Visualizing Functional Data with an Application to eBay's Online Auctions

Wolfgang Jank, Galit Shmueli, Catherine Plaisant, Ben Shneiderman

5.1 Introduction

Technological advances in the measurement, collection, and storage of data have led to more and more complex data structures. Examples of such structures include measurements of the behavior of individuals over time, digitized two- or three-dimensional images of the brain, and recordings of three- or even four-dimensional movements of objects traveling through space and time. Such data, although recorded in a discrete fashion, are usually thought of as continuous objects that are represented by functional relationships. This gave rise to functional data analysis (FDA), which was made popular by the monographs of Ramsay and Silverman (1997, 2002), where the center of interest is a set of curves, shapes, objects, or, more generally, a set of *functional observations*, in contrast to classical statistics where interest centers on a set of data vectors. In that sense, functional data is not only different from the data structure studied in classical statistics, but it actually generalizes it. Many of these new data structures require new statistical methods to unveil the information that they carry.

There are many examples of functional data. The year-round temperature at a weather station can be thought of as a continuous curve, starting in January and ending in December, where the amplitude of the curve signifies the temperature level at each day or at each hour. A collection of temperature curves from different weather stations is then a set of functional data. Similarly, the price during an online auction of a certain product can be represented by a curve, and a sample of multiple auction price curves for the same product is then a set of functional objects. Alternatively, the digitized image of a car passing through a highway toll booth can be described by a two-dimensional curve measuring the pixel color or intensity of that image. A collection of image curves from all of the cars passing through the toll booth during a single day can then be considered to be a set of functional data. Lastly, the movement of a person through time and space can be described by a four- (or even higher) dimensional hyperplane in x-, y-, z- and time coordinates. The collection of all such hyperplanes from people passing through the same space is again a set of functional data.

Data visualization is an important part of any statistical analysis and it serves many different purposes. Visualization is useful for understanding the general structure and nature of the data, such as the types of variables contained in the data (categorical, numerical, text, etc.), their value ranges, and the balance between them. Visualization is useful for detecting missing data, and it can also aid in pinpointing extreme observations and outliers. Moreover, unknown trends and patterns in the data are often uncovered with the help of visualization. After identifying such patterns, they can then be investigated more formally using statistical models. The exact nature of these models (e.g., linear vs. log-linear) is again often based on insight learned from visualization. Finally, model assumptions are typically verified through the visualization of residuals and other model-related variables.

While visualization is an important step in comprehending any set of data, different types of data require different types of visualization. Take for instance the example of cross-sectional data vs. time series data. While the information in cross-sectional data can often be displayed satisfactorily with the help of standard bar charts, boxplots, histograms or scatter plots, time series data require special graphs that can also capture the temporal information. The methods used to display time-series data range from rather simple time-series plots, to streaming video clips for discrete time series (Mills et al., 2005), to cluster- and calendar-based visualization for more complex representations (van Wijk and van Selow, 1999).

Functional data are different to ordinary data in both structure and concept, and thus require special visualization methods. While the reasons for visualizing functional data are similar to those for ordinary data, functional data entail additional challenges that require extra attention. One such challenge is the creation of functional observations. Functional data are typically obtained by recovering the continuous functional object from the discrete observed data via data-smoothing. The implication of this is that there are two levels to the study of functional data. The first level uses the discrete observed data to recover the continuous functional object. Visualizing data at this level is important for detecting anomalies that are related to the data generation process, such as data collection and data entry, as well as for assessing the fit of the smoothed curves to the discrete observed data. This is illustrated and discussed further in Sect. 5.3. The second and higher level of study operates on the functional objects themselves. Since at this level the functional objects are the observations of interest, visualization is now used for the same reasons mentioned previously for ordinary data: to detect patterns and trends, possible relationships, and also anomalies. In Sect. 5.4 we describe different visualizations that enhance data comprehension and support more formal analyses.

The visualization of functional data has not received much attention in the literature to date. Most of the work in this area has focused on the derivation of mathematical models for functional data, with visualization playing a minor role and typically appearing only as a side product of the analysis. Some noteworthy exceptions include the display of summary statistics, such as the mean and the variability of a set of functional objects, the use of phase-plane plots to understand the interplay of dynamics, and the graphing of functional principal components to study sources of variability within functional data (Ramsay and Silverman, 2002). Another exception is the work of Shmueli and Jank (2005) and Hyde et al. (2005), which is focused directly on the visualization of functional data, and which suggests a few novel ideas for the display of functional data, such as *calendar plots* and *rug plots*.

Most of the visualizations currently used for functional data are static in nature. By "static" we mean a graph that can no longer be modified by the user without rerunning a piece of software code after it has been generated. A static approach is useful for differentiating subsets of curves by attributes (e.g., by using color), or for spotting outliers. A static approach, however, does not permit an interactive exploration of the data. By "interactive" we mean that the user can perform operations such as zooming in and out, filtering the data, and obtaining details about the filtered data, all within the environment of the graphical interface. Interactive visualizations that can be used for the special structure of functional data are not straightforward to devise, and so solutions have only recently begun to receive consideration (Aris et al., 2005; Shmueli et al., 2006). In Sect. 5.5 we describe an interactive visualization tool designed for the display and exploration of functional data. We illustrate its features and benefits using the example of price curves, which capture the price evolution in online auctions.

The insightful display of functional data comes with many, many different challenges, and we are only scraping the tip of the iceberg in this essay. Functional data is challenging due to its high object dimensionality, complex functional relationships and the concurrency present among the functional objects. We discuss some of these extra challenges in Sect. 5.6.

5.2 Online Auction Data from eBay

eBay (www.eBay.com) is one of the major online marketplaces and currently the biggest consumer-to-consumer online auction site. eBay offers a vast amount of rich bidding data. Besides the time and amount of each bid placed, eBay also records plenty of information about the bidders, the seller, and the product being auctioned. On any given day, several million auctions take place on eBay, and all closed auctions from the last 15 days are made publicly available on eBay's website. This huge amount of information can be quite overwhelming and confusing to the user (i.e., either the seller, the potential buyer, or the auction house) that wishes to incorporate this information into his/her decision-making process. Data visualization can help to alleviate this confusion.

Online auctions lend themselves naturally to the use of functional data for a variety of reasons. Online auctions can be conceptualized as a series of bids placed over time. The finite time horizon of the auction allows the study of the price evolution between the start and the end of the auction. By "price evolution" we mean the changes in the price due to new bids as the auction approaches its end. Conceptualizing the price evolution as a continuous price curve allows the researcher to investigate price dynamics via the price curve's first and second derivatives.

It is worth noting that empirical research into online auctions has largely ignored the temporal dimension of the bidding data, and has instead only considered a condensed snapshot of the auction. That is, most research has only considered the end of the auction by, for example, concentrating only on the final price rather than on the entire price curve, or by only looking at the total number of bidders rather than the function describing the bidder arrival process. Considering only the end of the auction results in information loss, since such an approach entirely ignores the way in which that end-point was reached. Functional data analysis is a natural solution that allows us to avoid this information loss. In a recent series of papers, the first two authors have taken a functional approach and shown that pairing the price evolution with its dynamics leads to a better understanding of different auction profiles (Jank and Shmueli, 2005) or to more accurate forecasts of the final auction price (Wang et al., 2005).

Visualization at the Object Recovery Stage

5.3

Any functional data set consists of a collection of continuous functional objects, such as a set of continuous curves describing the temperature changes over the course of a year, or the price 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 human and measurement error results in discrete observations that are noisy realizations of the underlying continuous curve. Thus, the first step in every functional data analysis is to recover the underlying continuous functional object from the observed data. This is typically done with the help of smoothing techniques.

A variety of different smoothers exist. One very flexible and computationally efficient choice is the penalized smoothing spline (Ruppert et al., 2003). Let $\tau_1, \dots, \tau_L$ be a set of knots. Then, a polynomial spline of order p is given by

$$f(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \dots + \beta_p t^p + \sum_{l=1}^{L} \beta_{pl}(t - \tau_l)_+^p , \tag{5.1}$$

where $u_+ = uI_{[u \geq 0]}$ denotes the positive part of the function u. Define the roughness penalty

$$\mathrm{PEN}_m(t) = \int \{D^m f(t)\}^2 \mathrm{d}t , \tag{5.2}$$

where $D^m f$, $m = 1, 2, 3, \dots$, denotes the mth derivative of the function f. The penalized smoothing spline f minimizes the penalized squared error

$$\mathrm{PENSS}_{\lambda,m} = \int \{y(t) - f(t)\}^2 \mathrm{d}t + \lambda\, \mathrm{PEN}_m(t) , \tag{5.3}$$

where $y(t)$ denotes the observed data at time t and the smoothing parameter λ controls the trade-off between the data fit and the smoothness of the function f. Using $m = 2$ in (5.3) leads to the commonly encountered cubic smoothing spline. Other possible smoothers include the use of B-splines or radial basis functions (Ruppert et al., 2003).

We want to emphasize that we use a common set of smoothing parameters across all functional objects. For instance, for the penalized smoothing splines, we pick a common set of knots $\tau_1, \dots, \tau_L$, a common spline order p, and a common penalizing term λ, and apply this common set of smoothing parameters to all functional objects, $1 \leq i \leq n$. The rationale behind using a common set is that it allows us to make comparisons among the individual functional objects. Conversely, if one were to use, say, a large value of λ for object i but a small value for object i', then it is not quite clear whether an observed difference between i and i' is attributable to a difference in the underlying population or instead to the difference in the smoothing parameters.

The actual choice of the smoothing parameters is often driven by the context. In our application, we pick the location and number of knots to reflect the bid-arrival distribution, which is densest for the last day, and in particular for the last few moments of the auction. The choice of p depends, among other things, on whether higher order derivatives of the curve are also desired. The value of the penalty term λ is chosen by inspecting the resulting functional objects in order to ensure satisfactory results (Ramsay and Silverman, 2002). An alternative approach is to pick λ so as to balance the smoothness and the data fit (Wang et al., 2005). In particular, one can measure the degree of smoothness of the spline via its distance to the smoothest possible fit, a straight line through the data. The data fit, on the other hand, can be measured as the distance between the spline and the actual data points. One then chooses a value of λ that balances the two. We investigate and compare different smoothing parameters for our dataset in what follows.

The process of moving from observed data to functional data is then as follows. For a set of n functional objects, let t_{ij} denote the time of the jth observation $(1 \leq j \leq n_i)$ of the ith object $(1 \leq i \leq n)$, and let $y_{ij} = y(t_{ij})$ denote the corresponding measurements. Let $f_i(t)$ denote the penalized smoothing spline fitted to $y_{i1}, \ldots, y_{in_i}$. Then, functional data analysis is performed on the continuous curves $f_i(t)$ rather than on the noisy observations $y_{i1}, \ldots, y_{in_i}$. That is, after creating the functional objects $f_i(t)$, the observed data $y_{i1}, \ldots, y_{in_i}$ are discarded and subsequent modeling, estimation and inference are based on the functional objects only.

One important implication of this practice is that any error or inaccuracy in the smoothing step will propagate into the inferences and conclusions made based on the functional model. To make matters worse, the observed data are discarded after the functional data are created and are therefore often hard to retrieve, and any violation of the functional model is confounded with the error at the smoothing step. That is, it is hard to know whether a model violation is due to model misspecification or due to anomalies at the smoothing step. For this reason, it is important to carefully monitor the functional object recovery process and to detect inaccuracies early in the process using appropriate tools. Although measures for evaluating the goodness of fit of the functional object to the observed data are available (such as those based on the residual sums of squares, or criteria that include the roughness penalty), it is unwise to rely on these measures alone, and visualization becomes an indispensable tool in the process.

Consider Figs. 5.1–5.3 for illustration. The figures compare the functional objects recovered for three different smoothing scenarios. Specifically, for bidding data from 16 different eBay online auctions, Fig. 5.1 shows the functional objects obtained from penalized smoothing splines using a spline order $p = 2$ and a small smoothing parameter $\lambda = 0.001$. Figure 5.2 on the other hand corresponds to the same spline order ($p = 2$) but a larger smoothing parameter ($\lambda = 1$). In Fig. 5.3 we use a spline order $p = 4$, a smoothing parameter $\lambda = 10$, and a data preprocessing step via interpolation. The exact details of the smoothing are not of interest here and can be found elsewhere Jank and Shmueli (2005). What *is* of interest here though is the fact that Figs. 5.1-5.3 correspond to three *different* approaches to recovering functional objects from the *same* data. The researcher could have taken either one of these three approaches and used the resulting functional objects for subsequent analysis. However, as we will explain next, two of the three approaches lead to very unrepresentative functional objects and therefore probably to erroneous conclusions.

Statistical conclusions typically make sense only in the context of their application, and ignorance thereof will lead to wrong conclusions. This is no different for visualizations. As mentioned earlier, Figs. 5.1-5.3 show bidding data from 16 eBay auctions. All auctions are for the same item (a *Palm PDA M515* personal digital assistant), all lasted seven days, and all auctions were collected during the same time period (March to June, 2003) and had a retail value of about $250 at the time of collection. In that sense, all 16 auctions are comparable. The circles correspond to the observed bids (i.e., the times and sizes of the bids), while the solid lines correspond to the resulting functional objects via penalized smoothing splines. The objective at this stage is to recover, from the observed bidding data, the underlying price curve. The price curve describes the price evolution during an auction, and its derivatives measure the price dynamics. In that sense, the objective is to create a functional object that is representative of the evolution of price between the start and end of the seven-day auction. The process of bidding on eBay follows an ascending format and the price curve should naturally reflect that. This goal is somewhat complicated by

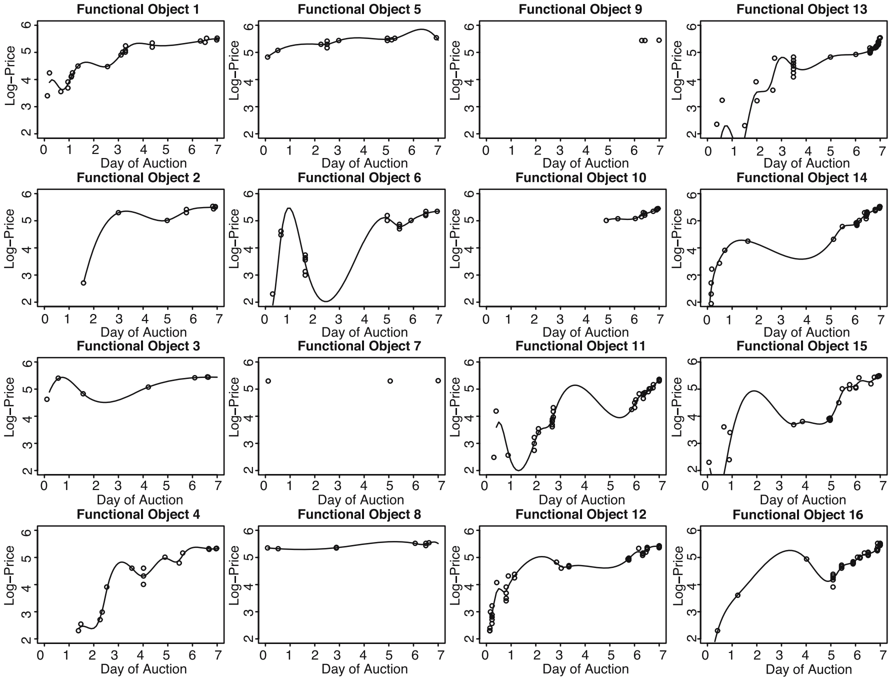
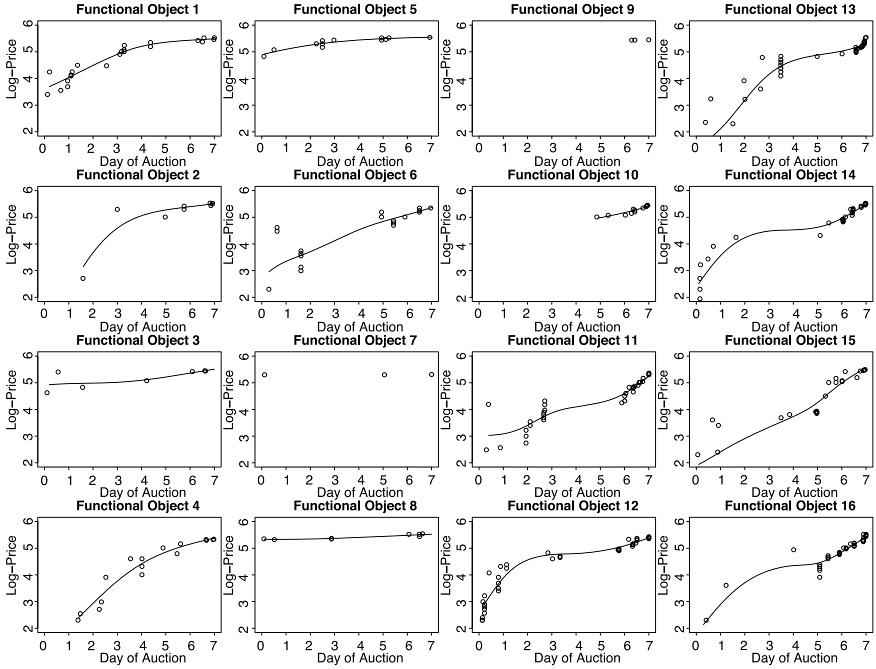


Figure 5.1. Creating functional objects: price curves using penalized smoothing splines with $p = 2$ and $\lambda = 0.001$. Note that the x-axis denotes the day of the auction, which is between 0 and 7, and the y-axis denotes the auction price on a log scale



Functional Object 1

Functional Object 5

Functional Object 9

Functional Object 13

Functional Object 2

Functional Object 6

Functional Object 10

Functional Object 14

Functional Object 3

Functional Object 7

Functional Object 11

Functional Object 15

Functional Object 4

Functional Object 8

Functional Object 12

Functional Object 16

Figure 5.2. Creating functional objects: price curves using penalized smoothing splines with $p = 2$ and $\lambda = 1$

the fact that observed bids do *not* increase monotonically due to eBay's proxy bidding system (Jank and Shmueli, 2005). Thus, creating representative functional objects is not a straightforward task.

Consider Fig. 5.1. We can see that the functional objects are very "wiggly" and certainly do not do a good job of representing the monotonic price increase in the auction. Moreover, we also notice that some of the objects (e.g., #2 and #10) only *partially* cover the seven-day period and thus do not represent the price evolution over the *entire* auction. The reason for this is the software: the penalized spline module *pspline* in *R*, by default, returns a function that is defined only on the range of the input data. Hence, since the bids for #2 and #10 cover only a small part of the duration of the auction, so does the resulting functional object. Lastly, we notice that there are no functional objects for #7 and #9. The reason for this is that the *pspline* module requires at least $2p + 1$ data points to estimate a smoothing spline of order p. This means that for a second-order smoothing spline we need at least $(2)(2) + 1 = 5$ points. However, both #7 and #9 only have three bids and thus no functional object is created. This loss of information is quite disturbing from a conceptual point of view, since data are in fact available for these two auctions and the missing (functional) data are a consequence of the functional object generation process. In summary, if the researcher were to use the smoothing approach from Fig. 5.1 "blindly" (i.e., without

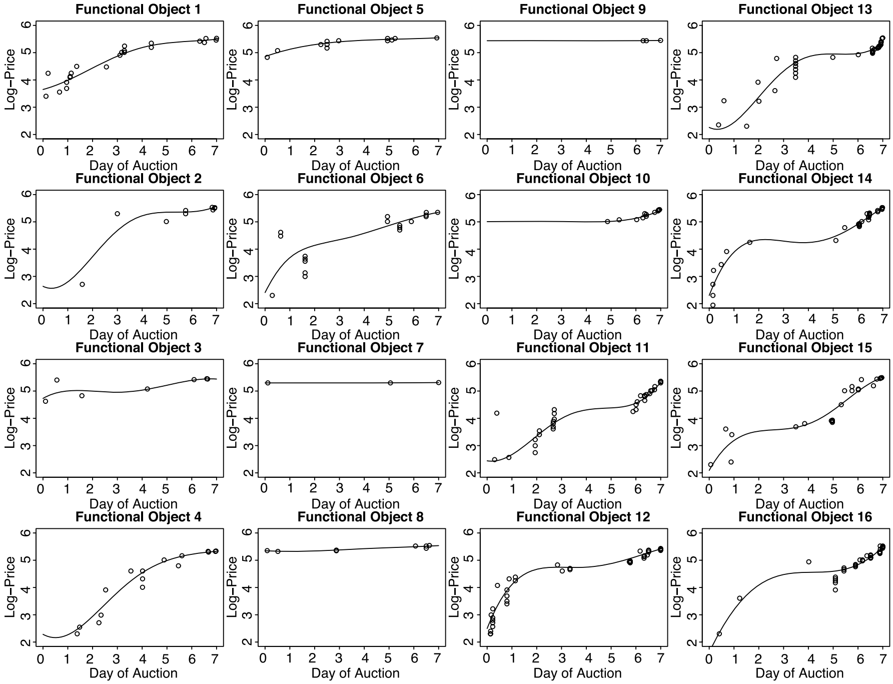


Figure 5.3. Creating functional objects: price curves using data preprocessing via interpolation of the bids and penalized smoothing splines with $p = 4$ and $\lambda = 10$

carefully checking the results), then very unrepresentative functional objects would be obtained and valuable information would be lost.

One reason for the poor representativeness of the objects in Fig. 5.1 is the low value of the smoothing parameter. Increasing λ to 1 (Fig. 5.2) results in much smoother (i.e., less wiggly) price curves. However, there are still some partial functional objects (#2, #10) and missing functional objects (#7, #9). Moreover, while the higher value of λ results in curves that are much less wiggly, some of the functional objects now appear to be too inflexible (e.g., #15 may be too similar to a straight line).

We can achieve a better fit (i.e., one with more flexibility, but only a little extra wiggliness) by increasing the order of the spline together with the magnitude of the smoothing parameter. We can also solve the problem of partial and missing functional objects by using a preprocessing step via interpolation. That is, we interpolate the observed bidding data and fit the smoothing spline to a discretized grid of the interpolating function. Specifically, let $\tilde{t}_{ij}$ denote the time that the jth bid is placed in auction i, and let $\tilde{y}_{ij}$ denote the corresponding bid amount. We interpolate the $\tilde{y}_{ij}$ values linearly to obtain the interpolating function $\tilde{y}_i(t)$. Let now $0 \leq t_j \leq 7$ be a common grid of time points. We evaluate all $\tilde{y}_i\,(t)$ values at the common grid points t_j to obtain $y_{ij} := \tilde{y}_i(t_j)$ and fit the smoothing spline to the y_{ij} values. In this way, we can ensure that we estimate the smoothing spline based on a sufficient number of points that cover the entire range of the seven-day auction. The results obtained from doing this can be seen in Fig. 5.3. Now the functional objects appear to be very representative of the price evolution, much better than in the previous two approaches. Equally importantly, there are now no missing or partial functional objects. Inference based on the objects in Fig. 5.3 is likely to yield the most reliable insights about the price evolution in online auctions.

The previous examples illustrate the importance of visualization at the object recovery stage. Although the causes of problems at this stage may often be quite trivial (e.g., unfortunate software default settings or poor parameter choices), they are typically hard to diagnose without the use of proper visualizations.

5.4 Visualizing Functional Observations

5.4.1 Visualizing Individual Objects and Their Dynamics

The first step in statistical analysis is typically to scrutinize data summaries and graphs. Data summaries include measures of central tendency, variability, skewness, etc. Traditionally, summary statistics are presented in numerical form. However, in the functional setting, each summary statistic is actually a functional object, such as the *mean function* or the *standard deviation function*. Since there are usually no analytical, closed-form representations of these functions, one resorts to graphical representations of the summary measures. The left panel in Fig. 5.4 shows the (pointwise) mean price curve (solid thick line) together with the 95 % pointwise upper and lower confidence curves (broken thick lines) for the 16 auctions from Sect. 5.3. We compute these pointwise measures in the following way. For an equally spaced grid $t_i \in [0,7]$, we compute the mean and standard deviation for the 16 auction prices at each grid point t_i. We use these two measures to construct 95 % upper and lower confidence bounds at each grid point. By interpolating the results, we obtain the mean and confidence curves in Fig. 5.4. Notice that since we only consider 16 auctions in this example, one can easily identify the minimum and maximum prices of all curves. In larger data sets, one may also want to add a curve for the (pointwise) minimum and maximum, respectively.

One of the main advantages of functional data analysis is that it allows for an estimation of derivatives. The nonparametric approach to the recovery of the functional object guarantees that local changes in the data are well-reflected, and yet the object's smoothness properties also allow for a reliable estimation of partial derivatives. For instance, setting $m = 4$ in the penalty term in (5.2) guarantees smooth first and second derivatives. Knowledge of the derivatives can result in an important advantage, especially for applications that experience change. Take the online auction setting as an example. While the price curve $f(t)$ describes the exact *position* of the price at any time point t, it does not reveal how fast the price is *moving*. Attributes that we typically associate with a moving object are its *velocity* (or its *speed*) and its *acceleration*. Velocity and acceleration can be computed via the first and second derivatives

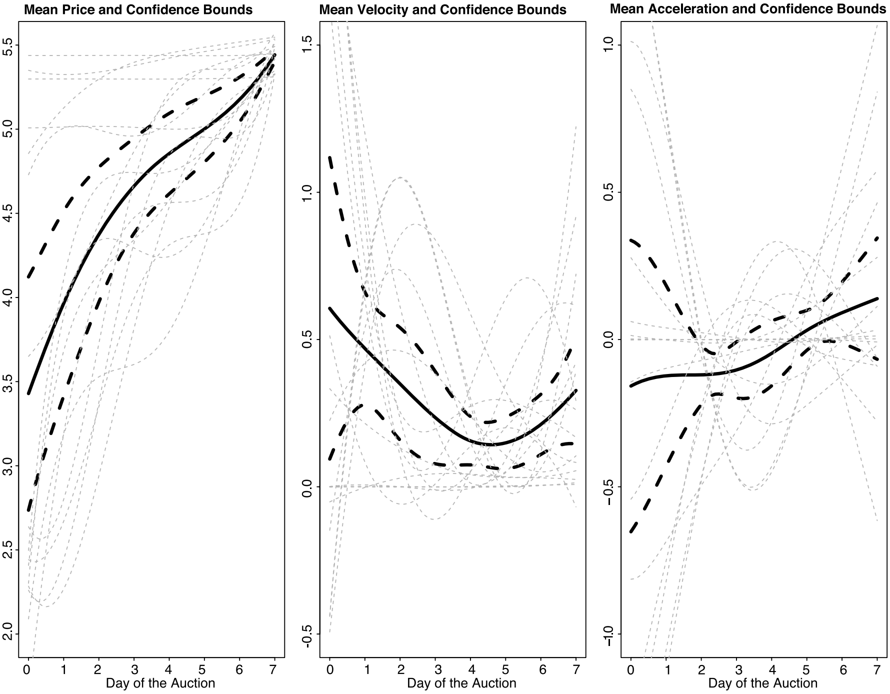


Figure 5.4. Summaries for functional objects: pointwise mean and 95 % pointwise confidence bounds for the price evolution, price velocity, and price acceleration of the 16 eBay online auctions

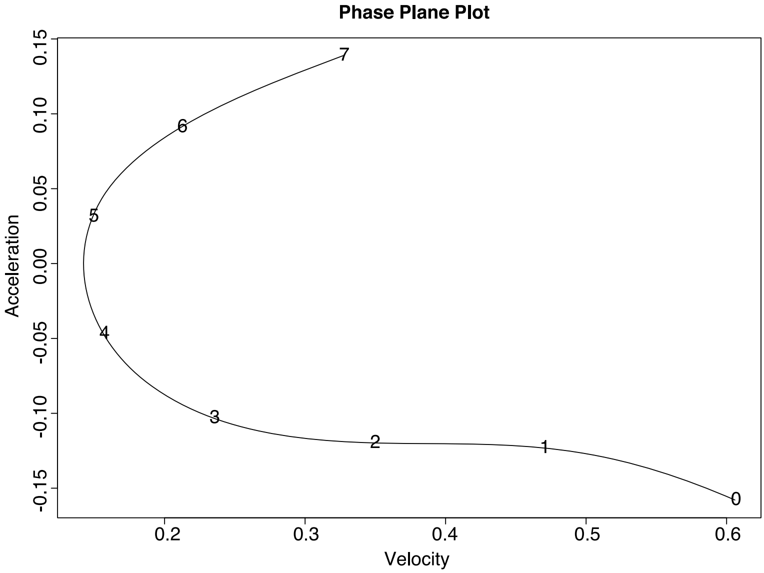


Figure 5.5. Phase-plane plot of the mean velocity vs. the mean acceleration. Each number on the curve indicates a particular day of the auction

of $f(t)$, respectively. Knowledge of the dynamics can be important for pinpointing the periods during which the auction price experiences only minor change, which in turn is important for forecasting the final price (Wang et al., 2005). The middle and right panels of Fig. 5.4 show the velocity and acceleration for the 16 eBay auctions together with the pointwise mean and confidence bounds.

Another way of investigating the interplay of dynamics is with the aid of so-called *phase-plane plots*. Phase-plane plots graph dynamics against one another. For instance, Fig. 5.5 shows a graph of mean velocity versus mean acceleration. The numbers on the curve indicate the day of the auction. We can see that at the start (day 0), high velocity is accompanied by low, negative acceleration (=deceleration). Acceleration precedes velocity, so deceleration *now* results in a lower velocity *tomorrow*, and consequently the velocity decreases to below 0.5 on day 1. This trend continues until the acceleration turns positive (between days 4 and 5), causing the velocity to pick up towards the end of the auction end. Phase-plane plots are useful for diagnosing whether the interplay of dynamics suggests a system that could be modeled by a suitable differential equation.

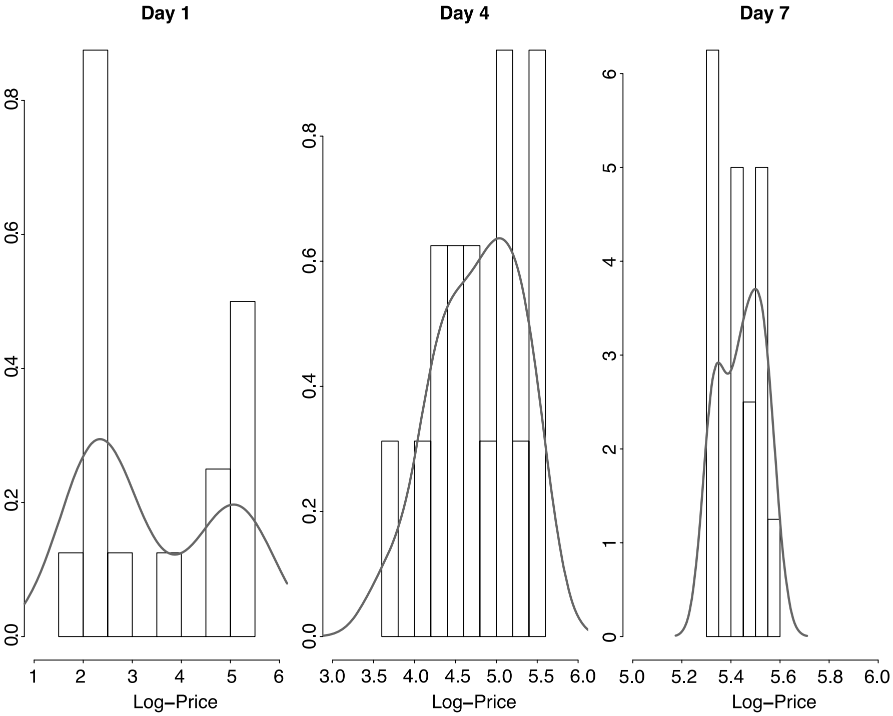


Figure 5.6. Distribution of functional objects: histograms of price (plotted on a log scale) at days 1, 4 and 7 of the 16 eBay online auctions. The *gray* line corresponds to a kernel density estimate with a Gaussian kernel

Another reason to explore the data is to investigate the distributions of individual variables. Since most parametric models require the response to follow a certain distribution (typically the normal distribution), this step is important for selecting the right model and for ensuring the appropriateness of the selected model. One standard tool for investigating the distribution of a numerical variable is the histogram. However, generalizing the idea of a histogram to the functional context is a challenging task, since the input variable is a continuous function. One solution is to graph the distribution of the functional object at only a few select snapshots in time. This can be done by discretizing the object and graphing pointwise histograms (or similar plots such as probability plots) at each time point. Figure 5.6 shows snapshots of the distributions of the 16 eBay price curves at days 1, 4 and 7. These snapshots allow conclusions to be drawn about the distribution of the entire functional object. Notice that Fig. 5.6 also shows kernel density estimates of the distributions and thus allows conclusions to be drawn about the evolution of the functional density over days 1–7. One can generalize this idea to obtain the density continuously over the entire functional object (rather than only at discrete time points). Specifically, Fig. 5.7 shows the density estimates evaluated over a fine grid and subsequently interpolated. We can see that the distribution is very flat at the beginning of the auction and it starts to peak towards the end.

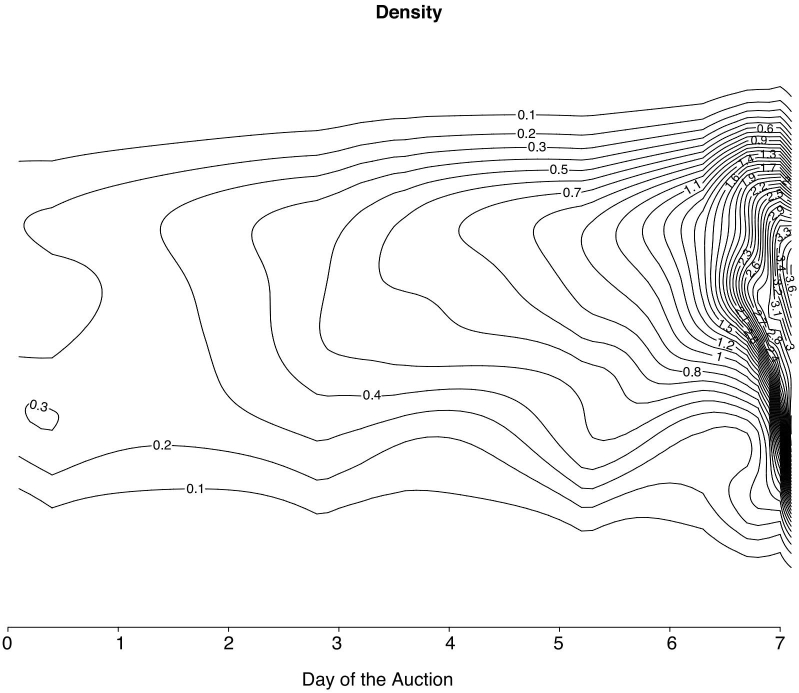


Figure 5.7. Contour plot of the density of the functional objects over the seven-day auction. The contour plot is obtained by calculating kernel density estimates (as done in Fig. 5.6) over a fine grid and subsequent interpolation over the seven-day period

5.4.2 Visualizing Relationships Among Functional Data

After examining each variable individually, the next step in exploratory data analysis is typically to investigate relationships across several variables. For two numerical variables, this is often accomplished with the help of scatterplots. One way of generalizing the traditional scatterplot to the functional setting is, again, to draw a sequence of *pointwise* scatterplots. Figure 5.8 shows scatterplots at days 1, 4 and 7 for the auction price versus the opening bid (on a log scale). We can see that the relationship between the two variables changes over the course of time. While there is a strong positive effect at the beginning of the auction (left panel), the magnitude of the effect decreases at day 4 (middle panel), and there is barely any effect at all (possibly even a slightly negative effect) at the end of the auction (right panel). This suggests that the relationship between the opening bid and the auction price can be modeled well using a time-varying coefficient model. Of course, one aspect that remains unexplored in this pointwise approach is a possible three-way interaction between the opening bid, the price and the time. Such an interaction could be detected using a three-dimensional scatterplot. However, as the left panel in Fig. 5.9 illustrates, three-dimensional graphs have the disadvantage that they are often cluttered and difficult to read. We can improve the interpretability by using smoothing. The right panel in

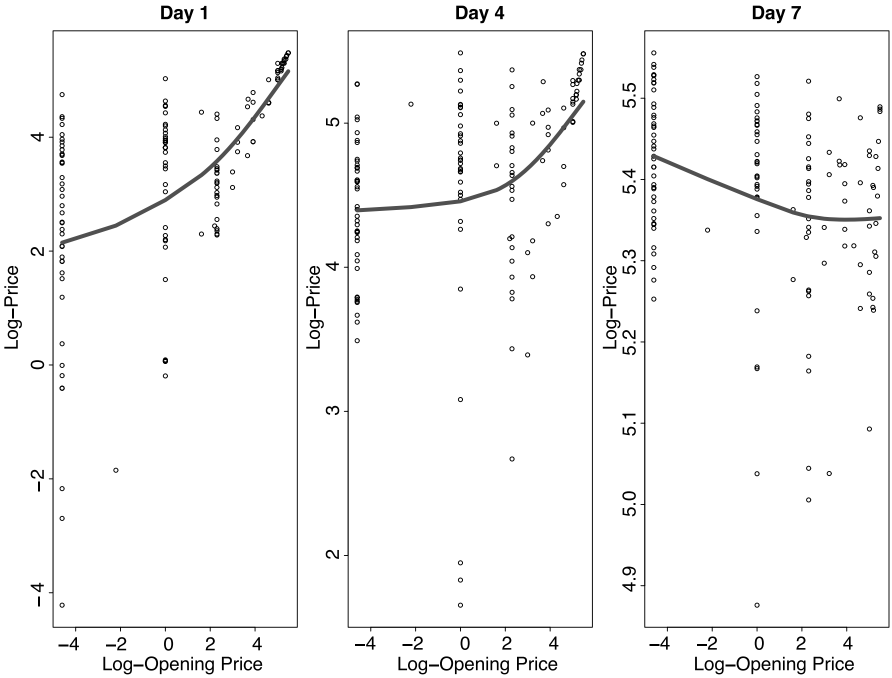


Figure 5.8. Relationships among functional objects: scatterplots of (log) price vs. (log) opening bid at days 1, 4 and 7 for a sample of eBay online auctions. The *solid gray line* corresponds to a cubic smoothing spline with three degrees of freedom

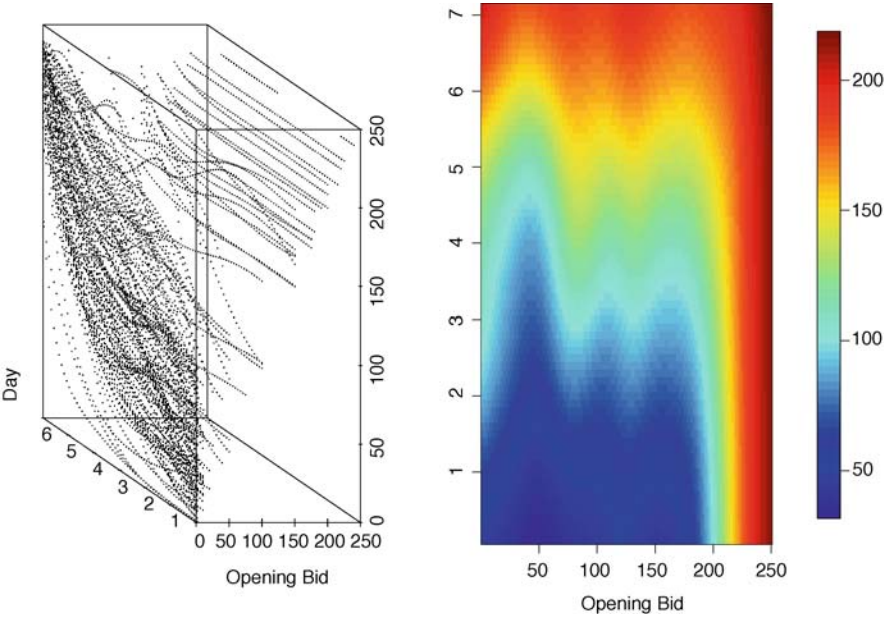


Figure 5.9. [This figure also appears in the color insert.] Relationships among functional objects: the *left panel* shows a 3-D scatterplot of opening bid (x), day of the auction (y) and price (z). The right panel shows a smoother version of the price surface, obtained using a Nadaraya–Watson smoother. In that plot, the x-axis is the opening bid and the y-axis is the day of the auction

Fig. 5.9 shows a smoother image of the price surface obtained using a Nadaraya–Watson smoother. The three-way relationship between opening bid, price and time is now much easier to see.

Visualizing Functional and Cross-sectional Information 5.4.3

As illustrated above, it is more challenging to visualize functional data than classical data. The visualization process is often further complicated by the coupling of functional observations with cross-sectional attribute data. For example, online auction data include not only the bid history (i.e., the times and sizes of bids), but also auction-specific attributes corresponding to auction design (e.g., length of the auction, magnitude of the opening bid, use of a secret reserve price, use of the "Buy It Now" option, etc.), bidder characteristics (e.g., bidder IDs and ratings), seller characteristics (e.g., seller ID and rating, seller location, whether or not a seller is a "Powerseller," etc.), and product characteristics (e.g., product category, product quality and quantity, product description, etc.). All of these characteristics correspond to cross-sectional information in that they do not change during the auction. The coupling of time series with cross-sectional information is important because the relationship between the two could be the main aim or at least one of the aims of the analysis.

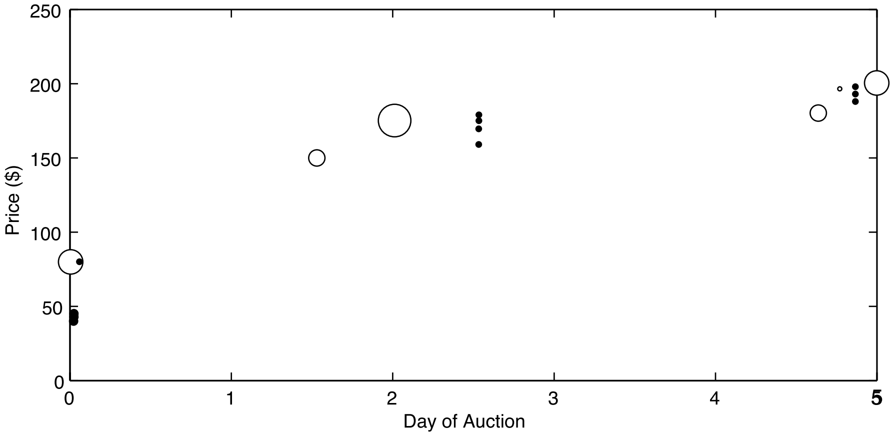


Figure 5.10. Profile plot of a single five-day auction. The *circles* represent bids, with circle size proportional to the bidder's eBay rating

Standard visualization tools are geared towards the display of either time-series data alone or cross-sectional data alone – almost never both.

The combination of time-series and cross-sectional data into one display is rare and requires careful, application-specific modifications of standard methods. Shmueli & Jank (2005) propose the use of *profile plots* for displaying the temporal sequence of bids together with additional auction attributes (such as a bidder's rating) in the same graph. This is illustrated in Fig. 5.10, which describes the sequence of bids in a five-day eBay auction. The circle size is proportional to the bidder's eBay rating. However, profile plots are more suitable for visualizing single auctions, and do not scale well.

Another type of plot that is suitable for visualizing functional data is the *rug plot* (Hyde et al., 2005). A rug plot displays curves (i.e., functional objects) over calendar time in order to explore the effects of event concurrency. Figure 5.11 shows a rug plot displaying the price curves of 217 eBay auctions of a *Palm M515 PDA* that took place over a three-month period. Each colored line represents an individual auction, and it is estimated via monotone smoothing (Ramsay and Silverman, 2005). Monotone smoothing is computationally more expensive than penalized smoothing, but it ensures that the resulting line increases monotonically. The black line represents the average daily closing price. We can see that daily prices vary quite significantly, and so does the daily variation in price (gray bands). Furthermore, we can see that there are time periods with many similar, almost parallel price curves for the same auction durations (e.g., seven-day auctions – green curves – at around 4/3 and also around 4/23). Moreover, the closing prices after 4/3 appear to be relatively low, and so does the associated price variability. Most auctions closing at that time are seven-day auctions with similar shapes. It would be interesting to see if one could establish a more formal relationship between similar price patterns (i.e., parallel price curves) and their effect on the price and its uncertainty.

The rug plot in this example combines functional data with attribute data via the time axis (calendar time on the x-axis takes into account the start and the end of

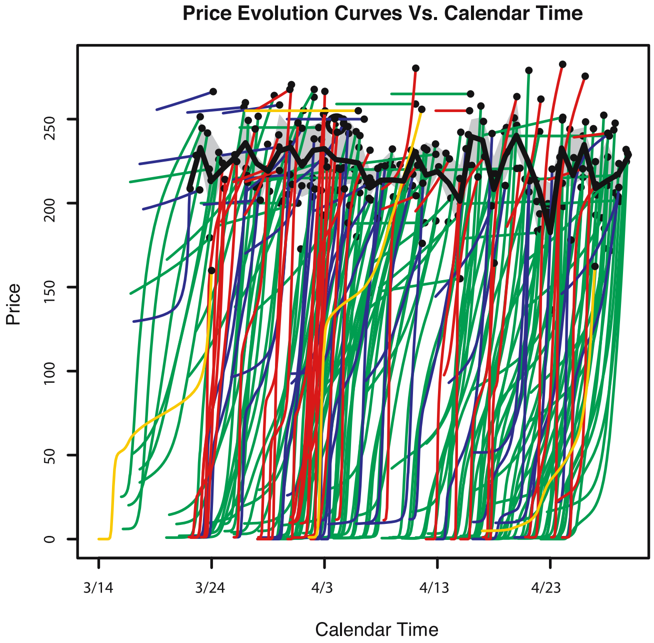


Figure 5.11. [This figure also appears in the color insert.] Rug plot displaying the price evolution (y-axis) of 217 online auctions over calendar time (x-axis) during a three-month period. The colored lines show the price path of each auction, with color indicating auction length (*yellow*, three days; *blue*, five days; *green*, seven days; *red*, ten days). The *dot* at the end of each line indicates the final price of the auction. The *black line* represents the average of the daily closing price, and the gray band is the interquartile range

the curve) and via color (different colors for different auction durations). Notice that the plot scales well for a large number of auctions, but it is limited in the number of attributes that can be coupled within the visualization.

Finally, trellis displays (Cleveland et al., 1996) are another method that supports the visualization of relationships between functional data and an attribute of interest. This is achieved by displaying a series of panels where the functional objects are displayed at different levels (or categories) of the attribute of interest (see for instance Shmueli and Jank, 2005). In general, while static graphs can capture some of the relationships between time series and cross-sectional information, they become less and less insightful as the dimensionality and complexity of the data increase. One of the reasons for this is that they have to accomplish meaningful visualizations at several data levels: relationships within cross-sectional data (e.g., find relationships between the opening bid and a seller's rating) and within time-series data (e.g., find an association between the bid magnitudes, which is a sequence over time, and the number of bids, which is yet another sequence over time). To complicate matters, these graphs also need to portray relationships across the different data types; for example, between the opening bid and the bid magnitudes. In short, the graphs have to be very flexible to accommodate all of these different criteria. Ideally, one would want to literally "dive" into the data and explore it interactively.

Information visualization tools apply several common strategies that enable user control over data displays (see Shneiderman and Plaisant (2004), Card et al. (1999) or Plaisant (2005)). A primary strategy is to allow the user to manipulate a set of widgets, such as dynamic query sliders that allow the user to select the ranges of the desired variables, in a process often called *conditioning*. The power of interaction is that users can rapidly (100 ms) and incrementally change the ranges and explore the effect of these changes on the display. For example, users can move a slider to gradually eliminate auctions with low starting prices and see if that removes time series plots that end with low, middle, or high closing prices. A second strategy is to have multiple views of the data, such as scattergram, histogram, tabular, or parallel coordinate views. The users can then select a single or multiple items in one view and see the results in another view ("brushing"). For example, users can select the time series with sharp increases near the finish in order to see if these had relatively few previous bids.

Selectivity and user control are essential, as they support exploration (to confirm hypotheses) and discovery (to generate new hypotheses) (Chen, 2004). The large number of possibly interesting features in high-dimensional data means that static displays and a fixed set of data-mining algorithms may not be enough. Users can quickly spot unusual outliers, bimodal distributions, spikes, long or short tails on one side of a distribution, and surprising clusters or gaps. Users may also detect strong or weak relationships, which can be positive or negative, linear, quadratic, sinusoidal, exponential, etc.

The strongest tools are likely to be those that combine data-mining algorithms with potent user interfaces (Shneiderman, 2002). These have the potential to provide thorough coverage through a systematic process of exploration in which users can decompose a complex problem domain into a series of simpler explorations with ranking criteria, and they guide user attention to potentially interesting features and relationships (Seo and Shneiderman, 2005).

5.5 Interactive Information Visualization of Functional and Cross-sectional Information via TimeSearcher

TimeSearcher is a time series visualization tool developed at the Human–Computer Interaction Laboratory (HCIL) of the University of Maryland. *TimeSearcher* enables users to see an overview of long time series (> 10 000 points), to view multivariate time series, to select data with rectangular time boxes, and to search for a selected pattern. Its main strength comes from its interactivity, which allows users to explore time series data in an active way. Unlike static graphs, an interactive approach can be more powerful and can lead to a better understanding of the data.

TimeSearcher can be used to visualize functional data if a discretized version of the curves is used as input. The level of discretization is chosen by the user, and is generally selected such that the interpolated points result in apparently continuous curves. In a collaborative project, the authors (two statisticians and two computer scientists from HCIL) further developed the tool to accommodate a particular type of functional data, namely price curves from online auctions. As described in Sect. 5.2, auction data include bid histories, which we convert to smooth curves, and additional attributes. To illustrate the enhanced features of *TimeSearcher* that support functional data exploration, we use a dataset of 34 magazine auctions on eBay that took place during the fall of 2004. The data include the bid histories (converted to curves) and the attributes for each auction.

The first step involves aligning the auctions of different durations that took place at different times. We chose to align the time scale so that in *TimeSearcher* the x-axis shows the proportion of the full duration of the auction. We then added the auction duration and the additional lost temporal information (day and time at which the auction commenced and finished) to the list of attributes.

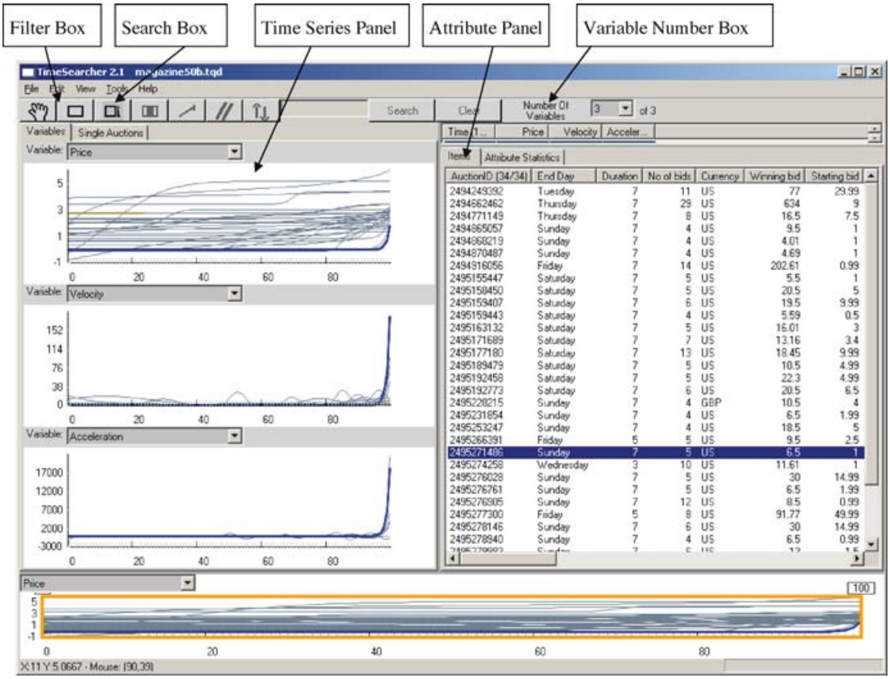
Capabilities of TimeSearcher 5.5.1

TimeSearcher was extended for the analysis of online auction data to include attribute data browsing with tabular views and filtering by attribute values and ranges (e.g., start date or seller), which were both tightly coupled to the time series visualization. The application is available for download from http://www.cs.umd.edu/hcil/timesearcher. Figure 5.12 shows the main screen of the visualization tool with a dataset of 34 eBay auctions for magazines. The time series are displayed in the left panel, with three series (i.e., three variables) for each auction: "Price" (top), "Velocity" (middle), and "Acceleration" (bottom), which correspond to the price curves and their first and second derivatives, as explained in the previous section. At the bottom of the screen, an overview of the entire time period covered by the auctions is provided to allow users to specify time periods of interest to be displayed in more detail in the left panel. On the right, the attribute panel shows a table of auction attributes. Each row corresponds to an auction, and each column to an attribute, starting with the auction ID number. In this dataset there are 21 attributes, and scrolling provides access to attributes that do not fit into the available space. Users can choose how much screen space is allocated for the different panels by dragging the separators between the panels, enlarging some panels and reducing others. All three panels are tightly coupled so that an interaction in one of the panels is immediately reflected in the other panels. Attributes are matched with time series using the auction ID number as a link.

The interactive visualization operations can be divided into time series operations (functional data) and attribute operations. We describe these next.

Functional Object Operations

TimeSearcher treats each time series, represented by a curve, as a single observation, and allows operations on the complete curve or on subsets of it. The following operations can be applied to the functional data (curves).



AuctionID (34/34)	End Day	Duration	No of bids	Currency	Winning bid	Starting bid
2494249392	Tuesday	7	11	US	77	29.99
2494662462	Thursday	7	29	US	634	9
2494771149	Thursday	7	8	US	16.5	7.5
2494865057	Sunday	7	4	US	9.5	1
2494868219	Sunday	7	4	US	4.01	1
2494870487	Sunday	7	4	US	4.69	1
2494916056	Friday	7	14	US	202.61	0.99
2495155447	Saturday	7	5	US	5.5	1
2495158450	Saturday	7	5	US	20.5	5
2495159407	Saturday	7	6	US	19.5	9.99
2495159443	Saturday	7	4	US	5.59	0.5
2495163132	Saturday	7	5	US	16.01	3
2495171689	Saturday	7	7	US	13.16	3.4
2495177180	Saturday	7	13	US	18.45	9.99
2495189479	Saturday	7	5	US	10.5	4.99
2495192458	Saturday	7	5	US	22.3	4.99
2495192773	Saturday	7	6	US	20.5	6.5
2495228215	Sunday	7	4	GBP	10.5	4
2495231854	Sunday	7	4	US	6.5	1.99
2495253247	Sunday	7	4	US	18.5	5
2495266391	Friday	5	5	US	9.5	2.5
2495271486	Sunday	7	5	US	6.5	1
2495274258	Wednesday	3	10	US	11.61	1
2495276028	Sunday	7	5	US	30	14.99
2495276761	Sunday	7	5	US	6.5	1.99
2495276905	Sunday	7	12	US	8.5	0.99
2495277300	Friday	5	8	US	91.77	49.99
2495278146	Sunday	7	6	US	30	14.99
2495278940	Sunday	7	4	US	6.5	0.99

Figure 5.12. The main screen of *TimeSearcher*, showing price curves and dynamics curves (*left*) coupled with attribute data (*right*) for 34 online auctions

Curve selection: A particular curve (or a set of curves) is selected by mouse-clicking on any point in that curve. The selected curve is then highlighted in blue (see Fig. 5.12). Hovering over a curve will highlight it in orange, thereby simplifying the task of mouse coordination.

Zooming: The overview panel at the bottom of the screen displays the time series for one of the variables and allows users to specify the part of the time series they want to zoom in on. The orange field of the view box determines the time range that is displayed on the upper left panels. Any one of the panels can be used for the display in the orange field. To zoom, users drag the sides of the box. By zooming in, the user can focus on a specific period in the data and see more details. In many cases zooming also results in better separation between the curves, enabling easier selection and deselection of lines. The box can also be dragged right and left to pan the display and show a different time period. Regardless of the range of the detail view, the overview always displays the entire time series and provides context for the detail view.

Focusing on a variable: To focus on a certain variable (price, velocity, or acceleration curves), users can choose to view only the panel on the left, which provides a larger view of those curves. This results in clearer separation between curves, which can be especially useful when there are many auctions. Users can specify the number of variables to be shown (here one, two or three) and select which of the variables should be displayed. This allows extra flexibility in terms of the choice of derivatives to display.

Filtering curves: Users can filter the curves to see only auctions of interest by using filter widgets called TimeBoxes. One can click on the TimeBox icon of the toolbar and draw a box on the time series panel of interest. Every curve that passes through the box (between the bottom and top edges of the box for the duration that the box occupies) is kept, while all of the other curves are filtered by graying them out. The corresponding auctions are also removed from the attribute panel on the right. Figure 5.13 shows a typical filter TimeBox used to view only the auctions that end with high price velocity. In the attributes panel, users can see that all of these auctions finished around the weekend. They can apply multiple TimeBoxes to the same or separate variables in order to form conjunctive queries (i.e., a logical AND combination of individual TimeBox queries). For example, users can search for auctions that end with low prices and high velocities.

Searching for patterns in curves: When comparing price curves and (especially) price dynamics, one useful tool is the pattern search. This is achieved by drawing a SearchBox on a selected curve for a certain time duration. The pattern is the part of the series that the SearchBox covers horizontally, and this SearchBox is searched Simultaneously for all other curves at any time point in the auction. There is a tolerance handle on the right of the SearchBox that allows the similarity to be specified. For example, users can search for auctions that have price curves with steep escalations at any time during the auction. TimeBoxes and SearchBoxes can be combined into multistep interactive searches.

Functional summaries: One can obtain numerical summaries for a set of functional objects using the *riverplot* in Fig. 5.14. The riverplot is a continuous form of the boxplot and displays the (pointwise) median together with the 25 % and 75 % confidence bounds. The riverplot allows for a condensed display of the average behavior of all curves together with the uncertainty around this average.

Attribute Operations

Manipulating the attribute data and observing the coupled functional data are useful ways of learning about relationships within the data across the different data types. The following operations support such explorations (in addition to more standard explorations of attribute data alone).

Sorting auctions: Users can sort the auctions by any attribute by clicking on the attribute name in the first row. A click sorts in ascending order, while the next click sorts in descending order. Sorting can be performed on numerical as well as text attributes. The sorting also recognizes day-of-the-week and time formats. The sorting is useful for learning about the ranges of the values for the different attributes, the existence of outliers, the absence of certain values, and possible errors and duplications in the data. Furthermore, sorting can allow users to visually spot patterns of "similar" auctions, by making auctions with similar attribute values appear consecutively in the auction list. Users can sort according to more than one column. In addition, the order of the attribute columns can be changed by clicking and dragging the attribute names to the right or left.

Highlighting groups of auctions: After the attribute/s of interest have been sorted, groups of auctions can be selected and their corresponding time series in the left panels are highlighted. For example, if the attributes table is sorted by the end day of the auction, it is easy to select all auctions that ended on a weekday from the table, and see the corresponding time series highlighted, which reveals that they are the auctions that tend to end with the highest prices (Fig. 5.12).

Summary statistics: The summary statistics tab shows the mean, standard deviation, minimum, max, median, and the quartiles for each attribute for the selected auctions. This is updated interactively when the auctions are filtered with TimeBoxes, or when users select a subset of auctions manually. For example, while the median seller rating of all auctions is 615, when users apply a TimeBox to select the auctions that started with a low price, the median seller rating jumps to 1487. Moving the TimeBox to select auctions that started with a high price results in a median seller rating of 243, which may imply that starting the auction with a low starting price is a strategy that tends to be employed by experienced sellers.

The array of interactive operations described above support data exploration, and Shmueli et al. (2006) describe how these operations can be used for the purpose of decision-making, through a semi-structured exploration. Exploration can be guided by a set of hypotheses, and the results can then help the user to find support for and direct the direct the user towards suitable formal statistical models. In particular, they show how insights gained from the visual exploration can improve seller, bidder, auction house, and other vendors' understanding of the market, thereby assisting their decision-making processes.

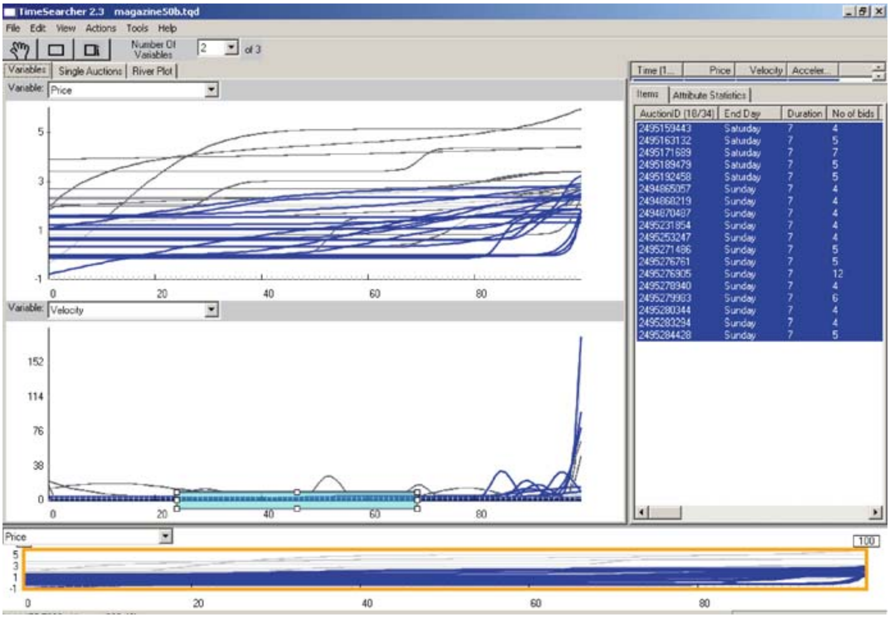
5.5.2 Forecasting with TimeSearcher

Functional object value forecasting is an area of functional data analysis that has not received much attention so far. It involves forecasting the value of a curve at a particular time t (either a particular curve in the data or the average curve), based on information contained in the functional data and the attribute data. We propose the following general forecasting procedure:

Select similar items: For a partial curve (e.g., an ongoing auction that has not closed), we select the subset of curves that are closest to the curve of interest in the sense of similar attributes and curve evolutions and dynamics. For the attribute criterion, this can be achieved either by sorting by attributes and selecting items with similar values for the relevant attributes (e.g., auctions of the same duration and with the same opening price), or directly by using a filtering facility that allows the user to specify limits on the values of each of the attributes of interest (this facility is currently not available in the public version of *TimeSearcher*). When curve-matching, TimeBoxes can be used to find curves that have similar structures during the time periods of interest (e.g., auctions with high price velocities on day 1 and high prices on day 3). We are currently working on developing a facility for "curve-matching" that is more automated. For instance, consider the case of forecasting the closing price of a seven-day auction that is scheduled to close on a Sunday, with an opening price of $0.99, and that has displayed very low dynamics so far. Let us assume that we observe this auction until day 6 (85 % of the auction duration). Figure 5.13 illustrates a selection of auctions that all have similar attributes to the above auction (all are seven days long, have an opening price of less than $5, and close on a weekend), and also have similar curve structures during the first six days of the auction (low velocities, as shown by the filtering box placed on the velocity curves).

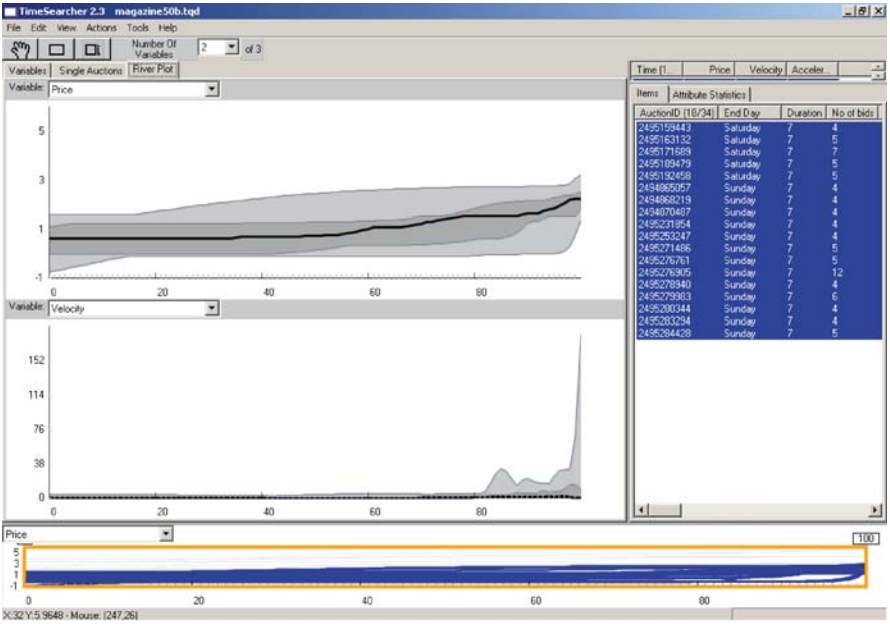
similar set: Make a forecast based on the We then use the selected "similar" set of curves to make a prediction for time *t* by examining their riverplots. The median at time *t* is then the forecast of interest, and the quartiles at that time can serve as a confidence interval. Although this is a very crude method, it is similar in concept to collaborative filtering. The key is to have a large enough dataset, so that the "similar" subset is large enough. To continue our illustration, Fig. 5.14 shows the riverplot of the subset of "similar" auctions. The forecasted closing price is then the median of the closing prices of the subset of auctions, and we can learn about the variability in these values from the percentile curves on the riverplot.

The forecasting module is still under development, with the goal being a more automated process. However, the underlying concept is that interactive visualization can support more advanced operations (including forecasting) than static visualization.



AuctionID (18/34)	End Day	Duration	No of bids
2495159443	Saturday	7	4
2495163132	Saturday	7	5
2495171689	Saturday	7	7
2495189479	Saturday	7	5
2495192458	Saturday	7	5
2494865057	Sunday	7	4
2494868219	Sunday	7	4
2494870487	Sunday	7	4
2495231854	Sunday	7	4
2495253247	Sunday	7	4
2495271486	Sunday	7	5
2495276761	Sunday	7	5
2495276905	Sunday	7	12
2495278940	Sunday	7	4
2495279983	Sunday	7	6
2495280344	Sunday	7	4
2495283294	Sunday	7	4
2495284428	Sunday	7	5

Figure 5.13. Filtering the data to find a set of "similar" auctions to an ongoing open auction



AuctionID (18/34)	End Day	Duration	No of bids
2495159443	Saturday	7	4
2495163132	Saturday	7	5
2495171689	Saturday	7	7
2495189479	Saturday	7	5
2495192458	Saturday	7	5
2494865057	Sunday	7	4
2494868219	Sunday	7	4
2494870487	Sunday	7	4
2495231854	Sunday	7	4
2495253247	Sunday	7	4
2495271486	Sunday	7	5
2495276761	Sunday	7	5
2495276905	Sunday	7	12
2495278940	Sunday	7	4
2495279983	Sunday	7	6
2495280344	Sunday	7	4
2495283294	Sunday	7	4
2495284428	Sunday	7	5

Figure 5.14. Riverplot of the subset of "similar" auctions. The *thick black line* is the pointwise median used for forecasting. The *dark gray bands* around the median show the 25 and 75 % percentile range and the *light gray bands* show the envelope for all similar auctions. This can be seen as a continuous form of box plot

Further Challenges and Future Directions 5.6

Functional data analysis is an area of statistical research that is receiving a growing amount of interest. To date, most of this interest has centered around developing new functional models and techniques for estimating them, while little effort has been expended on exploratory techniques, especially visualization. Classical statistics has become very popular due to both the availability of a wide array of models and the ability to check the appropriateness of these models. The results obtained by applying a particular model will only be wholeheartedly supported if the model is shown to be appropriate. However, this requires that the data can be compared to the model. In this sense, the widespread acceptance and usage of functional models is only going to happen when a range of visualization tools that perform similar tasks to their counterparts in classical statistics are made available.

In this paper, we have outlined a variety of functional visualizations that are available. However, significant challenges remain. These challenges range from concurrency of functional objects, to high dimensionality, to complex functional relationships.

Concurrency of Functional Events

5.6.1

The standard assumption in functional data analysis is independence of the functional observations in the data set. This assumption may not, however, always be plausible. For instance, if the functional object represents the evolution of the price in an online auction, then it is quite possible that the price in *one* auction is affected by the price of an object in *another* action. That is, if the price in one auction jumps to an unexpectedly high level, then this may cause some bidders to leave that auction and move on to another auction of a similar item. This results in a dependence in price between the two auctions. Or more generally, the result is a dependence between the two functional objects. Capturing this kind of dependence in a mathematical model is not a straightforward task. For a start, how can we unveil such a concurrency in graphical fashion? One promising attempt in this direction is the work of Hyde et al. (2005), which suggests that *rug plots* can be used for the functional objects and their derivatives.

Dimensionality of Functional Data

5.6.2

Another challenge when visualizing functional data is the dimensionality of the data. As pointed out earlier, it is not uncommon for functional data to have three, four or even more dimensions. Most standard visualization techniques work well for two dimensions at most, which is the number of dimensions of the paper that we write on and the computer screen that we look at. Moving beyond two dimensions is a challenge in any kind of visualization task, including that of visualizing functional data.

Complex Functional Relationships

5.6.3

In addition to the high dimensionality, functional data is also often characterized by complex functional relationships. Take for instance the movement of a object through time and space. This movement may be well characterized by a three- or four-dimensional *differential equation* (Ramsay and Silverman, 2002). However, how should we visualize a differential equation? One way is to use phase-plane plots like that in Fig. 5.5. Other approaches have been proposed in Schwalbe (1996).

References

Aris, A., Shneiderman, B., Plaisant, C., Shmueli, G. and Jank, W. (2005). Representing unevenly-spaced time series data for visualization and interactive exploration. In: *International Conference on Human Computer Interaction (INTERACT 2005)*, 12–16 Sept 2005, Rome, Italy.

Card, S., Mackinlay, J. and Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco, CA.

Chen, C. (2004). *Information Visualization: Beyond the Horizon*. Springer, Berlin.

Cleveland, W.S., Shyu, M. and Becker, R. (1996). The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5:123–155.

Hyde, V., Jank, W. and Shmueli, G. (2005). Investigating concurrency in online auctions through visualization. *The American Statistician*, 34(3):241–250.

Jank, W. and Shmueli, G. (2005). *Profiling price dynamics in online auctions using curve clustering*. Technical report, Smith School of Business, University of Maryland, College Park, MD.

Mills, K., Norminton, T. and Mills, S. (2005). Visualization of network scanning (poster presentation). In: *National Defense and Homeland Security Kickoff Workshop of the Statistical and Applied Mathematical Sciences Institute (SAMSI)*, 11–15 Sept 2005, Research Triangle Park, NC.

Plaisant, C. (2005). Information Visualization and the Challenge of Universal Access. In: *Exploring Geovisualization*. Elsevier, Oxford.

Ramsay, J.O. and Silverman, B.W. (2005). *Functional data analysis*, 2nd edn. Springer, New York.

Ramsay, J.O. and Silverman, B.W. (2002). *Applied functional data analysis: methods and case studies*. Springer, New York.

Ruppert, D., Wand, M.P. and Carroll, R.J. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.

Schwalbe, D. (1996). *VisualDSolve: Visualizing Differential Equations with Mathematica*. TELOS/Springer, Berlin.

Seo, J. and Shneiderman, B. (2005). A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4:99–113.

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

Shmueli, G., Jank, W., Aris, A., Plaisant, C. and Shneiderman, B. (2006). Exploring auction databases through interactive visualization. *Decision Support Systems*, 42(3):1521–1538.

Shneiderman, B. (2002). Inventing discovery tools: Combining information visualization with data mining. *Information Visualization*, 1:5–12.

Shneiderman, B. and Plaisant, C. (2004). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 4th edn. Addison-Wesley, Reading, MA.

van Wijk, J.J. and van Selow, E. (1999). Cluster and calendar-based visualization of time series data. In Wills, G. and Keim, D. (eds), *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)*. IEEE, Los Alamitos, CA, pp. 4–9.

Wang, S., Jank, W. and Shmueli, G. (2007). Explaining and Forecasting Online Auction Prices and their Dynamics using Functional Data Analysis. Forthcoming at the *Journal of Business and Economic Statistics*.