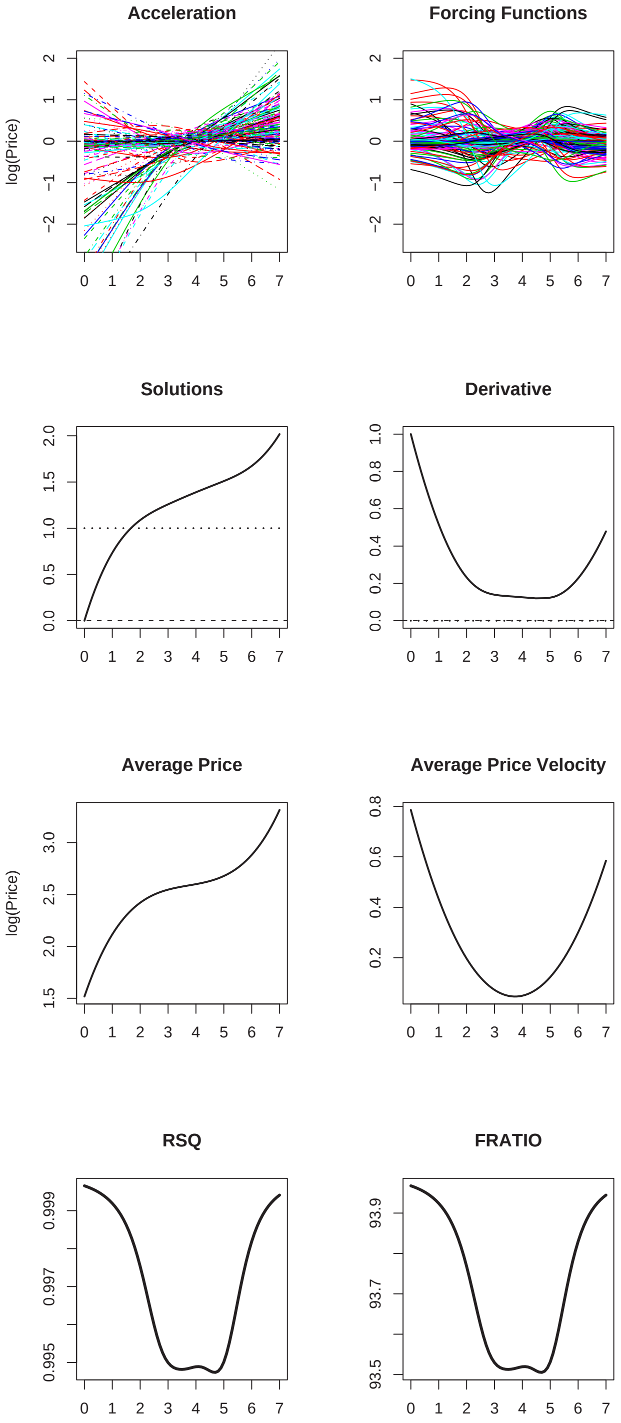


Figure 5.6: Model fitting results. Row 1: Comparison between the accelerations (2nd derivatives) and the observed forcing functions for the monotone 2nd order differential equation model; Row 2: Two solutions to the homogeneous differential equation $Lu = 0$; Row 3: Average price curve and average velocity curve; Row 4: Measures of model fit, RSQ and FRATIO. The x-axis shows the day of the auction.

equation model (5.13). We see that the fit also varies: it is best at the beginning and end of the auction, but somewhat weaker during mid-auction. A similar conclusion can be drawn from the point-wise F-ratio. The somewhat weaker model fit during mid-auction could be a result of the smaller number of observations during that time period (the "bidding draught" phase).

In summary, we learn that a second order differential equation model fits online auction data reasonably well. It captures the three phases of bidding and the interplay of dynamics that change over the course of the auction. We also see that the degree of model fit varies at different periods of the auction. This motivates our next step, which looks at *conditional* models for sub-populations of auctions. Perhaps the differences between these sub-populations can explain the variability in goodness of fit. But more importantly, our goal is to learn about the impact of different factors on the model parameters.

### 5.4.2 Dynamics of eBay Sub-Populations

Now that we have established that a differential equation model captures typical auction dynamics reasonably well, we explore the dynamics of different sub-populations using the multiple-comparisons procedure laid out in Section 3. To do this, we fit differential equation models separately to different auction sub-populations and test whether the resulting models are statistically different. We define sub-populations using characteristics of the auction, the item, the bidders, and the seller. These factors should, according to auction theory, affect the final price. In our case, we want to assess whether they also affect the dynamics of the entire price process. The factors that we consider are: item condition (new/used), item value (high/low), reserve price (yes/no), opening bid (high/low), early bidding (yes/no), jump bidding (yes/no), seller and bidder rating (high/low), winning bid (high/low) and number of bids (high/low) (see also the conditional PPPs in Figure 6). The parameter estimates of the fitted models are shown in Figure 5.7.

We can see that for some factors (e.g., condition and value) the basic shapes of the estimated coefficient functions $\omega^*$ are very similar. However, what *is* different is the timing and magnitude of this function. For instance, while new and used items appear to have almost identical dynamics during the first half of the auction, they differ in the second half. On the other hand, while low valued items appear to result in slightly higher dynamics during the first auction half, high valued items have a somewhat higher impact in the second half. Yet, while there are fine differences in dynamics for auctions of different condition and value, the overall similarity suggests that these two factors generally exhibit a very similar effect on auction dynamics, regardless of their value (this finding is further supported by testing their statistical difference, as shown below).

In contrast to the above factors, for other factors (e.g., winning bid or reserve price) the coefficient functions are very different. For instance in the case of the reserve price, auctions with a secret reserve price experience much more variability in the dynamics compared to auctions without one. For winning bid, auctions that end with a high winning bid see lower dynamics than those with a low winning bid. This result is probably correlated with the result for opening bid.

The above results indicate that among the various auction characteristics, different factors carry different weights not only on the final price but also on the price dynamics. They therefore possess different levels of explanatory power of the price process and perhaps even on predictability of the process. Awareness of this fact enables us to draw better inferences from the relationships between auction characteristics and auction price processes and to find a more explicit way to describe these relationships.

While Figure 5.7 suggests *some* differences between different auction subpopulations, the question arise whether these differences are also *statistically* significant. To that end, we employ the multiple comparison test derived in Section 5.3.4. The results are displayed in Figure 5.8. All confidence bands vary over time, emphasizing the time-varying sensitivity of our estimation procedure. As with classical confidence intervals, wider bands indicate a larger uncertainty about the true difference, and if the confidence band includes the value zero then the alternative of a population difference cannot be rejected for the time period under question (at a given significance level). Using these guidelines, we see the following: the presence of an early bid or a secret reserve price results in the largest difference in auction dynamics and this difference holds during most of the auction. Conversely, item condition, item value, and bidder rating result in essentially no difference in the auction dynamics during the entire auction period. For other factors (jump bidding, opening bid, or number of bids, etc.) the message is mixed: differences exist during some parts of the auction, but the difference is typically small.

In conclusion, it appears that different characteristics of the auction, the item,

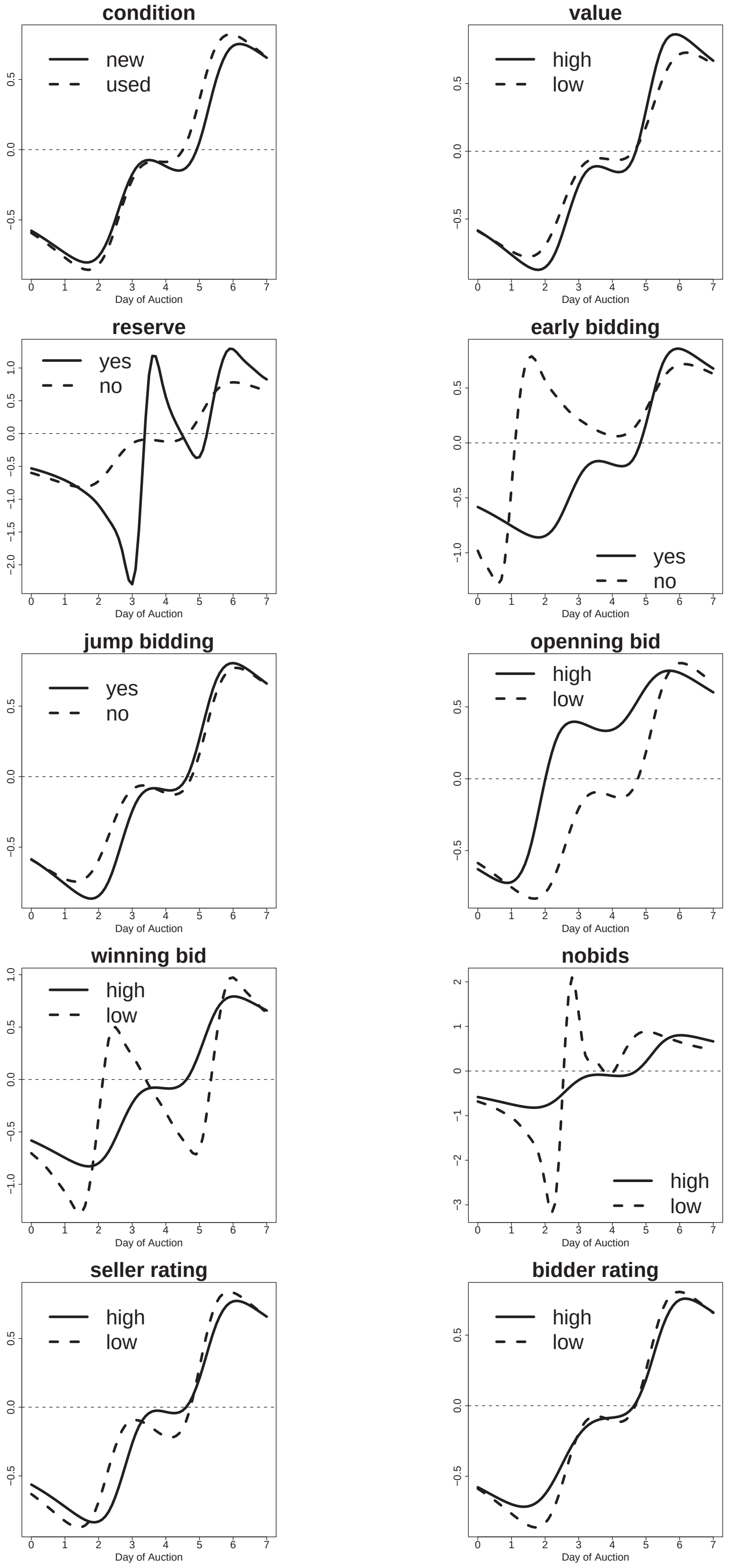


Figure 5.7: Estimated weight functions for the monotone 2nd order differential equation fitted to different online auction sub-populations.

the bidders and the seller lead to different price dynamics that change at different points in the auction and at different magnitudes. The timing and magnitude of the switching from early bidding to bidding draught, and from bidding draught to high-frequency late bidding can be different when factors such as item condition are considered.

## 5.5 Conclusions

In this work, we use linear differential equations to model the price process and its dynamics in online auctions, using a dataset on eBay auctions for Xbox play stations and Harry Potter books. We show that a monotone second-order linear differential equation model describes the relationship between the price and its dynamics well. We also explore the effect of different auction sub-populations on the dynamics and find that although auctions generally adhere to a three-phase process of price dynamics (early bidding, bidding draught, and bid sniping), certain sub-populations affect the timing and magnitude of changes in dynamics more than others.

This work is novel in several respects: To better understand the price dynamics in online auctions, we use an approach that directly captures process dynamics. Differential equations, in particular their functional version, are not (yet) a very common tool in statistics. Our work unveils a new and important application of this powerful tool set.

On the methodological side, we also propose a new multiple comparison test for

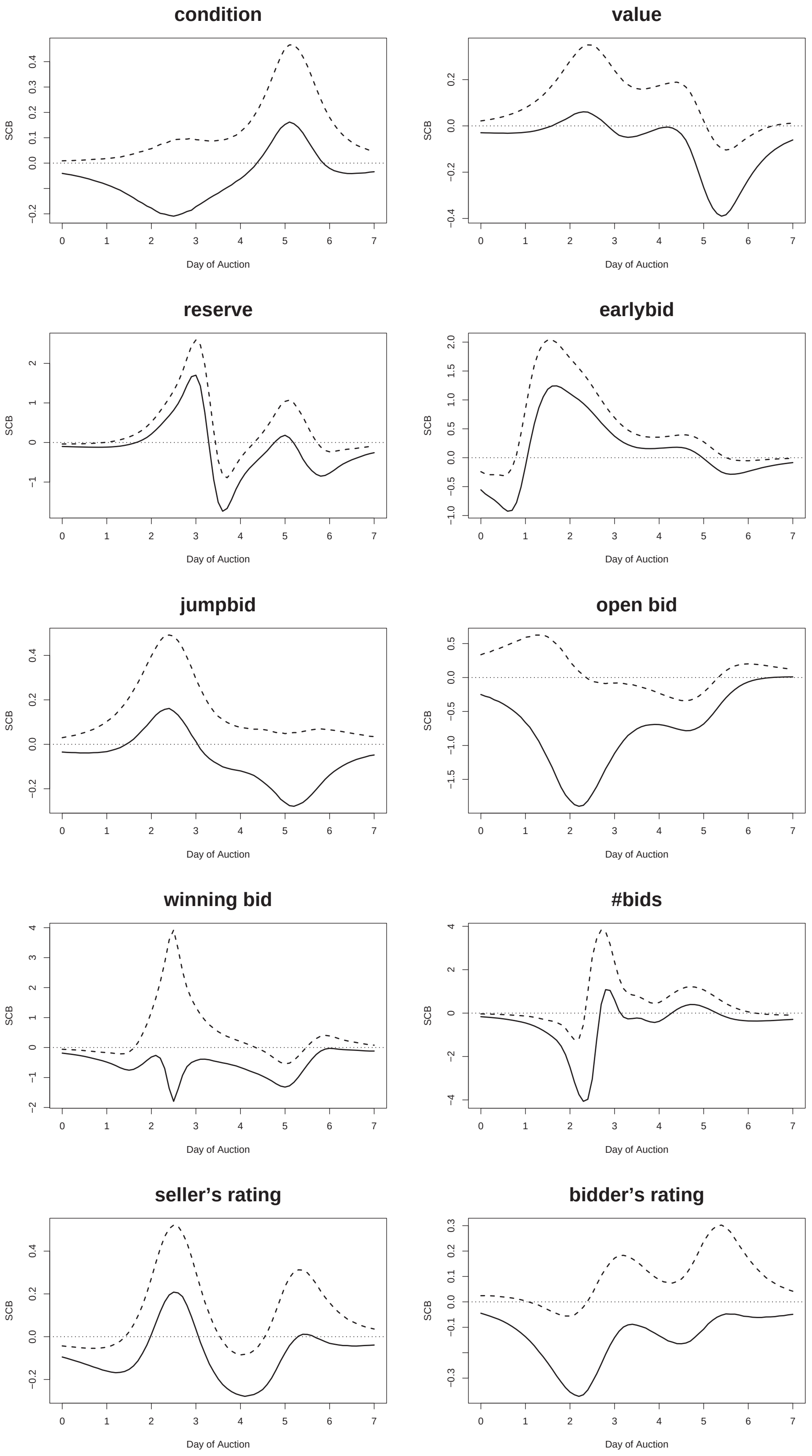


Figure 5.8: Simultaneous confidence bounds for the estimated weight functions from Figure 5.7.

functional models in the absence of replications. In our context, we use it to test the heterogeneity of dynamics across different auction sub-populations. The advantage of this new test is that it captures both global and local differences between models. This allows for the identification of differences indicative of timing and magnitude rather than merely overall shape.

# Chapter 6

# Model-based Functional Differential Equation Trees [3]

## 6.1 Introduction

In the previous chapter, we proposed a principal differential equation (PDA) framework to characterize online auctions and the factors that distinguish them. In doing so, we proposed a new multiple comparison test for differential equation models. While we found that auctions with different characteristics are associated with different dynamics, or more specifically, with different differential equation models, we found it hard to embed these characteristics into a common model. Thus, we now develop a novel modeling approach that can incorporate covariate information into a dynamic model. We accomplish this by combining the ideas of differential equations and regression tree models.

Tree models often give simple descriptions of complex, nonlinear relationships between several predictors and a univariate or multivariate response. A classical reference is the monograph *Classification and Regression Trees* by [16]. Tree-structured methods are extended in [91] to repeated measures and longitudinal data by modifying the split function so as to accommodate multiple responses.

Fitting a multivariate regression tree can be unsuccessful when the response is a high dimensional vector such as a continuous function. Two ways to fit trees to

[3] Paper in preparation

functional data are explored in [107]. Both approaches proceed by first reducing the dimensionality of the data and then fitting a standard multivariate tree to the reduced response. In the first approach, the dimensionality is reduced by representing the response as a linear combination of spline basis functions, while in the second one, the dimensionality is reduced using principal component analysis, retaining only the first several principal components.

Because constant fits in each node tend to produce large and hard to interpret trees (see e.g., [17]), research on incorporating (simple) parametric models into trees has recently received attention. Researchers from both statistical and machine learning communities have suggested algorithms to attach parametric models to leaf nodes or employ linear combinations to obtain splits in inner nodes. Such approaches are known as *functional trees* [32] in machine learning field with the most notable being 'M5' [75]. In statistics, Loh and his coworkers made key contribution in attaching parametric models to terminal nodes (see [63, 57, 17]). Regression trees with a constant fit in each terminal node are embedded into a maximum likelihood estimation framework in [102], where such trees are called "maximum likelihood trees". Furthermore, [109] takes the integration of parametric models into trees one step further by embedding recursive partitioning into statistical model estimation and variable selection. Within their framework, every leaf is associated with a conventionally fitted model such as, e.g., a maximum likelihood model or a linear regression. The model's objective function is used for estimating the parameters and the split points. This approach provides us the benefits of using the same objective function for partitioning as well as for parameter estimation. And the statistical formulation of the algorithm makes it easy to interpret the results.

Building on these ideas, we propose a functional-tree framework based on differential equation models. Our method allows the incorporation of dynamic models into the tree context. The incorporation of dynamics into trees of functional data makes our method new and different from extant methods, that either only deal with discrete observations, or only embed regular parametric models (such as linear regression models and maximum likelihood models) into trees.

Section 6.2 gives a brief overview of how regression trees are developed in the single-outcome setting and the modifications used to extend to multiple responses. A review of the methods of fitting trees to functional data and model-based recursive partitioning techniques is also given in this section. Section 6.3 presents our new method for estimating functional differential equation tree models. Our method attaches a differential equation model to every leaf node. Section 6.4 features a comparison of different tree models applied to our online auction data. Section 6.5 gives a brief conclusion.

## 6.2 Review of Regression Tree Methodology

This section gives a brief overview of how regression trees are developed in the single-outcome setting, the modifications necessary to extend to multiple responses, and also to a functional response. In the third part of this section, a brief review of the model-based recursive partitioning methods is given.

### 6.2.1 Univariate Response

Consider the familiar regression setting with $p$ predictors $X_1, X_2, \ldots, X_p$ and continuous response $Y$. We assume that complete data are available on all subjects. A regression tree is grown as follows. For each subgroup or *node*:

1. Examine every allowable split on each predictor variable.
2. Select and execute (create left and right daughter nodes) the *best* of these splits.

Steps 1 and 2 are then reapplied to each of the daughter nodes, and so on. The initial or *root* node comprises the entire sample. What constitutes an allowable split in Step 1 is defined in Chapter 2 of [16]. In short, the covariates are examined one at a time. For ordered covariates, an allowable split into two subsamples (nodes) is such that the covariate values in one node are all greater than those in the complementary node. The allowable splits therefore preserve ordering. For unordered categorical variables, any split into two disjoint subsets of the categories is permitted. "Best" in Step 2 is assessed in terms of the *split function* $\phi(s, g)$ that can be evaluated for any split $s$ of any node $g$. Two such split functions are espoused in [16]: least squares (LS) and least absolute deviations (LAD). The LS split function is made explicit below so that subsequent reformulations can be referenced.

Let $g$ designate a node of the tree. That is, $g$ contains a subsample $\{(x_i', y_i)\}$, where $x_i' = (x_{i1}, x_{i2}, \ldots, x_{ip})$. Let $N_g$ be the total number of cases in $g$ and let $\bar{y}(g) = N_g^{-1} \sum_{i \in g} y_i$ be the response average for node $g$. Then, the within node sum-of-squares is given by $SS(g) = \sum_{i \in g}(y_i - \bar{y}(g))^2$. Now suppose a split $s$ partitions

$g$ into left and right daughter nodes $g_L$ and $g_R$. The LS split function is $\phi(s, g) = SS(g) - SS(g_L) - SS(g_R)$, and the best split $s^*$ of $g$ is the split such that $\phi(s^*, g) = \max\limits_{s \in \Omega} \phi(s, g)$, where $\Omega$ is the set of all allowable splits $s$ of $g$.

An LS regression tree is constructed by recursively splitting nodes so as to maximize the above $\phi$ function. The function is such that we create smaller and smaller nodes of progressively increased homogeneity on account of the nonnegativity of $\phi$: $\phi \geqslant 0$ since $SS(g) \geqslant SS(g_L) + SS(g_R) \forall s$. This nonnegativity also holds for least absolute deviations and is an essential property of a split function.

### 6.2.2 Multiple Responses

The regression tree methodology built in [16] was extended in [91] to repeated measures and longitudinal data by modifying the split functions so as to accommodate multiple responses. Several split functions are developed based either on deviations around subgroup mean vectors or on two sample statistics measuring subgroup separation.

In the multiple response setting, we consider a situation in which, besides the vector of covariances, each individual has a $T \times 1$ vector of responses $y_i' = (y_{i1}, y_{i2}, \ldots, y_{iT})$ is considered. Define $V(\theta, g)$ to be the model covariance matrix of the responses for node $g$ depending on unknown parameters $\theta$. Allowing $\hat{\theta}$ to be the $T(T + 1)/2$ sample covariances $s_{jk}$ enables us to proceed without making any assumptions on the covariance structure. However, both efficiency and interpretation gains can be made by restricting the dimension of $\theta$. Considering the well known instability resulting from overparameterizing covariance matrices addressed in [23], low dimensional $\theta$ is always preferred. We write $\mu(g)$ to denote the $T \times 1$ vector of response means for individuals within a given node $g$.

Two types of split functions are developed to handle multiple response data by [91]: one that focuses on the mean structure with the covariance as nuisance, and another that primarily focuses on the covariance structure.

## Mean Structure

An immediate generalization of the least squares split function for the single outcome case given above is obtained by replacing $SS(g)$ with

$$SS(g) = \sum_{i \in g} (y_i - \mu(g))^T V(\theta, g)^{-1} (y_i - \mu(g)). \tag{6.1}$$

Then, the split function $\phi_m$ for evaluating a split $s$ of $g$ into $g_L$ and $g_R$ is as before:

$$\phi_m(s, g) = SS(g) - \left[SS(g_L) + SS(g_R)\right]. \tag{6.2}$$

This function allows for a different covariance matrix for each of $g$, $g_L$, and $g_R$ because the parameter estimates $\hat{\theta}$, $\hat{\theta}_L$, $\hat{\theta}_R$ can differ. To ensure that $\phi_m \geqslant 0$ and that maximizing $\phi_m$ improves homogeneity, it is required that for each candidate split, the covariance parameters are determined from the parent node $g$ so that

$$V(\theta, g) = V(\theta_L, g_L) = V(\theta_R, g_R), \tag{6.3}$$

and only the mean function is updated. Having determined and implemented the best split, the resulting daughter nodes become the new parent nodes and the covariance parameters are reestimated for each.

## Covariance Structure

It has been pointed out by researchers that heterogeneity in longitudinal data can also affect covariances. In [69] variance heteroscedasticity is modeled as a function of covariates in the generalized linear models framework with a univariate outcome. For a multivariate outcome, covariance heteroscedasticity is described as a function of covariates using the regression tree paradigm in [91]. First one accounts for the mean structure, and then applies the split functions to residuals for detecting covariance heterogeneity.

Analogous to the within node measures of loss, functions that assess how closely the sample covariance matrix conforms to the hypothesized covariance matrix are considered in [91]. Conformity is measured via a matrix norm:

$$\phi_c(s,g) = \log(\|S(g) - V(\theta,g)\|) - [\log(\|S(g_L) - V(\theta_L,g_L)\|) - \log(\|S(g_R) + V(\theta_R,g_R)\|)]\,. \tag{6.4}$$

The preceding form is motivated by analogy with the normal theory of likelihood ratio test for equality of covariance matrices. The matrix norm $\|\cdot\|$ can be selected in accordance with what constitutes a meaningful distance measure for the problem at hand. A common choice is the squared Euclidean norm, which affords simple updating algorithms for several basic choices for $V$. An alternate loss function is presented in [62].

### 6.2.3 Functional Response

While regression trees were successfully applied to longitudinal data in [91] and [92], they could be unsuccessful if the response is a high dimensional vector that can be thought of as a discretization of a continuos response (or so called *functional response*). Such a problem is illustrated in [107] in an international call example. The problem in [107] is to predict a customer's time-of-day pattern for international calling from the information in the customer's first two international calls. In Section 3 of [107], they show that fitting a standard multivariate tree to time of day distributions represented as histograms gives a poor fit and decision rules that are not sensible. The authors present two procedures that reduce the dimension of the response and then fit a multivariate decision tree to lower dimensional responses. In the first approach, each individual's response curve is represented as a linear combination of spline basis functions, penalizing for roughness, and then a multivariate regression tree is fit to the coefficients of the basis functions. In the second, a multivariate regression tree is fit to the first several principal component scores of the multivariate responses. It is shown that the decision rules based on the spline tree and the principal component tree are similar and both lead to sensible results.

### Spline Trees

Denote the functional response for individual $i$ by $Y_i(t), i = 1, \ldots, N$, where $t = (t_1, \ldots, t_m)$ are the time points of the observed discretized response values. If $Y_i(t)$ is smooth, then it can be approximated by a linear combination of basis functions $\{\beta_1, \ldots, \beta_q\}$, and the coefficients of the linear combination for each individual can be used as the response for a multivariate tree. If a roughness penalty is imposed on the approximation, then each response is approximated by only a few basis functions, and the response vector is low dimensional. Generally, the lower the dimension of the response vector, the faster multivariate trees can be fit.

To fit a spline tree, take

$$Y_i(t) = f_i(t) + \epsilon_i(t), \tag{6.5}$$

where

$$f_i(t) = \sum_{j=1}^{q} \delta_{ij}\beta_j(t) \tag{6.6}$$

for a set of basis functions $\beta = (\beta_1(t), \ldots, \beta_q(t))$ and a coefficient vector $\delta_i = (\delta_{i1}, \ldots, \delta_{iq})^T$, where $\epsilon_i(t)$ is white noise with mean zero and constant variance. Then the $q \times N$ estimated coefficient matrix $\hat{\delta} = \left[\hat{\delta_1}, \ldots, \hat{\delta_N}\right]$ minimizes the penalized sum of squares

$$S(\delta) = \sum_{i=1}^{N}\sum_{j=1}^{m}(Y_i(t_j) - f_i(t_j))^2 + \lambda \int [D^2 f(t)]^2 dt. \tag{6.7}$$

Green and Silverman (1994) show that $\hat{\delta}$ is defined by

$$\hat{\delta} = (\beta^T\beta + \lambda K)^{-1}\beta^T \mathrm{Y}(t), \tag{6.8}$$

where $\beta$ is the $m \times q$ basis matrix, $\mathrm{Y}(t)$ is the $m \times N$ matrix of responses, $\lambda$ is a smoothing parameter that is the same for all observations, and $K_{jk} = \int D^2\beta_j(t)D^2\beta_k(t)dt$.

The estimated coefficients $\hat{\delta}_i, i = 1, \ldots, N$ are then used as responses in the multivariate regression tree. Fitting proceeds as in Section 6.2.2, except that now the "responses" are the estimated coefficient vectors instead of the original responses of long vectors and the "prediction" at a node is the mean estimated coefficient vector for responses in the node. The least squares split function for the current node $g$ is thus given by

$$SS(g) = (\hat{\delta}_i - \bar{\hat{\delta}})^T\beta^T\beta(\hat{\delta}_i - \bar{\hat{\delta}}) \tag{6.9}$$

where $\hat{\delta}_i$ is the estimated coefficient and $\bar{\hat{\delta}}$ is the mean estimated coefficient vector for responses in the current node. As before, splitting proceeds by comparing the split function $SS(g)$ before splitting to the split function $SS(g_L) + SS(g_R)$ after splitting, choosing the split that gives the largest decrease in the total least squared loss. In other words, the best split $s^*$ of $g$ is the split which maximizes $\phi(s, g) = SS(g) - [SS(g_L) + SS(g_R)]$. Namely, $\phi(s^*, g) = \max_{s\in\Omega} \phi(s, g)$, where $\Omega$ is the set of all allowable splits $s$ of $g$.

Note that the predicted curve $\hat{f(t)}$ at each node can be computed as

$$\hat{f(t)} = \sum_{j=1}^{q} \bar{\hat{\delta}}_j \beta_j(t). \tag{6.10}$$

## Principal Components Trees

Instead of reducing the dimension of the response by treating it as a curve, [107] also reduces the dimension by treating it as a vector and applying principal components analysis, retaining only the first several principal components. The authors take

$$\gamma_i = \sum_{j=1}^{m} \theta_j Y_i(t_j), \tag{6.11}$$

where $\theta_j$ is the weighting coefficient and the principal component scores $\gamma_i$ are the uncorrelated linear combinations of the response $Y(t)$ with variances that are as large as possible. More details may be found in [5].

The first several principal components that explain a great portion of the total variance are used as responses in a multivariate regression tree. In [107], the first six principal components in their international call application were used to fit the tree, since the first six components together explain 45% of the total variance. The split function for the principal component tree is similar to the spline tree. And the predicted curve $\hat{f(t)}$ at each node is again computed by the average of the $\hat{f}_i(t)$ in the node as in equation (6.10).

### 6.2.4 Model-Based Recursive Partitioning Methods

As mentioned in Section 6.1, incorporation of (simple) parametric models into trees has received increasing interest over the past decade. Several algorithms have been suggested both in the statistical and machine learning communities that attach parametric models to terminal nodes or employ linear combinations to obtain splits in inner nodes (see [32, 75, 63, 57, 17, 18, 102]). Based on the ideas of the above research, in [109] the integration of parametric models into trees is carried one step further. A rigorous theoretical foundation is provided by introducing a new unified framework that embeds recursive partitioning into statistical model estimation and variable selection. In this section, we give a brief review of that work.

A parametric model $M(Y, \theta)$ is considered in the work of [109], where $Y$ are (possibly vector-valued) observations and $\theta \in \Theta$ is a $k$-dimensional vector of parameters. Given $N$ observations $Y_i$ $(i = 1, \ldots, N)$ the model can be fitted by minimizing some objective function $\Psi(Y, \theta)$ yielding the parameter estimate $\hat{\theta}$

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{i=1}^{N} \Psi(Y_i, \theta). \tag{6.12}$$

Estimators of this type include various well-known estimation techniques, the most popular being ordinary least squares (OLS) or maximum likelihood (ML) among other M-type estimators. In the case of OLS, $\Psi$ is typically the error sum of squares and, in the case of ML, it is the negative log-likelihood.

Let $(Z_1, \ldots, Z_L)$ be a set of partitioning variables. It is assumed that a partition $\{\beta_b\}_{(b=1,\ldots,B)}$ of the space $Z = Z_1 \times \cdots \times Z_L$ exists with $B$ cells (or segments) such that within each cell $\beta_b$, a model $M(Y, \theta_b)$ with a cell-specific parameter $\theta_b$ holds.

The basic idea of the recursive partitioning algorithm is that each node is associated with a single model. First, the designated model $M(Y, \theta)$ is fitted to all observations in the current node by estimating $\hat{\theta}$ via minimization of the objective function $\Psi$. Second, a fluctuation test for parameter instability with respect to every ordering $Z_1, \ldots, Z_L$ is performed to assess whether splitting the node is necessary. If there is significant instability with respect to any of the partitioning variables $Z_l$, the variable $Z_l$ associated with the highest parameter instability is selected. Third, the split point(s) that locally optimize $\Psi$ is (are) computed. Finally, the node is split into $B$ locally optimal segments and the procedure is repeated. If no more significant instabilities can be found, the recursion stops and returns a tree where each terminal node is associated with a model of type $M(Y, \theta)$.

## 6.3 Model-Based Functional Differential Equation Trees

Following the idea of [109], we now establish a functional-tree framework based on differential equation models. Our method allows the incorporation of dynamics via differential equations into the tree context. We combine the PDA techniques described in [86] for fitting differential equation models to functional data and the recursive partitioning method proposed in the work of [109] to construct functional trees with each terminal node associated with a certain differential equation model. At the same time, both splitting of the nodes and estimation of the parameters are based on the same objective function. The incorporation of dynamics into trees of functional data makes our method new and different from the methods described previously. Previous methods either only deal with discrete observations or only embed regular parametric models such as linear regression models and maximum likelihood models into trees.

Consider a differential equation model

$$D^m y_i = -\omega_0 y_i - \omega_1 D y_i - \ldots - \omega_{m-1} D^{m-1} y_i \tag{6.13}$$

where $y_i, i = 1, \ldots, N$ are functional observations and coefficient functions $\omega_j, j = 0, \ldots, m-1$ are functions of time $t$. To find such a differential equation model we need to identify a linear operator

$$L = \omega_0 I + \omega_1 D + \ldots + \omega_{m-1} D^{m-1} + D^m \tag{6.14}$$

that comes as close as possible to satisfying the homogeneous linear differential equation $Ly_i = 0$ for each observation $y_i$. Since we wish the operator $L$ to annihilate the given data functions $y_i$ as nearly as possible, we regard the function $Ly_i$ as the residual error from the fit provided by the linear differential operator $L$. Then the model can be fitted by minimizing the sum of squared norms

$$SSE_{PDA}(L) = \sum_{i=1}^{N} \int [Ly_i(t)]^2 dt = \sum_{i=1}^{N} \int [\sum_{j=0}^{m} \omega_j(t)(D^j y_i)(t)]^2 dt \tag{6.15}$$

which can be minimized over the $m$ weight functions $\omega_i$. Note that $\omega_m(t) = 1$ for all $t$ as in Eq. (6.14). We define Eq. (6.15) as our objective function and denote it by $\Psi(Y, \omega)$. The parameter estimate for $\omega$, given $N$ observations $Y_i$ $(i = 1, \ldots, N)$, can be represented by

$$\hat{\omega} = \arg\min_{\omega \in \Omega} \sum_{i=1}^{N} \Psi(Y_i, \omega). \tag{6.16}$$

For ease of reference, we denote the model (6.13) by $M(Y, \omega)$.

We have seen in the previous chapter that differential equations provide a good representation of auction dynamics. But we have also seen that dynamics vary by auction sub-populations such as items for high vs. low price. Our goal is to develop a differential equation methodology that accounts for this variability.

The basic idea is that each node is associated with a single model. To assess whether splitting the node is necessary, a fluctuation test for parameter instability is performed. If there is significant instability with respect to any of the partitioning variables $Z_l$, we split the node into $B$ locally optimal segments and repeat the procedure. If no more significant instabilities can be found, the recursion stops and returns a tree where each terminal node is associated with a differential equation model $M(Y, \omega)$.

More formally, we assume that a partition $\{\beta_b\}_{(b=1,\ldots,B)}$ of the spaces $Z = Z_1 \times \ldots \times Z_L$ exists with $B$ cells (or segments) such that in each cell $\beta_b$ a differential equation model $M(Y, \omega)$ with a cell-specific parameter $\omega_b$ holds. We denote this segmented model by $M_\beta(Y, \omega)$ where $\omega$ now the full combined parameter $\omega = (\omega_1, \ldots, \omega_B)^T$

The steps of the algorithm are as follows:

1. Fit a differential equation model $M(Y, \omega)$ to all observations in the current node by estimating $\hat{\omega}$ via minimization of the objective function $\Psi$.

2. Assess the stability of the parameters w.r.t. every ordering $Z_1, \ldots, Z_L$. If there is some overall instability, choose the variable $Z_{l^*}$ associated with the smallest $p$-value (or the highest parameter instability) for partitioning, otherwise stop.

3. Search for the locally optimal split point(s) in $Z_{l^*}$ by minimizing the objective function of the model $\psi$.

4. Split the node into daughter nodes and repeat the procedure.

The details for steps 1-3 are specified next. To keep notation simple, the dependence on the current segment is suppressed and the symbols established for the global model are used, i.e., $N$ for the number of observations in the current node, $\hat{\omega}$ for the associated parameter estimate and $B = 2$ for the number of daughter nodes chosen.

### 6.3.1 Parameter Estimation via Basis Expansion

To get smooth estimates of the weight functions $\omega_j$, [86] uses a fixed set of basis functions to approximate them. Let $\phi_k, k = 1, \ldots, K$ be a set of $K$ such basis functions, and let $\phi$ denote the $K$-dimensional vector function $(\phi_1, \ldots, \phi_K)^T$. We assume that

$$\omega_j \approx \sum_k c_{jk} \phi_k \tag{6.17}$$

where the $mK$-th coefficients $c_{jk}$ define the approximations and must be estimated from the data. Let the $(mK)$-vector $c$ contain these coefficients, where index $k$ varies within index $j$.

We can approximate the criterion $SSE_{PDA}(L)$ in terms of $c$ as a quadratic form $\hat{F}(c|y)$ that can be minimized by standard numerical algebraic techniques. We have

$$\hat{F}(c|y) = C + c^T Rc + 2c^T s \tag{6.18}$$

where the constant $C$ does not depend on $c$, and hence the estimate $\hat{c}$ is given by the solution of the equation $Rc = -s$:

$$\hat{c} = -R^T s. \tag{6.19}$$

The symmetric matrix $R$ is of order $mK$, and consists of an $m \times m$ array of $K \times K$ submatrices $R_{jk}$ of the form

$$R_{jk} = N^{-1} \int \phi(t)\phi(t)^T \sum_i D^j y_i(t) D^k y_i(t) dt \tag{6.20}$$

for $j = 0, \ldots, m-1$.

### 6.3.2 Testing for Parameter Instability

We start partitioning based on some simple rule, i.e., the variances of the variables. We do each splitting based on the variable which has the largest variance in the current node. For example, assume that there are N observations $y_i, i = 1, \ldots, N$ in the current node and $L$ partitioning variables $Z_l, l = 1, \ldots, L$. To decide on which variable to split the current node, we compare the variance of those $L$ variables $Z_l, \ldots, Z_L$. The $Z_l$ with the highest variance will be chosen as the splitting variable. Using an exhaustive search, we find the minimal value of the objective function $\Psi(Y, \omega)$.

While the rule described above is a simple enough starting point, there exist cases that cannot be solved by this rule: There is no way to assess the stability of the parameters with respect to every ordering $Z_1, \ldots, Z_L$. The splitting step we address is of no different from the classical regression tree except for the objective function.

As an alternative, we adopt the parameter instability assessing method described by [109]. The basic idea of their method is to check whether the score functions $\hat{\psi}_i$ ($\hat{\psi}_i = \hat{\psi}(Y_i, \hat{\omega}), \psi = \frac{\partial}{\partial \omega}\Psi(Y, \omega)$) fluctuate randomly around their mean 0 or exhibit systematic deviations from 0 over $Z_l$. These deviations can be captured by the empirical fluctuation process

$$W_l(t) = \hat{J}^{-1/2} N^{-1/2} \sum_{i=1}^{\lfloor Nt \rfloor} \hat{\phi}_{\sigma(Z_{il})}, \quad (0 \leqslant t \leqslant 1) \tag{6.21}$$

where $\hat{J} = N^{-1} \sum_{i=1}^{N} \psi(Y_i, \hat{\omega}) \psi(Y_i, \hat{\omega})^T$ is an estimate of the covariance matrix $COV(\psi(Y, \hat{\omega}))$, and $\sigma(Z_{il})$ is the ordering permutation which gives the antirank of the observation $Z_{il}$ in the vector $Z_l = (Z_{1l}, \ldots, Z_{Nl})^T$. Thus, $W_l(t)$ is simply the partial sum process of the scores ordered by the variable $Z_l$, scaled by the number of observations $n$ and a suitable estimate $\hat{J}$ of the covariance matrix $COV(\psi(Y, \hat{\omega}))$. This empirical fluctuation process is governed by a functional central limit theorem under appropriate assumptions and null hypothesis of parameter stability. These assumptions include: (1) The $N$ observations $Y_i$'s are independent and distributed according to some distribution $F$ with $m$-dimensional parameter $\omega$ (The independence is assumed here for convenience and can be weakened in practice); (2) The observations are uniquely ordered by some external variable; (3) The estimate $\hat{J}$ of the covariance matrix is non-singular. We are interested in testing the hypothesis $H_0 : \omega_j = \omega^*, j = 0, \ldots, m-1$. As shown in [108] and [109], under the assumptions stated above and under $H_0$, such empirical fluctuation process converges to an $m$-dimensional Brownian bridge $W^0$. A test statistic can be derived by applying a scalar functional $\lambda(\cdot)$ capturing the fluctuation in the empirical process to the fluctuation process $\lambda(W_l(\cdot))$ and the corresponding limiting distribution is simply the same functional applied to the limiting process $\lambda(W^0(\cdot))$. The corresponding $p$-value $p_l$ can be computed. To test whether there is some overall instability in the current node, we check whether the minimal $p$-value falls below a pre-specified significance level $\alpha$. If this is the case, the variable $Z_{l*}$ associated with the minimal $p$-value is chosen for splitting the model in the next step of the algorithm.

The general framework for testing parameter stability described here (and also in [109]) is called a generalized M-fluctuation test and has been established by [108]. A large number of structural change tests suggested both in the econometrics and statistics literature has been shown to be encompassed in this framework, and [109] give an overview of these tests. In principle, each of the tests from this framework can be used in the recursive partitioning algorithm, but two different test statistics seem to be particularly attractive and we employ them to assess numerical and categorical partitioning variables $Z_l$ respectively.

**Assessing Numerical Predictor Variables:** The sup$LM$ statistic proposed by [6] is suitable for capturing the instabilities over a numerical variable $Z_l$:

$$\lambda_{supLM}(W_l) = \max_{i=\underline{i},\ldots,\bar{i}} \left( \frac{i}{N} \frac{N-i}{N} \right)^{-1} \|W_l \left( \frac{i}{N} \right) \|_2^2, \tag{6.22}$$

which is the maximum of the squared $L_2$ norm of the empirical fluctuation process scaled by its variance function. This type of statistic first appeared in [6], and can be interpreted as the $LM$ statistic against a single change point alternative where the potential change point is shifted over the interval $[\underline{i}, \bar{i}]$. The interval is typically defined by requiring some minimal segment size $\underline{i}$ and then $\bar{i} = N - \underline{i}$. The limiting distribution of (6.22), as shown in [6], is given by the supremum of a squared, m-dimensional tied-down Bessel process $sup_t \left(t(1-t)\right)^{-1} \left\|W^0(t)\right\|_2^2$ from which the corresponding $p$-value $p_l$ can be computed.

**Assessing Categorical Predictor Variables:** By definition, a categorical variable $Z_l$ with $C$ different levels or categories has ties and a total ordering of the observations is not available. Therefore, a different statistic is needed to capture its instability. An appropriate statistic is one that is insensitive to the ordering of the $C$ levels and of the ordering of observations within each level. As described in [39], one such statistic can be developed as follows. Divide the span of category levels into $C$ windows $I_1, \ldots, I_C$. For component $l$, the test statistic is given by

$$\lambda_{\chi^2}(W_l) = \sum_{c=1}^{C} \frac{|I_c|}{N}^{-1} \left\| \Delta_{I_c} W_l \left( \frac{i}{N} \right) \right\|_2^2 \tag{6.23}$$

where $|I_c|$ is the length of interval $I_c$ (namely, number of observations in category $c$), and $\Delta_{I_c} W_l$ is the increment of the empirical fluctuation process over the observations in category $c = 1, \ldots, C$ (i.e., essentially the sum of the scores in category c). The test statistic is then the weighted sum of the squared $L_2$ norm of the increments which has an asymptotic $\chi^2$ distribution with $m \cdot (C-1)$ degrees of freedom. The $p$-value $p_l$ can also be computed correspondingly. More details are given in [39, 109].

As mentioned above, the score function we use to assess the parameter instability is $\hat{\psi}_i$ ($\hat{\psi}_i = \hat{\psi}(Y_i, \hat{\omega}), \psi = \frac{\partial}{\partial \omega} \Psi(Y, \omega)$). Because problems arise when we differentiate with respect to the infinite dimensional weight function $\omega(t)$, we use the basis expansion method again to reexpress $\psi$.

It has been showed in Section 6.3.1 that the weight function can be represented by (6.17), where $\phi = (\phi_1, \ldots, \phi_K)^T$ is a K-dimensional vector of basis functions. Let $c_j = (c_{j1}, \ldots, c_{jK})^T$ be a K-vector of the coefficients. Then Eq. (6.17) can be expressed in matrix form

$$\omega_j(t) \approx c_j^T \phi(t). \tag{6.24}$$

Substituting $\omega_j(t)$ in the objective function $\Psi(X, \omega) = SSE_{PDA}(L)$ as given in Eq.

(6.15) with its matrix representation, we get

$$\Psi(X,c) = \sum_{i=1}^{N} \int \left[ \sum_{h=0}^{m} c_h^T \phi(t) D^h y_i(t) \right]^2 dt. \tag{6.25}$$

Then the $j$-th score function can be calculated as

$$\begin{aligned} \psi_j &= \frac{1}{K} \frac{\partial}{\partial c_j} \int \left[ \sum_{h=0}^{m} c_h^T \phi(t) D^h y_i(t) \right]^2 \\ &= \frac{2}{K} \int \left[ \sum_{h=0}^{m} c_h^T \phi(t) D^h y_i(t) D^j y_i(t) \right] \phi(t) dt, \\ & \quad j = 1, \ldots, m-1. \end{aligned} \tag{6.26}$$

The integral involved in this expression can be evaluated numerically (e.g., using the trapezoidal rule over a fine mesh of equally-spaced values of $t$).

### 6.3.3 Splitting

In this step the fitted model is split with respect to the variable $Z_{l*}$ into a segmented model with $B$ segments. For a fixed number of splits (e.g., we choose 2), two rival segmentations can be compared easily by comparing the segmented objective function $\sum_{b=1}^{B} \sum_{i \in I_b} \Psi(X_i, \omega_b)$. The optimal partition is found by performing an exhaustive search over all conceivable partitions with $B$ segments. See [109] for more details.

## 6.4 Comparison of Different Tree Models for Online Auction Dynamics

We now apply the various tree methods to the data from Chapter 3.

First, we apply the multivariate tree from Section 6.2.2. We first sample the step functions of the live bids (see Chapter 3 for description of live bid reconstruction) on a fine mesh (with 0.01 intervals) and obtain 190 vectors of length 71. A multivariate regression tree is fitted to these vectors and pruned to seven nodes. Figure 6.1 displays the fitted tree after pruning. Figure 6.2 shows the estimated price curve at each terminal node. Two variables, the opening price and the winning price are recruited into the tree splitting procedure. We see that the estimated price curves for leaf nodes L4, L5 and L7 are very similar; the main difference is their magnitude. Similarly, price curves for leaf nodes L1 and L6 are similar, except for differences in magnitude. The multivariate regression tree thus partitions our auctions into price paths of 3 or 4 different shapes where each shape can differ in magnitude. The magnitude and shape of each price curve is determined by the opening bid.

Next we implement the spline tree from Section 6.2.3. We first recover the functional objects using B-splines of order 6. A tree is then fit to the $190 \times 18$ matrix of estimated spline coefficients and pruned to seven nodes. Figure 6.3 displays the resulting tree. The results appear very similar to the multivariate tree. One of the main differences is the estimated price function for the 5th leaf node L5 which differs significantly from the multivariate tree in magnitude and shape. But other

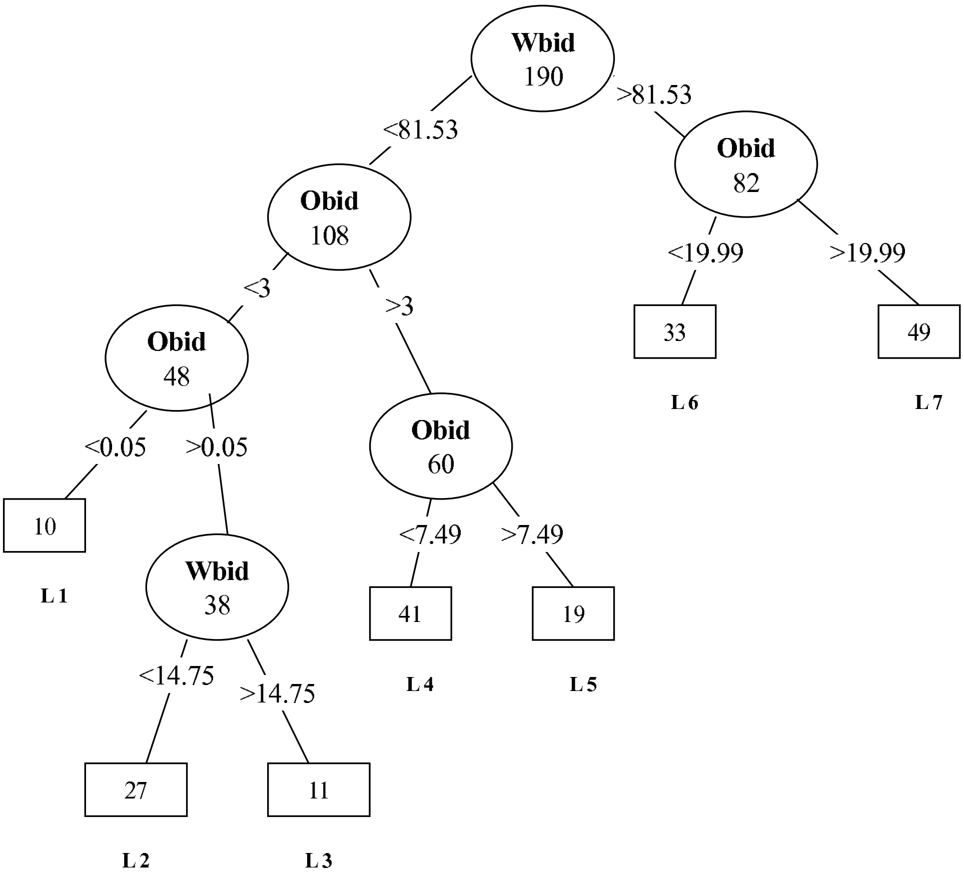


Figure 6.1: The fitted multivariate regression tree. The number in each node is the sample size. "$L1, \cdots, L7''$ denotes the 7 terminal/leaf nodes.

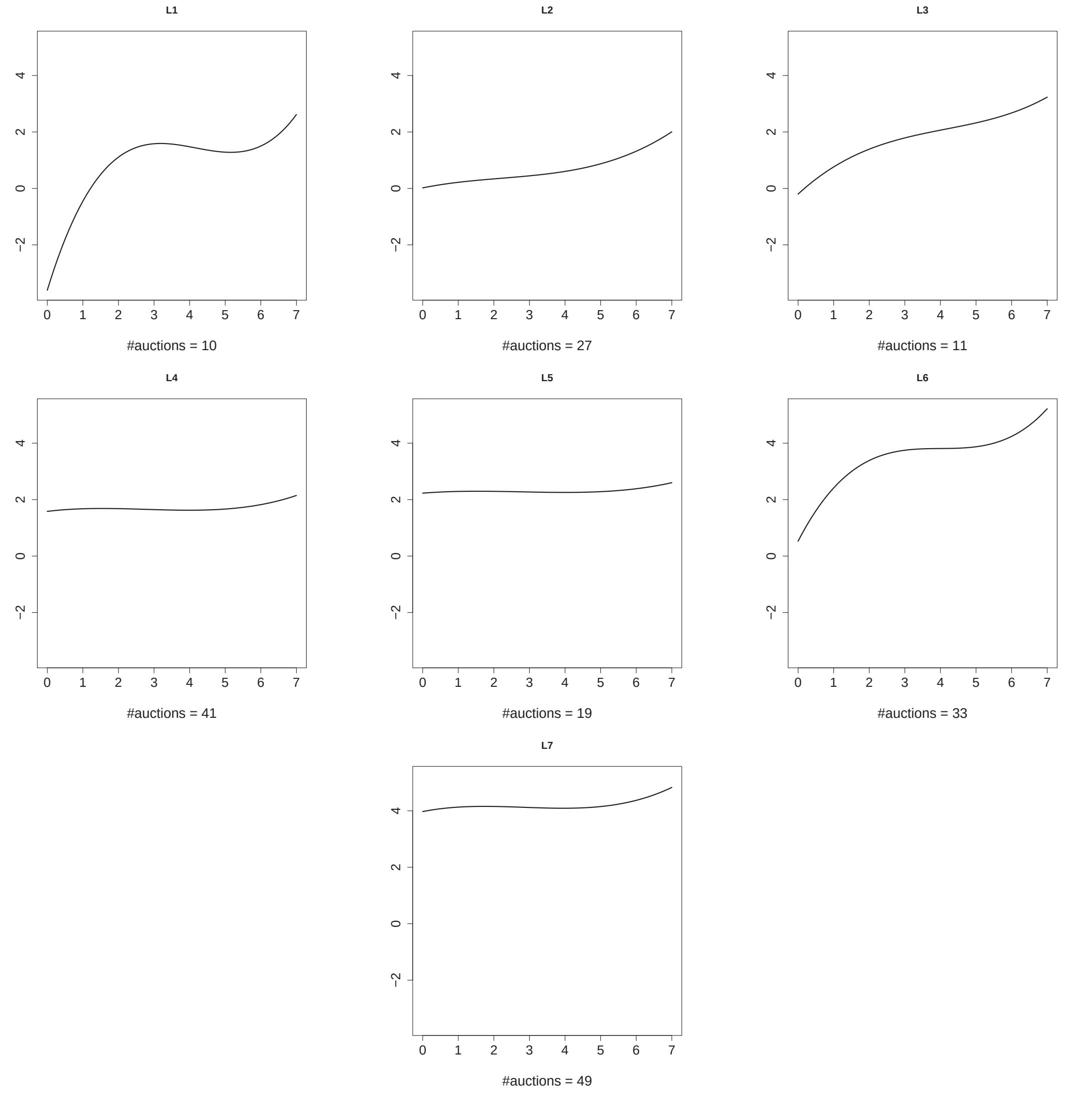


Figure 6.2: Mean price curve for each leaf node of the fitted multivariate regression tree shown in Figure 6.1. The panels from left to right and top to bottom correspond to the terminal nodes reading from left to right in Figure 6.1. The $x$-axes represent time of auctions and the $y$-axes represent amounts of prices on log scale.)

than that, it appears that the spline tree, which is based on a smooth representation of the auction's price path, does not differ much from the multivariate tree for which no smoothing was used.

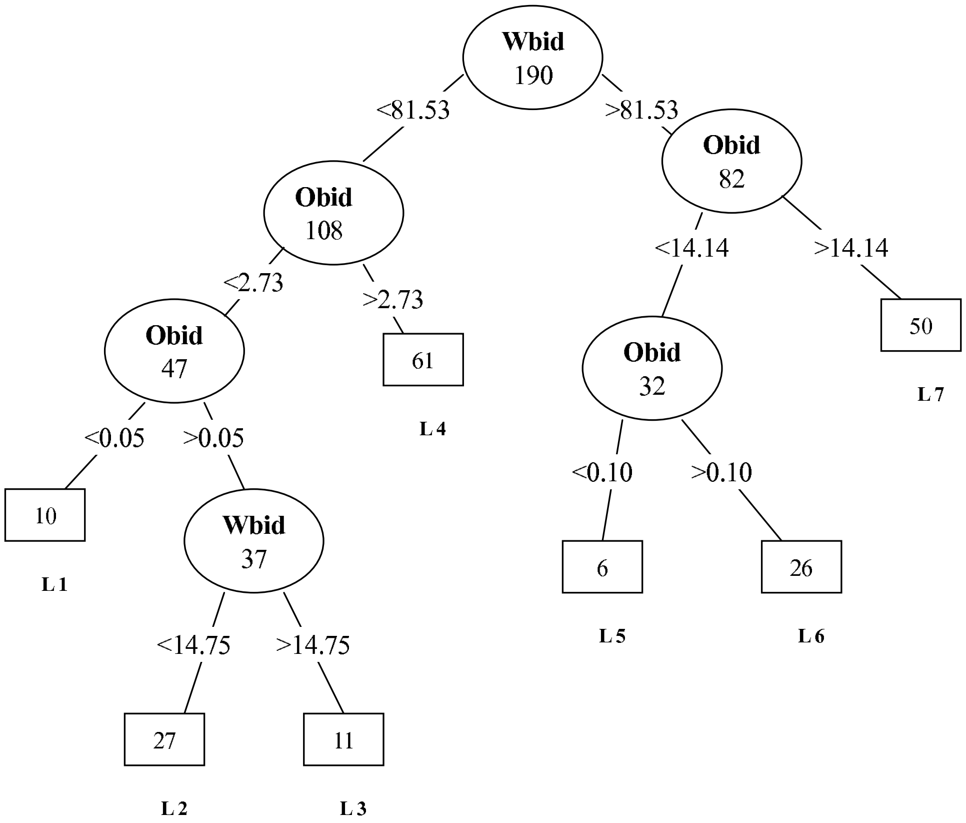


Figure 6.3: The fitted spline tree. The number in each node is the sample size. "$L1, \cdots, L7''$ denotes the 7 terminal/leaf nodes.

For comparison, we also investigate the principal component tree described in Section 6.2.3. Figure 6.5 gives the first two principal component loadings of the auction prices for our data. The first component contrasts prices during the duration of an auction for three typical phases: the beginning phase, the middle phase and the closing phase. This component alone explains 82.79% of the total variance. The first two principal components together explain 93.48% of the total variance. A

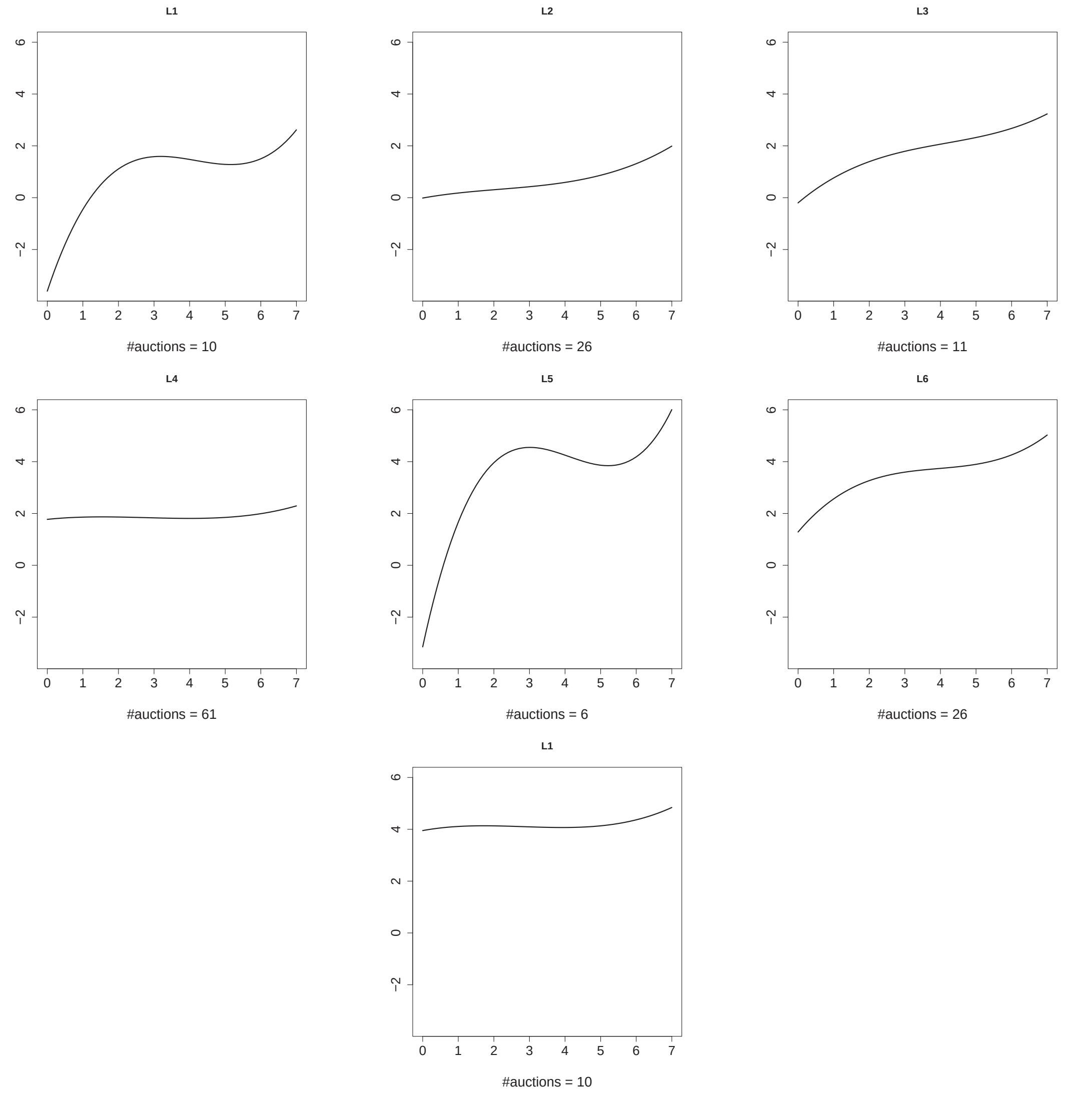


Figure 6.4: Mean price curve for each leaf node of the fitted spline tree shown in Figure 6.3. The panels from left to right and top to bottom correspond to the terminal nodes reading from left to right in Figure 6.3. The $x$-axes represent time of auctions and the $y$-axes represent amounts of prices on log scale.

multivariate tree is fit to the first two principal component scores and pruned to eight leaf nodes. The resulting tree is shown in Figures 6.6 and 6.7. We can see that interestingly, the distribution of the shapes of the price curves has changed: almost all price shapes are now marked by little or no early activity and little late activity (leaf nodes L2, L4, L5, L7, L8). The regression tree identifies one shape of gradual price increase (leaf node L3) and one shape of intense early activity as well as moderate late activity (leaf node L1 and L6).

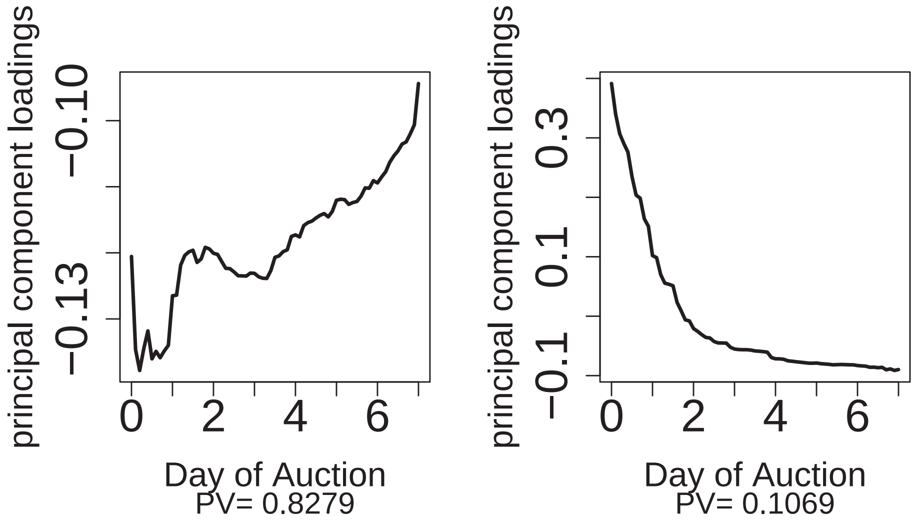


Figure 6.5: The first two principal component loadings of the auction prices are displayed from left to right panels. PV indicates the amount of total variation accounted for by each principal component.

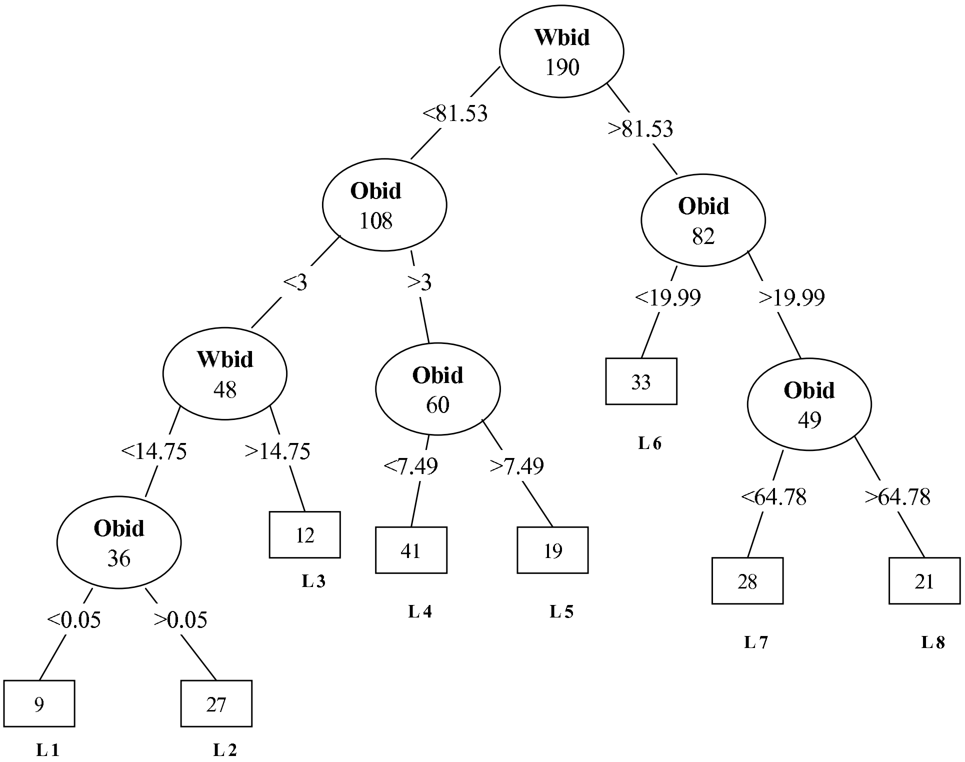


Figure 6.6: The fitted principal component tree. The number in each node is the sample size. "$L1, \cdots, L8''$ denotes the 8 terminal/leaf nodes.

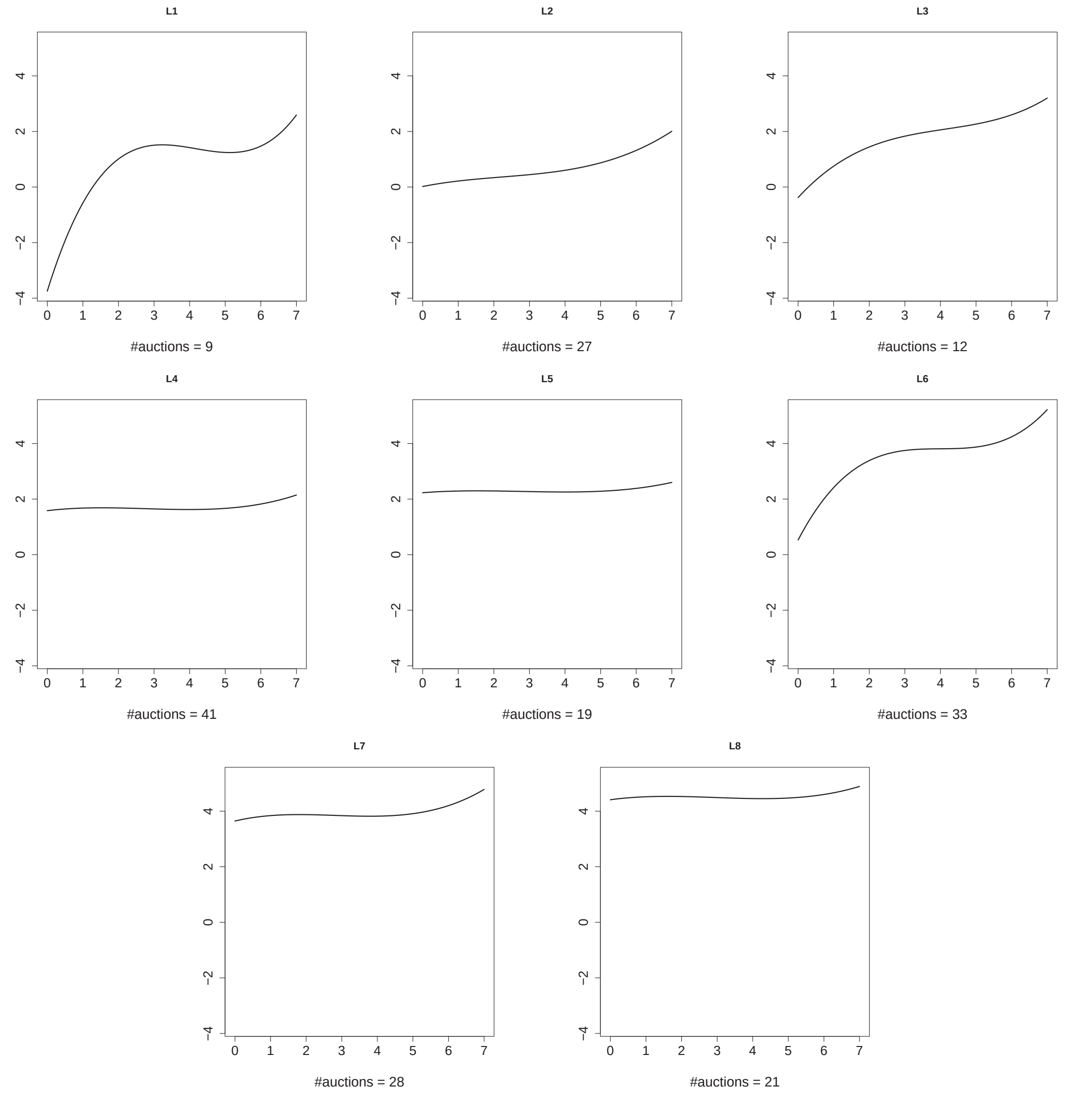


Figure 6.7: Mean price curve for each leaf node of the fitted principal component tree shown in Figure 6.6. The panels from left to right and top to bottom correspond to the terminal nodes reading from left to right in Figure 6.6. The $x$-axes represent time of auctions and the $y$-axes represent amounts of prices on log scale.

Lastly, we fit the model-based functional differential equation tree proposed in Section 6.3 to our data. The advantages of using this are: (1) The objective function used for parameter estimation is also used for partitioning; (2) The recursive partitioning allows for modeling of non-linear relationships and automated detection of interactions among the explanatory variables; (3) The use of differential equation models provides us with a segmented model that we can analyze and interpret by sub-populations; (4) In contrast to the trees introduced previously, now we also model the relationship between the dynamics.

Figure 6.8 gives the fitted functional differential equation tree (FDET), pruned to eight leaf nodes. We see that it adds more explanatory power by introducing a new splitting variable (the number of bids). We note first that the estimated price curves now cover a wider range of different shapes (shape 1: fast initial increase followed by a slow-down and then a late spurt; shape 2: little initial activity and moderate late spurt; shape 3: almost linear increase; and more). We also note that these shapes are separated into three groups by the covariate opening price. For auctions with opening price lower than \$0.99, most of the estimated price curves follow shape 1 (leaf nodes L1 and L3); for auctions with opening price between \$0.99 and \$4.99, the estimated price curves follow shape 2 (leaf nodes L4 and L5); for those with opening price higher than \$4.99, whether the estimated price curves follow shape 2 or shape 3 is jointly determined both by opening price and number of bids. For an auction opening at a price higher than \$4.99, if the number of bids is less than 23, its estimated price curve resembles shape 2 (leaf nodes L6 and L7). As does an auction that opens at a price higher than \$4.99 with number of bids more than 23, its estimated price curve looks like shape 3, a flat straight line (leaf node L8). This indicates to us that the level of opening price plays a very important role in partitioning the auction dynamics. In most cases, a low opening price easily attracts people to place their bids in the auction, thus leading to large amount of bidding activity at early stage of the auction (early bidding), which causes sharp increase in price velocity and therefore, in price curve. Since much of the initial bidding energy is used up in this case, there is few bidding energy left for the late stage of the beginning of the auction, thus resulting a short period of slow-down in both price velocity and price curve during the middle of the auction. However, low level of opening price easily regenerates bidding energy and therefore leads to a huge spurt towards the end of the auction. In contrast, for auctions with high opening prices, the auction dynamics are interactively affected by number of bids and opening price. In general, a high opening price suppresses bidding activity initially, while with the progress on the auction, the impact of the opening price on the dynamics fades away as more bids are place, resulting a spurt in the price velocity and consequently the price curve (this leads to the cases shown in L6 and L7). Interestingly, too many bids placed over the course of an auction with high opening price might not lead late increase (leaf node L8).The number of bids, which is the second covariate that enters the tree, further separates the shapes of estimated price curves into finer groups, when the opening price is high. This suggests that the number of bids plays an important role in categorizing the price dynamics conditional at the high level of opening price.

We investigate the predictive performance of the functional differential equa-

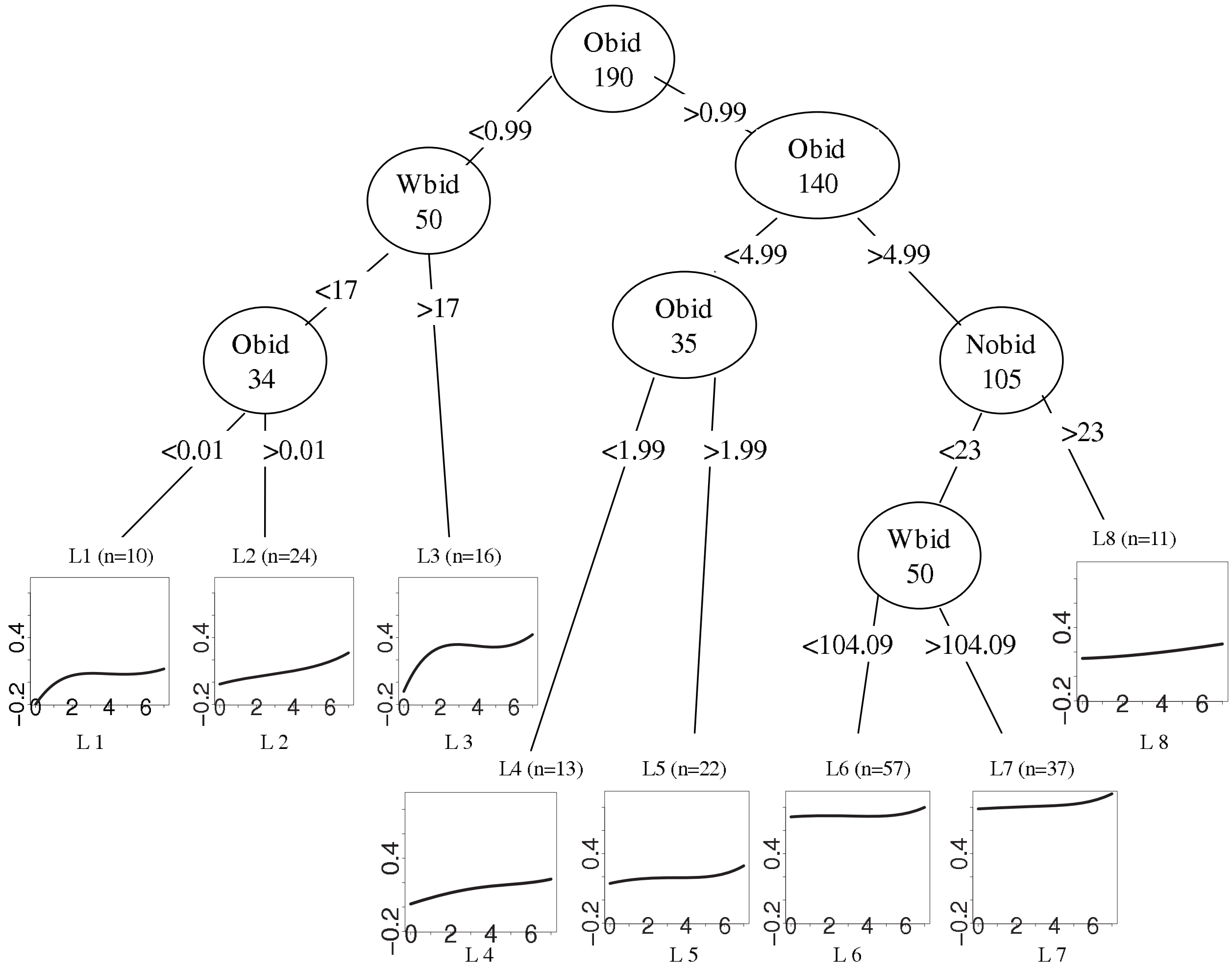


Figure 6.8: The fitted model-based differential equation tree. The number in each node is the sample size. “$L1, \cdots, L8''$ denotes the 8 terminal/leaf nodes. The panels associated with the leaf nodes are the mean price curves for corresponding leaf nodes of the fitted differential equation tree. In these panels, the $x$-axes represent time of auctions and the $y$-axes represent amounts of prices on log scale.

tion tree on a holdout test sample of size 60 auctions (≈ 30% of the data set). A tree is trained on a learning sample of 130 auctions (≈ 70% of the data set), and the resulting rule is used to predict the price curves of the 60 auctions in the test sample based on their auction-related characteristic information. We also measure forecast accuracy on the test sample using the mean-absolute-percentage-error (MAPE). The result is shown in Figure 6.9. MAPE is less than 5% for almost the entire auction period, except at the very start of the auction. This exception may be attributed to an instability which often appears at the auction start and noise caused by such instability. On the other hand, spline instability may be another reason for this exception.

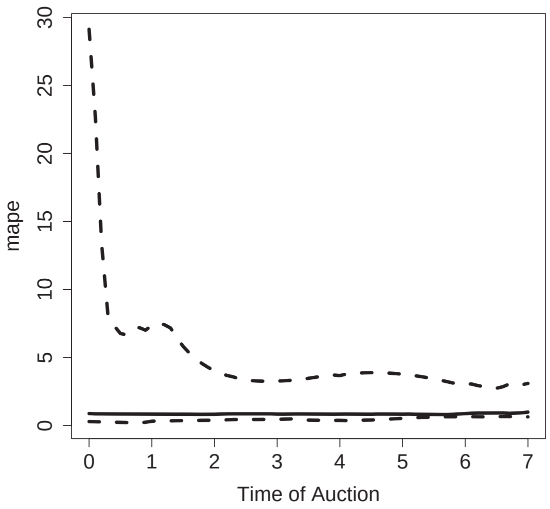


Figure 6.9: Mean Absolute Percentage Errors (MAPEs). MAPE is the error between the forecasted price curve and the true functional price curve; The dotted lines correspond to the 5th and 95th percentiles.

## 6.5 Conclusions

In this Chapter, we give a brief overview of extant tree models, from a univariate response, through multivariate response, to a functional response. A brief review is also given of model-based recursive partitioning methods. Based on these tree methods, we propose a functional-tree framework that is based on differential equation models. We compare different tree models by applying them to online auction dynamics. We show that our functional differential equation tree model generates a well balanced tree which is more interpretable. This work is novel in the sense that it incorporates dynamics into trees of functional data. The extant methods either only deal with discrete observations, or only embed regular parametric models such as linear regression models or maximum likelihood into trees. While our method fits trees to functional data, and incorporates dynamics into the tree by embedding differential equation models into the tree. This allows for the efficient extraction of meaningful subgroups of functional data based on their dynamics.

# Chapter 7

## Conclusions and Future Research

This dissertation studies the price dynamics within and across online auctions using a modern set of statistical analysis tools called *Functional Data Analysis*. As a practical case study, exploratory analyses, statistical modeling and statistical inference were performed for 190 7-day auctions on Xbox gaming systems and Harry Potter and the Half-Blood Prince books on eBay.com.

This research was divided into three phases. In phase I, we used the functional context to systematically describe the empirical regularities of auction dynamics. A new *dynamic forecasting system* was developed to predict the price of an ongoing auction. This model allows dynamic forecasting of an ongoing auction. We apply our forecasting system to real data from eBay on a diverse set of auctions and find that the combination of static and time-varying information creates a powerful forecasting system. The model produces forecasts with low errors, accomodates the changing price-dynamics well, and outperforms standard forecasting methods like double exponential smoothing which severely under-predicts the price-evolution. We conducted a sensitivity analysis over the forecasting accuracy to different choices of the knots and smoothing parameter, and found that the magnitude of the MAPE values, the measurement of the forecasting accuracy, change very little in the standard errors.

In phase II, we used *differential equations models* to capture the dynamics in online auctions. As a preliminary step towards arriving at a suitable differential equation model, we first performed exploratory analysis based on Phase Plane Plots of price-dynamics. We showed that a second-order linear differential equation well-approximates the three-phase dynamics that take place during an eBay auction. We then use a novel multiple-comparison test to compare the dynamics models of sub-populations of auctions, where the grouping is based on characteristics of the auction, the item, the seller, and the bidders.

In phase III, to better incorporate the different characteristics of the auction, item, bidders and seller information into the differential equation, we extended the model-based recursive partitioning methods developed by [109] to the functional context and proposed *Model-based Functional Differential Equation Trees.* We compared this new tree-method with trees either based on high-dimensional multivariate responses or functional responses.

## 7.1 Contributions

The contributions of the research in this dissertation are summarized as follows:

- Systematic investigation of the empirical regularities of auction dynamics using a functional regression context.
- Development of a new dynamic forecasting system for the price of an ongoing auction.

- Identification and characterization of second-order linear differential equation for modeling price dynamics of online auctions based on Phase Plane Plots and simulations.

- Introduction of a new multiple comparison test for the dynamics heterogeneity of functional data objects across different sub-populations, which captures both global and local differences between objects.

- Extension of model-based recursive partitioning methods to functional data objects based on differential equation models.

## 7.2 Future Work

Modeling online auctions using functional data analysis is still in the developing stage. The dynamic forecasting model, differential equation model and functional differential equation tree model developed in this work are only implemented on auctions of the same duration. But the lessons learned form it can be used to extend the models to auctions of different length. Modeling auctions of different durations is challenging since it involves registration of misaligned curves (see e.g. [86] or [48]). However, in the auction context the misaligned curves are of different length which poses additional difficulties.

Another extension is to incorporate a concurrency component. In online auctions, bidders have the option to inspect and follow multiple auctions at the same time. This places new challenges for modeling, especially in the functional framework. Some solutions via visualization of concurrent functional objects and modeling of concurrent final prices are proposed in a related series of papers [52, 41]. Finally, further research is required to better understand the exact role of price dynamics and their impact on economic theory. One possible avenue is the exploration of functional differential equation models in the auction context [50].

In this dissertation, we have tried to understand and explain the price dynamics of online auctions using different FDA techniques. But more work still needs to be done to better understand the exact role of price dynamics and their impact on economic theory, therefore deriving a helpful platform for auction participants.

The models developed in this dissertation are general for many other functional contexts. They can be easily adapted to functional data in the same fashion of online auctions, although modeling and forecasting performances and results will likely be application-dependent. We are interested in the future in investigating how our models can be extended to those counterparts.

Finally, an enhancement world will be functional bivariate modeling. There has been substantial research studying the relationship between a set of predictor variables (e.g., the opening bid, the seller's rating, condition of the item, ect.) and a single response variable (e.g., the price) in online auctions via univariate regression modeling. While there are interactions between the price process and the bidder process, simple univariate modeling of the price process can cause information loss. A bivariate functional model for online auctions is required, where both price and bid time are modeled bivariatly. This is challenging because fitting a bivariate functional model requires high dimensional smoothing, carefully checking of the model, and thus new ways of checking model assumptions, residuals, etc. An integrated way of fitting and model checking for high dimensional functional data via a series of classical statistical techniques could potentially fulfill this purpose.

# Appendix A

# Web Crawler

A web crawler is a program or automated script which browses the internet in a methodical, automated manner. Many online services use them to create a copy of all the visited pages for later processing. For instance, search engines index the downloaded pages to provide faster searches. In general, crawlers start with a list of URLs to visit, called the seeds. And the list of URLs will be recursively visited based on a certain schedule.

In our context, we focus on the bid information of one specific item. Thus, we will be crawling a URL instead of a list. Since it is often difficult to retrace the bid history after the end of the auction, we have to make our crawler recursively visit the destination before the auction ends. Of course, it is good practice to keep the frequency as low as possible in order to avoid overloading the opposite server. With this in the back of our minds and based on a basic crawling package created by Dr. Gove N. Allen (see [2] and http://www.gove.net for details.), we wrote two short programs. Sample scripts used for our eBay data collection are provided in the following Sections. The scripts collect eBay bid information and bid histories during 2005. There are two types of outputs: “bid information” and “bid history.” “Bid information” (e.g., item id, start time, num of bids, start price, currency unit, ship price, seller, rating, reserve status, item condition) for all collected items will be put into one file, while “bid history” for each item is written in another individual separate file. Generally, most efforts are being put on the string retrieve and comparison. The script “secretagent.txt” contains the functions used in “ebay.crawler” (see appendix A.1). Some of these functions are basic crawler functions used to get the contents of a webpage. Others are utility functions used to help us identify information on the webpage. And some IO functions are used to write the “bid information” and “bid history” into separate formatted files like excel files. Finally, “ebay.crawler” is a sample main function which controls the entire process. See Appendix A.1 for detailed descriptions of these functions.

## A.1 SECRET Agent

The script “secretagent.txt” contains the functions used in “ebay.crawler” (see appendix A.2). Some of these functions are basic crawler functions used to get the contents of a webpage. Others are utility functions used to help us identify information on the webpage. And some IO functions are used to write the “bid information” and “bid history” into separate formatted files like excel files. This script was written based on a basic crawling package created by Dr. Gove N. Allen (see [2] and http://www.gove.net for details.), modifications were made to accommodate our needs in collecting online auctions data from eBay.

### General subprocedures:

```
sub print(TheData)
        wscript.echo theData
end sub
```

```
function file_exists(filename)
        # Create the File System Object
        Set temp_fso = CreateObject(“Scripting.FileSystemObject”)
         if temp_fso.FileExists(filename) then
        file_exists = true
        else
        file_exists = false
        end if
end function

#count the the number of occuerance of key_str in the given string from start_str to end_str in doc
function count_btw(doc, key_str, start_str, end_str)
        temp_pos = doc.pos
        doc.moveto(start_str)
        sub_str = doc.gettext(end_str)
        start = 1
        count = 0
        do
        count = count + 1
        sp = instr(start, sub_str, key_str)
        if sp = 0 then
        count = count -1
         end if
        start = sp + len(key_str)
        loop while sp <> 0 and start <len(sub_str)
        doc.pos = temp_pos
        count_btw = count
end function

#count the the number of occuerance of key_str in the given string from current to end_str in doc
function count_till(doc, key_str, end_str)
        temp_pos = doc.pos
        sub_str = doc.gettext(end_str)
        start = 1
        count = 0
        do
        count = count + 1
        sp = instr(start, sub_str, key_str)
        if sp = 0 then
        count = count -1
        end if
        start = sp + len(key_str)
        loop while sp <>0 and start < len(sub_str)
        doc.pos = temp_pos
        count_till = count
end function

function stripWhiteSpace(theData)
        dim retval, onechar, x
        retval = ””
        for x=1 to len(theData)
        onechar=mid(theData, x, 1)
        if asc(onechar) = chr(9) or asc(onechar) = chr(13) then
        retval=retval & ” ”
        elseif asc(onechar) > 31 then
        retval=retval & onechar
        end if
        next
        do while instr(1, retval, ” ”)>0
        retval=replace(retval,” ”,” ”)
        loop
        stripWhiteSpace = trim(retval)
end function

function stripTags(theData)
        dim retval, dataon, onechar, x
        retval = ””
        dataon = true
```

```
          for x=1 to len(theData)
          onechar=mid(theData, x, 1)
          if onechar = "<" then
          dataon=false
          elseif onechar=">" then
          dataon=true
          elseif dataon then
          retval=retval & onechar
          end if
          next
          stripTags = retval
end function

function symbol_trans(str)
          newstr = replace(str,"&amp;","&")
          symbol_trans = newstr
end function

function time_adj(time1)
           temp1 = left(time1, 9)
          temp2 = right(time1, 12)
          time_adj = temp1&" "&temp2
end function

function ship_adj(price)
          temp1 = left(price, len(price)-1 )
          ship_adj = temp1
end function

function price_unit(temp_str)
          if left(temp_str,1) = "G" then
          left_temp = "GBP"
          elseif left(temp_str,1) = "U" then
          left_temp = "US"
```

```
        elseif left(temp_str,1) = "E" then
        left_temp = "EUR"
        end if
        price_unit = left_temp
end function

function price_retr(temp_str)
        #print "str"&temp_str
        right_temp = right(temp_str, len(temp_str)-4)
        price_retr = right_temp
end function

function gettext_notag(sa, endchar)
        nextstr = sa.getText(endchar)
        nextstr = ltrim(nextstr)
        if len(nextstr) < 1 then
        sa.moveto ">"
        nextstr = sa.getText(endchar)
        end if
        gettext_notag = symbol_trans(nextstr)
end function

———-functions for retriving bid information ———-
function retr_item_num(subdoc)
        subdoc.moveto "<title>"
        subdoc.moveto "item "
        retr_item_num= subdoc.gettext_next(10)
end function

function retr_start_time(subdoc)
        subdoc.moveto "Current bid:"
        subdoc.moveto "Start time:"
        subdoc.moveto "< td >"
        start_time= gettext_notag(subdoc,"<")
```

```
    retr_start_time = time_adj(start_time)
    'print "Start time: " & start_time
end function

function retr_hist_addr(subdoc)
    subdoc.moveto "History:"
    subdoc.moveto "<a href=""""
    retr_hist_addr = gettext_notag(subdoc, """")
end function

function retr_ship_price(subdoc)
    temp1 = subdoc.pos
    subdoc.moveto "Shipping costs:"
    temp2 = subdoc.pos
    'the agent skips the situation without shipping cost
    if (temp2 - temp1 = 0) or (temp2- temp1 >2000) then
    ship_price = "N/A"
    subdoc.pos = temp1
    else
    subdoc.moveto "></td>US $"
    ship_mv = subdoc.pos - temp2
    if ship_mv > 0 then
    ship_price= gettext_notag(subdoc,"-")
    ship_price = ship_adj(ship_price)
    else
    ship_price="N/A"
    end if
    end if
    retr_ship_price = ship_price
end function

function retr_seller(subdoc)
     subdoc.moveto "Seller information"
    subdoc.moveto "<tr>"
    subdoc.moveto "<td"
    subdoc.moveto "<td"
    subdoc.moveto "<a href"
    subdoc.moveto ">"
    seller= gettext_notag(subdoc,"<")
    'print "Seller: " & seller
    retr_seller = seller
end function

function retr_rating(subdoc)
    subdoc.moveto "<a href"
    subdoc.moveto ">"
    'print "temp = "&gettext_notag(subdoc,"< /a >")
    retr_rating= gettext_notag(subdoc,"< /a >")
    'print "Rating: " & rating
end function

——— functions for retrieving bid history ———
function retr_user(doc)
    doc.moveto "<td"
    doc.moveto "<td"
    if count_till(doc, "<a href", "<td") > 0 then
    doc.moveto "<a href"
    doc.moveto ">"
    retr_user = gettext_notag(doc, "<")
    else
    retr_user = "Private"
    end if
end function

function retr_user_rating(doc)
    temp_pos = doc.pos
    doc.moveto "<a href"
    doc.moveto ">"
    retr_user_rating = gettext_notag(doc,”<”)
    if not isnumeric(retr_user_rating) then
    retr_user_rating = ”N/A”
    doc.pos = temp_pos
    end if
end function


function retr_hist_price(doc)
    doc.moveto ”<td”
    doc.moveto ”>”
    retr_hist_price = gettext_notag(doc,”<”)
end function


function retr_bid_time(doc)
    doc.moveto ”<td”
    doc.moveto ”<img”
    doc.moveto ”>”
    retr_bid_time = gettext_notag(doc,”<”)
    retr_bid_time = time_adj(retr_bid_time)
end function


——— function for updating exist old record ———
function update(filename, doc, v, num)
    Dim xlApp, xlBook, xlSht
    Set xlApp = CreateObject(”Excel.Application”)
    set xlBook = xlApp.WorkBooks.Open(filename)
    set xlSht = xlApp.activesheet
     xlApp.DisplayAlerts = False


    ind = false
    update = false
    nw = 0
    end_line = 0
    do while xlSht.Cells(end_line+1,12) = ”old” or xlSht.Cells(end_line+1,12) = ”closed” or xlSht.Cells(end_line+1,12)
```

```
= ”updated” or xlSht.Cells(end_line+1,12) = ”status”
    if xlSht.Cells(end_line+1,12) = ”updated” then
    xlSht.Cells(end_line+1,12) = ”old”
    end if
    end_line = end_line + 1
    loop

    for i = 1 to end_line
    if ((trim(xlSht.Cells(i,1)) = trim(v(0))) and (trim(xlSht.Cells(i,3)) <> trim(v(2)))) then
    for j = 1 to 11
    xlSht.Cells(i, j) = v(j-1)
    next
    xlSht.Cells(i, 12) = ”updated”
    update = true
    elseif (trim(xlSht.Cells(i,1)) <> trim(v(0))) then
    nw = nw + 1
    end if
    next
    if nw = end_line then
    for j = 1 to 11
    xlSht.Cells(end_line+1, j) = v(j-1)
    next
    xlSht.Cells(end_line+1, 12) = ”updated”
    if end_line >num then
    xlSht.Cells(end_line+1-num, 12) = ”closed”
    end if
    update = true
    end if
    xlBook.Save
    xlBook.Close SaveChanges=True
    xlApp.Quit
    #always deallocate after use...
    set xlSht = Nothing
    Set xlBook = Nothing
    Set xlApp = Nothing
end function

——— Classes ———
#class definition for secretAgent
Class secretAgent
dim pos, formdata, text, url, http, from, useragent, fso, f
Sub create(agentName, studyURL, researcherEmail)
     pos=1
    from=researcherEmail
    useragent=agentName & ”(” & studyURL & ”)”
    set http=createObject(”MSXML2.ServerXMLHTTP.3.0”)
    'set http=createObject(”WinHttp.WinHttpRequest.5”)
    Set fso = CreateObject(”Scripting.FileSystemObject”)
End Sub
sub getdoc(theURL)
    url=theURL
    pos=1
    http.open ”POST”, theURL, False
    http.setRequestHeader ”user-Agent”, useragent
    http.setRequestHeader ”From”, from
    http.send ””
    text=http.responseText
end sub
sub getImage(theURL, filename)
    http.Open ”GET”, theURL, false
    http.Send()
    Set adodbStream = CreateObject(”ADODB.Stream”)
    adodbStream.Open
    adodbStream.Type = 1 'adTypeBinary
    adodbStream.Write http.responseBody
    adodbStream.SaveToFile filename, 2 'adSaveCreateOverWrite
    adodbStream.Close
    Set adodbStream = Nothing
```

```
        text="Image saved to: " & filename
end sub
sub savePage(filename)
        Set f = fso.OpenTextFile(filename, 2, True)
        f.write text
        f.Close
end sub
sub openFile(filename)
        pos=1
        set f = fso.OpenTextFile(filename, 1, false)
        text = f.ReadAll
        f.close
end sub
function MoveBackTo(FindText)
        If pos < 1 Then pos = 1
        pos = InStrRev(text, FindText,pos) + Len(FindText)
        If pos = Len(FindText) then
        MoveBackTo=false
        else
        MoveBackTo=true
        end if
End function
function moveTo(TheData)
        sp=instr(pos,text,TheData)
        if sp=0 then
        moveto=false
        else
        moveto=true
        pos = sp + len(theData)
        end if
end function
function getText(theData)
        sp=instr(pos,text,TheData)
        if sp = 0 then
        'str not found
        getText = ""
        else
        theLen=sp-pos
        getText=mid(text, pos, theLen )
        pos = sp + len(theData)
        end if
end function
function getText_next(length)
        getText_next=mid(text, pos, length )
        pos = pos + length
end function
sub CreatData(filename, theData)
        Set f = fso.OpenTextFile(filename, 2, true)'
        f.writeLine theData
        f.Close
end sub
sub recordData(filename, theData)
        Set f = fso.OpenTextFile(filename, 8, true)', -2)'True)
        f.writeLine theData
        'f.write theData
        f.Close
end sub
sub clearData(filename)
        if fso.FileExists (filename) Then fso.DeleteFile filename, true
end sub
Sub print()
        WScript.Echo text
End Sub
sub AddFormData(theName, theValue)
        if formData > "" then formData=FormData&"&"
        formData=FormData & theName & "=" & urlencode(theValue)
end sub
sub clearFormData()
```

```
        formData=””
end sub
sub postdoc(theURL)
        pos=1
        url=theURL
        http.open ”POST”, theURL, False
        http.setRequestHeader ”Content-Type”, ”application/x-www-form-urlencoded”
        http.setRequestHeader ”From”, from
        http.setRequestHeader ”User-Agent”, useragent
        http.send formdata
        text=http.responseText
end sub
Function URLEncode(sRawURL)
        Dim iLoop
        Dim sRtn
        Dim sTmp
        Const sValidChars =”1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz:/._-$()˜”
        If Len(sRawURL) > 0 Then
        ' Loop through each char
        For iLoop = 1 To Len(sRawURL)
        sTmp = Mid(sRawURL, iLoop, 1)
        If InStr(1, sValidChars, sTmp, vbBinaryCompare) = 0 Then
        ' If not ValidChar, convert to HEX and p
        ' refix with %
        sTmp = Hex(Asc(sTmp))
        If sTmp = ”20” Then
        sTmp = ”+”
        ElseIf Len(sTmp) = 1 Then
        sTmp = ”%0” & sTmp
        Else
        sTmp = ”%” & sTmp
        End If
        End If
        sRtn = sRtn & sTmp
```

```
          Next
          URLEncode = sRtn
          End If
End Function

 End Class
```

### Description of each utility function:

- "create" – initialize the 'secretAgent' by filling some user information into variables
- "getDoc" – given a website, getting text of it and saving it. In this function, we first set URL and start position (from where we start to retrieve content, usually we start from 1). Then some of the http's fields are defined, e.g. http method is set to POST. Finally, we get the text from the website.
- "getImage" – similar to getDoc, this function retrieve the image and save it to defined file. (I have not gotten a chance to use this function)
- "savePage" – save current content in 'text' to a specified file.
- "openFile" – inverse operation of "savePage" : open a file and put all its contents into text.
- "moveTo" – It makes use of basic VB script function "instr". Given a string, the "moveTo" function will move current position pointer to the first occurrence that string. Actually the pointer will point to the position follow the string.

- “getText” – Given a string, retrieve all text before the first occurrence of the string.
- “CreatData” – Open a file for writing data and start write data from the beginning If the file already exists, the contents are overwritten.
- “recordData” – Open a file and start writing at the end. Contents are not overwritten.

For more information of academic data collection in electronic environments, please see [2].

## A.2 ebay.crawler

Next, I wrote a sample script for retrieving the auctions on Xboxes on EBAY.COM. The main script names ebay.crawler.wsf with secretagent.ws. To run this code,

- In WinXp click Start→All Programs→ Accessories→Command Prompt to open terminal for input command line.
- In this terminal, change directory to where the scripts are.
- type: Wscript.exe ebay.crawler.wsf
- Finally, when executing the scripts, the top 3 hot books will prompted out and will be recorded into temp.txt file.

**General subprocedures:**

```
<job>
      <script language=“VBScript” src=“secretagent.ws”> ’can also use“JScript”
```

```
dim sa 'documentation variable
dim subdoc
dim hisdoc
 num = 6
Set sa = new secretAgent
Set subdoc = new secretAgent
set hisdoc = new secretAgent
sa.create “ECR Agent 1.0”, “http://www.smith.umd.edu/dit/statschallenges/”, “gshmueliumd.edu”
subdoc.create “ECR Agent 1.0”, “http://www.smith.umd.edu/dit/statschallenges/”, “gshmueliumd.edu”
hisdoc.create “ECR Agent 1.0”,          “http://www.smith.umd.edu/dit/statschallenges/”, “gshmueliumd.edu”
folder_name = “xbox”
page_addr = “http://product.ebay.com/Microsoft-Xbox-Game-console-black_W0QQfvcsZ1452QQsoprZ43557637”
if (not file_exists(“../results_”&folder_name&”/bid_info.xls”)) then
  sa.creatData ”../results_”&folder_name&”/bid_info.xls”, ”item_num” & chr(9) & ”start_time” & chr(9) & ”num_bids” & chr(9) & ”start_price” & chr(9) & ”unit” & chr(9) & ”ship_price” & chr(9) & ”seller” & chr(9) & ”rating” & chr(9) & ”reserve” & chr(9) & ”condition” & chr(9) & ”title” & chr(9) & ”status”'records the data we just collected
end if
sa.getDoc page_addr
sa.pos = 1
k = 0
sa.moveto ”dSI(”
startTime = timer()
do
sa.getDoc page_addr
sa.pos = 1
k = 0
sa.moveto ”dSI(”
if count_till(sa, ”Optimize your selling success”, ”About eBay”) <> 0 then
sa.moveto ”Optimize your selling success”
end if
do
k = k + 1
```

```
’The following loop skips the unwanted ”buynow” item
do
temp = sa.pos
sa.moveto ”compareLimitTest(this)”
temp2 = sa.pos
pt_move = temp2 - temp
’print ”move : ”&pt_move
sa.moveto ”<td”
sa.moveto ”<td”
sa.moveto ”<a href=”””
subaddr = gettext_notag(sa,””””)
sa.moveto ”>”
title = gettext_notag(sa,”<”)
’print ”Title: ” & title
sa.moveto ”<td”
sa.moveto ”>”
condition = rtrim(gettext_notag(sa,”<”))
’print ”Condition: ” & condition
’sa.moveto ”<td”
’sa.moveto ”<td”
’sa.moveto ”<td”
sa.moveto ”ebcBid””>”
bids = trim(gettext_notag(sa,”<”))
loop while (len(bids)<1) and pt_move > 0
if pt_move >0 then
’The agent goes to the page for current item
subdoc.getDoc subaddr
item_num = retr_item_num(subdoc)
’print ”item_num”&item_num
if count_btw(subdoc, ”Reserve”, ”Current bid:”, ”Time left:”) = 1 then
reserved = 1
else
reserved = 0
end if
```

```
start_time = retr_start_time(subdoc)
hist_addr = retr_hist_addr(subdoc)
subdoc.moveto ">"
num_bids= trim(gettext_notag(subdoc,"b"))
if (IsNumeric(num_bids)) then
if num_bids <> 0 then
'print "1bids"&num_bids
subdoc.moveto "("
temp_str= gettext_notag(subdoc,"s")
unit = price_unit(temp_str)
'print "unit : " & unit
start_price = price_retr(temp_str)
'print "Starting bid: " & unit &" "&start_price
end if
else
unit = "N/A"
'if isnumeric(cint(bids)) then
' num_bids = bids
'else
num_bids = "N/A"
'end if
start_price = "N/A"
end if
ship_price = retr_ship_price(subdoc)
seller = retr_seller(subdoc)
rating = retr_rating(subdoc)
v = array(item_num,start_time, num_bids,start_price,unit, ship_price,seller,rating,reserved,condition, title)
'print "item num"&v(0)
if ( len(trim(v(0))) = 10 )then
updated = update("D:/crawler/results_"&folder_name&"/bid_info.xls", sa, v, num)
else
updated = false
end if
```

```
if (isnumeric(num_bids) and updated ) then
if num_bids <> 0 then
'——— retrieve bid history ————
hisdoc.getDoc hist_addr
hisdoc.moveto "<b>User ID</b>"
hisdoc.moveto "</tr>"
hisdoc.creatdata "../results_"&folder_name&"/"&trim(item_num)&".xls", "item name"&"("&item_num&")"&"
: "&title
'total = cint(num_bids)
i = 0
do while count_btw(hisdoc, "<td", "<tr", "</tr>") < 4
hisdoc.moveto "</tr>"
loop
do while (count_till(hisdoc, "About eBay", "</tr>") = 0 )
i = i +1
hisdoc.moveto "<tr"
user_id = retr_user(hisdoc)
user_rating = retr_user_rating(hisdoc)
hist_price = retr_hist_price(hisdoc)
hist_unit = price_unit(hist_price)
hist_price = price_retr(hist_price)
bid_time = retr_bid_time(hisdoc)
hisdoc.moveto "</tr>"
hisdoc.recorddata "../results_"&folder_name&"/"&trim(item_num)&".xls", item_num&chr(9)
&user_id&chr(9)&user _rating&chr(9)&hist_price&chr(9)&hist_unit&chr(9)&bid_time

do while count_btw(hisdoc, "<td", "<tr", "</tr>") <4 and count_till(hisdoc, "About eBay", "</tr>") = 0
hisdoc.moveto "</tr>"
loop
loop
end if
end if
end if
'print item_num&" item(s) have been recorded!"
```

```
          loop while k < num
          endTime = Timer()
          intHours = Abs( (endTime-startTime)/3600)
          'intMinutes = Abs( (endTime-startTime)/60)
          'intSeconds = (endTime-startTime)
          loop while intHours < 24
          if (not file_exists("../results_"&folder_name&"/bid_history_info.xls")) then hisdoc.creatdata "..
/results_"&folder_name&"/bid_history_info.xls", "item #"&chr(9)&"user id"&chr(9)
&"user rating"&chr(9)&"bid price"&chr(9)&"price unit"&chr(9)&"bid time" end if
          temp = gen_hist_file("D:/crawler/results_"&folder_name&"/bid_info.xls", "D:/crawler/results_"
&folder_name&"/bid_history_info.xls","D:/crawler/results_"&folder_name&"̈)
          print " items' history have been combined into file 'bid_history_info.xls'"
          </script>
</job>
```

Based on observation, most of the time, there will be more than 1 item that are approaching to close. Thus, in this sample code, without missing any item, we set up a strategy by which we always collect on the top 6 items that are about to be closed. In the main loop, we also filter out the items marked as "buy-it-now." After that, we start to retrieve the bid information for each item and write these bid information into the excel file "bid-information.xls." For any one of these six auctions, if there arrives a new bid, the old bid information recorded for this particular auction so far will be updated accordingly. Meanwhile, if there is a incoming new bid, the script will dig into the item link to find the bid history for this item. The corresponding bid history will be recorded separately into another excel file "bid-history.xls." To have a diverse enough data set, we focus our collection on one high-valued product (e.g. Xbox gaming systems) and one low-valued product (e.g. Harry Potter books).

Remarks: It is necessary to check the format of the webpage.

# Appendix B

## Sensitivity Analysis for Penalized Spline Smoothing

In order to test the sensitivity of the penalized spline smoothing with respect to changes in the knot-allocation and with respect to the choice of $\lambda$, we check the robustness of the results of forecasting online auction price curves (see Chapter 4 for details). The choice of our smoothing parameters is governed by reasonable fit. Since there is wide range of choices that lead to reasonable curve approximations, we investigate the sensitivity of the forecasting accuracy to different choices of the knots and smoothing parameter $\lambda$. Table B.1 shows the forecasting accuracy in terms of $\text{MAPE}_1$ (between the forecasted price and the functional curve) and $\text{MAPE}_2$ (between the forecasted curve and the actual current auction price) for three different sets of knots. Similarly, Table B.2 shows the sensitivity to the choice of $\lambda$. In both cases we see that the magnitude of the MAPE values remains in the area of 10%-30%, with very little change in the standard errors.

# Appendix C

## Sensitivity Analysis for Regularized Basis Approach

Since the regularized basis approach is used in the work of Chapter 5 and Chapter 6, we base our sensitivity analysis on the contents of Chapter 5. We investigate the sensitivity of the model (the 2nd order linear differential equation model

Table B.1: Sensitivity analysis of knot selection based on different knot scenarios. $\Upsilon2$ is the one used in this paper.

| Set | Knots | $\text{MAPE}_1$ | | $\text{MAPE}_2$ | |
|---|---|---|---|---|---|
| | | Mean | Std.Err. | Mean | Std.Err. |
| $\Upsilon1$ | 0,1,2,3,4,5,6,6.25,6.5,6.75 , 6.8750,7 | 0.18 | 0.05 | 0.29 | 0.04 |
| $\Upsilon2$ | 0,1,2,3,4,5,6,6.25,6.5,6.75, 6.8125,6.8750,6.9375,7 | 0.12 | 0.02 | 0.23 | 0.02 |
| $\Upsilon3$ | 0,0.5,1,1.5,2,3,4,5,6,6.25,6.5, 6.75,6.8125,6.8750,6.9375,7 | 0.26 | 0.04 | 0.31 | 0.03 |

Table B.2: Sensitivity analysis of $\lambda$ selection (knots fixed to $\Upsilon2$).

| $\lambda$ | $\text{MAPE}_1$ | | $\text{MAPE}_2$ | |
|---|---|---|---|---|
| | Mean | Std.Err. | Mean | Std.Err. |
| 0.1 | 0.28 | 0.04 | 0.32 | 0.04 |
| 0.3 | 0.23 | 0.03 | 0.28 | 0.03 |
| 0.5 | 0.21 | 0.03 | 0.28 | 0.03 |
| 0.7 | 0.18 | 0.02 | 0.26 | 0.03 |
| 0.9 | 0.16 | 0.02 | 0.25 | 0.02 |
| 1 | 0.16 | 0.03 | 0.27 | 0.03 |
| 5 | 0.15 | 0.03 | 0.27 | 0.03 |
| 10 | 0.12 | 0.02 | 0.24 | 0.02 |
| 15 | 0.12 | 0.02 | 0.23 | 0.02 |
| 20 | 0.12 | 0.02 | 0.23 | 0.02 |
| 25 | 0.12 | 0.02 | 0.23 | 0.02 |
| 30 | 0.12 | 0.02 | 0.23 | 0.02 |
| 40 | 0.11 | 0.02 | 0.23 | 0.02 |
| 50 | 0.11 | 0.02 | 0.23 | 0.02 |

as described in Chapter 5) fit to different knot choices (Table C.1) and smoothing parameter choices (Table C.2). Figure C.1 shows the estimated coefficient curves $\omega^*$ and the model-fit measures (RSQ and FRATIO) for the different sets of knots. Similarly, Figure C.2 shows these measures for different choices of $\lambda$. We can see that while the model is a bit more sensitive to $\lambda$, the qualitative nature of the fit does not change by much for different knots or smoothing parameters.

Table C.1: Sensitivity to different sets of knots. $\Upsilon3$ is the set used in this paper.

| Set | Knots |
|---|---|
| $\Upsilon1$ | 0,1,2,3,4,5,6,7 |
| $\Upsilon2$ | 0,1,2,3,4,5,6,6.25,6.5,6.75 ,6.8750,7 |
| $\Upsilon3$ | 0,1,2,3,4,5,6,6.25,6.5,6.75,6.8125,6.8750,6.9375,7 |
| $\Upsilon4$ | 0,0.5,1,1.5,2,3,4,5,6,6.25,6.5,6.75,6.8125,6.8750,6.9375,7 |

Table C.2: Sensitivity to different values of $\lambda$ (with a common set of knots $\Upsilon3$).

|  | Smoothing parameters |
|---|---|
| $\lambda$ | 0.1,0.3,0.5,0.7,0.9,1,5,10,20,25,30,40,50 |

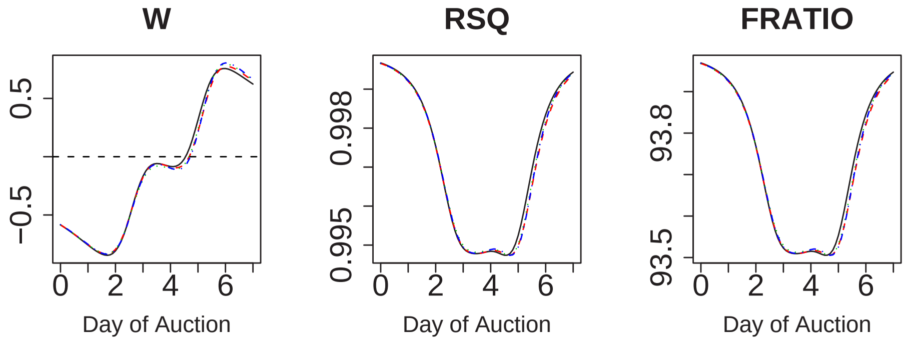


Figure C.1: Sensitivity of model fit to different knots.

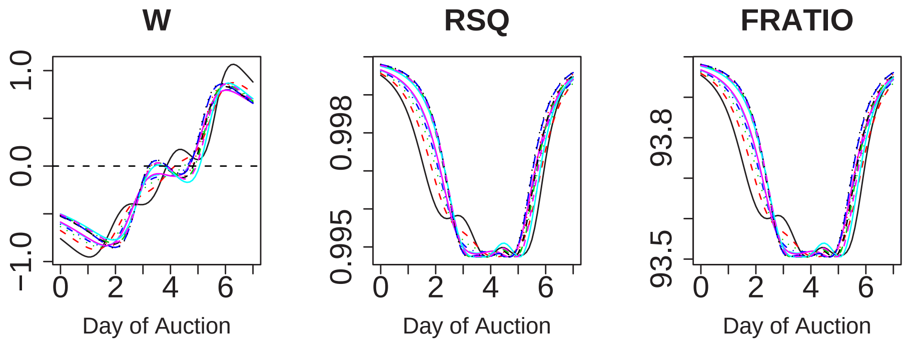


Figure C.2: Sensitivity of model fit to different smoothing parameters.

# BIBLIOGRAPHY

[1] C. Abraham, P. A. Cornillion, E. Matzner-Lober and N. Molinari, "Unsupervised curve-clustering using B-spline", *Scandinavian Journal of Statistics*, Vol.30, pp.581-595, 2003.

[2] G. N. Allen, D. L. Burk and G. B. Davis, "Academic data collection in electronic environments: Defining acceptable use of internet resources", *MIS Quarterly*, **30(3), 599-610, 12p, 1 chart** (AN 21940319), Sep 2006.

[3] R. Almgren, "Financial derivatives and partial differential equations", *American Mathematical Monthly*, 2002.

[4] R. Almgren, "Financial derivatives and partial differential equations", *American Mathematical Monthly*, 2002.

[5] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 2nd ed., New York: Wiley, 1984.

[6] D. W. K. Andrews, "Tests for parameter instability and structural change withunkown change point", *Econometrica*, **61, 821-856**, 1993.

[7] D. Ariely, A. E. Roth and A. Ockenfels, "An experimental analysis of ending rules in internet auctions", *The RAND Journal of Economics*, accepted, 2005.

[8] S. Ba and P. A. Pavlou , "Evidence of the effect of trustbuilding technology in electronic markets: Price premiums and buyer behavior", *MIS Quarterly*, **26, 269-289**, 2002.

[9] P. Bajari and A. Hortacsu, “The winner’s curse, reserve prices and endogenous entry: Empirical insights from eBay auctions”, *Rand Journal of Economics*, **3(2), 329-355**, 2003.

[10] P. Bajari anad A. Hortacsu, “Economic insights from internet auctions”, Nber working paper, No. ~w10076, 2004.

[11] R. Bapna, P. Goes, A. Gupta and Y. Jin, “User heterogeneity and its impact on electronic auction market design: An empirical exploration”, *MIS Quarterly*, **28(1)**, 2004.

[12] R. Bapna, P. Goes, and A. Gupta, “Analysis and design of business-to-consumer online auctions”, *Management Science*, **49, 85-101**, 2003.

[13] R. Bapna, W. Jank and G. Shmueli, “Consumer surplus in oline auctions”, Working paper, University of Connecticut, 2004.

[14] P. C. Besse, H. Cardot and D. B. Stephenson, “Autoregressive forecasting of some functional climatic variations”, *Scandinavian Journal of Statistics*, **27, 673-687**, 2000.

[15] S. Borle, P. Boatwright and J. B. Kadane, “The timing of bid placement and extent of multiple bidding: An empirical investigation using eBay online auctions”, *Statistical Science (Forthcoming)*, 2006.

[16] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Belmont, CA: Wadsworth, 1984.

[17] K. Y. Chan and W. Y. Loh, “LOTUS: An algorithm for building accurate and comprehensible logistic regression trees”, *Journal of Computational and Graphical Statistics*, **13(4), 826-852**, 2004.

[18] Y. Choi, H. Ahn and J. J. Chen, “Regression trees for analysis of count data with extra poisson variation”, *Computational Statistics & Data Analysis*, **49, 893-915**, 2005.

[19] E. A. Coddington and L. Levinson, *Theory of Ordinary Differential Equations*, McGraw-Hill, New York, 1955.

[20] A. Cuevas, M. Febrero and R. Fraiman, “Linear functional regression: The case of fixed design and functional response”, *The canadian Journal of Statistics*, **30, 285-300**, 2002.

[21] K. D. Daniel and D. Hirshleifer, “A theory of costly sequential bidding”, Technical report, Kellogg Graduate School of Management, Northwestern University, 1998.

[22] C. Dellarocas, “The digitization of word-of-mouth: Promise and challenges of online reputation mechanisms” *Management Science*, October, 2003.

[23] P. Diggle, “An approach to the analysis of repeated measures”, *Biometrics*, **44, 959-971**, 1988.

[24] R. F. Easley and R. Tenorio, “Jump bidding strategies in internet auctions”, *Management Science*, **50(10), 1407-1419**, October, 2004.

[25] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman&Hall, 1993.

[26] R. F. Engle and J. R. Russel, “Autoregressive conditional duration: A new model for irregularly spaced transaction data”, *Econometrica*, **66(5), 1127-1162**, 1998.

[27] M. Escabias, A. M. Aguilera and M. J. Valderrama, “Modeling environmental data by functional principal component logistic regression”, *Environmetrics*, **16(1), 95-107**, 2004.

[28] J. Fan anad S. K. Lin, “Test of significance when data are curves”, *Journal of the American Statistical Association*, **93, 1007-1021**, 1998.

[29] J. J. Faraway, “Regression analysis for a functional response”, *Technometrics*, **39, 254-261**, 1997.

[30] J. P. Fouque, G. Papanicolaou and K. R. Sircar, *Derivatives in Financial Markets with Stochastic Volatility*, Cambridge University Press, 2000.

[31] R. Fraiman and C. Muniz, “Trimmed means for functional data”, *Test*, **10, 419-440**, 2001.

[32] J. Gamma, “Functional trees”, *Machine Learning*, **55, 219-250**, 2004.

[33] R. Ghani and H. Simmons, “Predicting the end-price of online auctions”, *Proceedings of the International Workshop on Data Mining and Adaptive Modeling Methods for Economics and Management*, held in conjuction

with the *15th European Conference on Machine Learning (ECML/PKDDD)*, http://www.accenture.com/xdoc/en/services/technology/publications /priceprediction.pdf, 2004.

[34] W. Guo, "Inference in smoothing spline analysis of vairance", *Journal of the Royal Statistical Society, Series B*, **64, 887-898**, 2002.

[35] P. J. Green and B. W. Silverman, *Nonparametric Regression and Generalized Linear Models*, London: Chapman & Hall, 1994.

[36] P. Hall, D. S. Poskitt and B. Presnell, "A functional data-analytic approach to signal discrimination", *Technometrics*, **43, 1-9**, 2001.

[37] G. Z. He, H. G. Muller and J. I. Wang, "Functional canonical analysis for square integrable stochastic processes", *Journal of Multivariate Analysis*, **85, 54-77**, 2003.

[38] K. Hendricks and H. J. Paarsch, "A survey of recent empirical work concerning auctions", *The Canadian Journal of Economics*, **28(2), 403-426**, 1995.

[39] N. L. Hjort and A. Koning, "Tests for constancy of model parameters over time", *Nonparametric Statistics*, **14, 113-132**, 2002.

[40] A. Hortacsu and L. Cabral, "Dynamics of seller reputation: Theory and evidence from eBay", Working paper, University of Chicago, 2005.

[41] V. Hyde, W. Jank, and G. Shmueli, "Investigating concurrency in online auctions through visualization", *The American Statistician*, **60(3), 241-25**, 2006.

[42] M. Isaac, T. C. Salmon and A. Zillante, "A theory of jump bidding in ascending auctions", *Journal of Economic Behavior and Organization*, **62(1), 144-164**, 2007

[43] G. M. James, T. J. Hastie and C. A. Sugar, "Principal component models for sparse functional data", *Biometrika*, **87, 587-602**, 2000.

[44] G. M. James and T. J. Hastie, "Functional linear discriminant analysis for irregularly sampled curves", *Journal of the Royal Statistical Society, Series B, Methodological*, **63, 533-550**, 2001.

[45] G. M. James, "Generalized linear models with functional predictors", *Journal of the Royal Statistical Society, Series B*,**64, 411–432**, 2002.

[46] G. M. James,"Generalized linear models with functional predictors", *Journal of the Royal Statistical Society, Series B*, **64, 411-432**, 2002.

[47] G. M. James and C. A. Sugar, "Clustering sparsely sampled functional data", *Journal of the American Statistical Association*, **98, 397-408**, 2003.

[48] G. M. James, "Curve alignment by moments", Under review, http://www-rcf.usc.edu/∼gareth, 2004.

[49] G. M. James, and A. Sood, "Performing hypothesis tests on the shape of functional data", *Computational Statistics and Data Analysis*, **50, 1774-1792**, 2006.

[50] W. Jank and G. Shmueli, "Dynamic profiling of online auctions using curve clustering", Working paper, Robert H. Smith School of Business, University of Maryland, http://www.rhsmith.umd.edu/dit/wjank/AuctionProfiling.pdf, 2004.

[51] W. Jank and G. Shmueli, "Functional data analysis in electronic commerce research", *Statistical Science*, **21(2), 155-166**, 2005.

[52] W. Jank and G. Shmueli, "Modeling concurrency of events in online auctions via spatio-temporal semiparametric models", forthcoming in the *Journal of the Royal Statistical Society, Series C*, 2006.

[53] W. Jank , G. Shmueli, C. Plaisant, and B. Shneiderman, *Visualizing Functional Data with an Application to eBay's Online Auctions*, Forthcoming in Chen, Haerdle and Unwin (eds.) *Handbook on Computational Statistics on Data Visualization*, Springer Verlag, Heidelberg, 2006.

[54] I. Karatzas and S. E. Shreve, *Methods of Mathematical Finance*, Springer-Verlag, New York, Inc., 1998.

[55] V. Kargin and A. Onatski, "Curve forecasting by functional auto-regression", Discussion paper No.: 0405-18, Department of Economics, Columbia University, New York, http://www.columbia.edu/cu/economics/discpapr/DP0405-18.pdf, 2004.

[56] P. Klemperer, "Auction theory: A guide to the literature", *Journal of Economic Surveys*, **13(3), 227-286**, 1999.

[57] H. Kim and W. Y. Loh, “Classification trees with unbiased multiway splits”, *Journal of the American Statistical Association*, **96(454), 589-604)**, 2001.

[58] O. Koppius, *Information Architecture and Electronic Market Performance*, in: Ph.d. Thesis, Rsm / Erasmus University, Rotterdam, 2002.

[59] V. Krishna, *Auction Theory*, Academic Press, San Diego, 2002.

[60] X. Leng and H. G. Mueller, “Clasification using functional data analysis for temporal gene expression data”, *Bioinformatics*, **22, 68-76**, 2006.

[61] W. Liu, M. Jamshidian and Y. Zhang, “Multiple comparison of several linear regression models”, *Journal of the American Statistical Association*, **99, 395-404**, 2004.

[62] W. L. Loh, “Estimating covariance matrices”, *The Annals of Statistics*, **19, 283-296**, 1991.

[63] W. Y. Loh, “Regression trees with unbiased variable selection and interaction detection”, *Statistica Sinica*, **12, 361-386**, 2002.

[64] N. G. Mankiw, D. Romer, and D. N. Weil, “A contribution to the empirics of economic growth”, *Quarterly Journal of Economics*, **107(2), 407-437**, 1992.

[65] D. Meyer, F. Leisch and K. Hornik, “The support vector machine under test”, *Neurocomputing*, **55(1-2), 169-186**, 2003.

[66] P. R. Milgrom and R. J. Weber, “A theory of auctions and competitive bidding”, *Econometrica*, **50(5), 1089-1122**, 1982.

[67] S. Mithas and J. L. Jones, “Do auction parameters affect buyer surplus in e-auctions for procurement?”, *Production and Operations Management (Forthcoming)*, **1-33**, 2006.

[68] J. N. Morgan and J. A. Sonquist, “Problems in the analysis of survey data, and a proposal”, *Journal of the American Statistical Association*, **58(302), 415-434**, 1963.

[69] J. A. Nelder and D. Pregibon, “An extended quasi-likelihood function”, *Biometrika*, **74, 221-232**, 1987.

[70] F. A. Ocana, A. M. Aguilera and M. J. Valderrama, “Functional principal components analysis by choice of norm”, *Journal of Multivariate Analysis*, **71, 262-276**, 1999.

[71] A. Ockenfels and A. E. Roth, “Strategic late-bidding in continuous-time second-price internet auctions”, Working Paper, University of Magdeburg, 2001.

[72] C. Octavian, “Reserve prices in repeated multi-unit auctions: Theory and estimation”, *The Econometrics of Auctions*, Toulouse, France, May 12-13, 2006.

[73] R. T. Ogden, C. E. Miller, K. Takezawa and S. Ninomiya, “Functional regression in crop lodging assessment with digital images”, *Journal of Agricultural, Biological, and Environmental Statistics*, **7, 389-402**, 2002.

[74] R. M. Pfeiffer, E.B ura, A. Smith and J. L. Rutter, “Two approaches to mutation detection based on functional data”, *Statistics in Medicine*, **21, 3447-3464**, 2002.

[75] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publ., San Mateo, California, 1993.

[76] S. J. Ratcliffe, L. R. Leader and G. Z. Heller, "Functional data analysis with application to periodically stimulated foetal heart rate data I: Functional regression", *Statistics in Medicine*, **21, 1103-1114**, 2002.

[77] S. J. Ratcliffe, G. Z. Heller and L. R. Leader, "Functional data analysis with application to periodically stimulated foetal heart rate data II: Functional regression", *Statistics in Medicine*, **21, 1115-1127**, 2002.

[78] J. O. Ramsay, R. D. Bock and T. Gasser, "Comparison of height acceleration curves in the fels, zurich, and berkeley growth data", *Annals of Human Biology*, **22, 413-426**, 1995.

[79] J. O. Ramsay and K. Munhall, V. Gracco and D. Ostry, "Functional data analysis of lip motion", *Journal of Acoustical Society of America*, **99, 3718-3727**, 1996.

[80] J. O. Ramsay, "Principal differential analysis: Data reduction by differential operators", *Journal of the Royal Statistical Society, Series B*, **58, 495-508**.58, pp., 1996.

[81] J. O. Ramsay, "Estimating smooth monotone functions", *Journal of the Royal Statistical Society, Series B*, **60, 365-375**, 1998.

[82] J. O. Ramsay, "Differential equation models for statistical functions", *Canadian Journal of Statistics*, **28, 225-240**, 2000.

[83] J. O. Ramsay, "Function components of variation in handwriting", *Journal of the American Statistical Association*, **95, 09-15**, 2000.

[84] J. O. Ramsay and J. B. Ramsey, "Functional data analysis of the dynamics of the monthly index of nondurable goods production", *Journal of Econometrics*, **107, 327-344**, 2001.

[85] J. O. Ramsay and B. W. Silverman, *Applied Functional Data Analysis: Methods and Case Studies*, Springer-Verlag New York, Inc., 1st edition, 2002.

[86] J.O. Ramsay and B.W. Silverman, *Functional Data Analysis*, Springer-Verlag New York, 2nd edition, 2005.

[87] S. K. Reddy and M. Dass, "Modeling online auction dynamics of fina art using functional data analysis", *Statistical Science*, **21, 179-193**, 2006.

[88] D. Lucking-Reiley, D. Bryan, N. Prasad and D. Reeves, "Pennies from eBay: The determinants of price in online auctions", *Technical report, University of Arizona*,

http://www.vanderbilt.edu/econ/reiley/papers/PenniesFromEBay.pdf, 2000.

[89] N. Rossi, X. Wang and J. O. Ramsay, "Nonparametric item response function estimates with the EM algorithm", *Journal of Educational and Behavioral Statistics*, **27, 291-317**, 2002.

[90] A. E. Roth and A. Ockenfels, "Last-minute bidding and the rules for ending second-price auctions: Evidence from eBay and amazon on the internet", em *American Economic Review*, **92(4), 1093–1103**, 2002.

[91] M. R. Segal, “Tree-structured methods for longitudinal data”, *Journal of the American Statistical Association*, **87, 407-418**, 1992.

[92] M. R. Segal, “Representative curves for longitudinal data via regression trees”, *Journal of Computational and Graphical Statistics*, **3(2), 214-233**, 1994.

[93] G. Shmueli and W. Jank, *Modelling the Dynamics of Online Auctions: A Modern Statistical Approach*, Forthcoming in Kauffman and Tallon (Eds.), Economics Information Systems and Ecommerce Research II: Advanced Empirical Methods, part of *Advances in Management Information Systems Series*, M.E. Sharpe, Armonk, NY, 2005.

[94] W. Jank and G. Shmueli, *Studying Heterogeneity of Price Evolution in eBay Auctions via Functional Clustering*, Forthcoming in Adomavicius and Gupta (Eds.), *Handbook of Information Systems Series: Business Computing*, Elsevier, 2006.

[95] G. Shmueli and W. Jank, “Visualizing online auctions”, *Journal of Computational and Graphical Statistics*, **14(2), 299–319**, 2005.

[96] G. Shmueli, R. P. Russo and W. Jank, “The Barista: A model for bid arrivals in online auctions”, Working paper, Robert H. Smith School of Business, University of Maryland,

http://www.rhsmith.umd.edu/dit/wjank/BARRISTA_BidArrivals.pdf, 2005.

[97] G. Shmueli, W. Jank, A. Aris, C. Plaisant and B. Shneiderman, "Exploring auction databases through interactive visualization", *Decision Support Systems*, to appear, 2006.

[98] J. S. Simonoff, *Smoothing Methods in Statistics*, Springer-Verlag, New York, 1st edition, 1996.

[99] C. W. Smith, *Auctions: The Social Construction of Values.* New York: Free Press.

[100] R. M. Solow, "A contribution to the theory of economic growth", *Quarterly Journal of Economics*, **70(1), 65-94**, 1956.

[101] J. D. Spurrier, "Exact confidence bounds for all contrasts of three or more regression lines", *Journal of the American Statistical Association*, **94, 483-488**, 1999.

[102] X. Su, M. Wang and J. Fan, "Maximum likelihood regression trees", *Journal of Computational and Graphical Statistics*, **13, 586-598**, 2004.

[103] T. Tarpey and K. K. J. Kineteder, "Clustering functional data", *Journal of Classification*, **20, 93-114**, 2003.

[104] M. J. Valderrama, F. A. Ocana and A. M. Aguilera, "Forecasting PC-ARIMA models for functional data", http://www.quantlet.de/scripts/compstat2002_wh /paper/invited/F_valderrama.pdf, 2002.

[105] S. Wang, W. Jank and G. Shmueli, "Explaining and forecasting online auction prices and their dynamics using functional data analysis", forthcoming in *Journal of Business and Economic Statistics*, 2006.

[106] S.Wang, W. Jank, G. Shmmueli and P. J. Smith, "Modeling price dynamics in eBay auctions using principal differential analysis", submitted to *Journal of the American Statistical Association*, 2006.

[107] Y. Yu and D. Lambert, "Fitting trees to functional data, with an application to time-of-day patterns", *Journal of Computational and Graphical Statistics*, **8(4), 749-762**, 1999.

[108] A. Zeileis and K. Hornik, "Generalized M-fluctuation tests for parameter instability". Report 80, SFB "Adaptive Information Systems and Modelling in Economics and Management Science", 2003. URL `http://www.wu-wien.ac.at/am/reports.htm#80`.

[109] A. Zeileis, T. Hothorn and K. HOrnik, "Model-based recursive partitioning", Research Report Series:Report 19, Department of Statistics and Mathematics, Wirtschaftsuniversität Wien, http://statistik.wu-wien.ac.at, 2005.