Modelling price paths in on-line auctions: smoothing sparse and unevenly sampled curves by using semiparametric mixed models

Florian Reithinger,
*Ludwig-Maximilians-Universität, München, Germany*

Wolfgang Jank,
*University of Maryland, College Park, USA*

Gerhard Tutz
*Ludwig-Maximilians-Universität, München, Germany*

and Galit Shmueli
*University of Maryland, College Park, USA*

[Received August 2006. Final revision August 2007]

**Summary.** On-line auctions pose many challenges for the empirical researcher, one of which is the effective and reliable modelling of price paths. We propose a novel way of modelling price paths in eBay's on-line auctions by using functional data analysis. One of the practical challenges is that the functional objects are sampled only very sparsely and unevenly. Most approaches rely on smoothing to recover the underlying functional object from the data, which can be difficult if the data are irregularly distributed. We present a new approach that can overcome this challenge. The approach is based on the ideas of mixed models. Specifically, we propose a semiparametric mixed model with boosting to recover the functional object. As well as being able to handle sparse and unevenly distributed data, the model also results in conceptually more meaningful functional objects. In particular, we motivate our method within the framework of eBay's on-line auctions. On-line auctions produce monotonic increasing price curves that are often correlated across auctions. The semiparametric mixed model accounts for this correlation in a parsimonious way. It also manages to capture the underlying monotonic trend in the data without imposing model constraints. Our application shows that the resulting functional objects are conceptually more appealing. Moreover, when used to forecast the outcome of an on-line auction, our approach also results in more accurate price predictions compared with standard approaches. We illustrate our model on a set of 183 closed auctions for Palm M515 personal digital assistants.

**Keywords.** Boosting; eBay; Mixed model; Non-parametric methods; On-line auction; Penalized splines; Smoothing

1. **Introduction**

On-line auctions pose many challenges for the empirical researcher, many of which are due to the technological advancements in measurement, collection and storage of data, which have
led to increasingly complex data structures. Examples include measurements of individuals’ behaviour over time, digitized two- or three-dimensional images of the brain and recordings of three- or even four-dimensional movements of objects travelling through space and time. Another example is the price path during an on-line auction. Such data, although recorded in a discrete fashion, are usually thought of as continuous objects represented by functional relationships. This gives rise to functional data analysis. In functional data analysis (Ramsay and Silverman, 2002, 2005) the centre of interest is a set of curves, shapes, objects or, more generally, a set of functional observations. This is in contrast with classical statistics where the interest centres on a set of data vectors. In that sense, functional data are not only different from the data structure that is studied in classical statistics, but also it actually generalizes it. Many of these new data structures call for new statistical methods to unveil the information that they carry.

Any set of functional data consists of a collection of continuous functional objects such as a set of continuous curves describing the changes in temperature over the course of a year or the price increase in an on-line auction. Despite their continuous nature, limitations in human perception and measurement capabilities allow us to observe these curves only at discrete time points. Thus, the first step in a typical functional data analysis is to recover, from the observed data, the underlying continuous functional object. This recovery is typically done with the help of smoothing methods.

When recovering the functional object, we encounter a variety of challenges, two of which are sparse and unevenly distributed data. Smoothing methods often operate locally, which means that sparse data can lead to curves that are very unrepresentative of the underlying functional object. The problem of sparse and unevenly distributed data is very acute since increasingly more real world processes generate such types of data. One example is on-line auctions where the arrival of data is determined by many different sources that act independently of one another: sellers who decide when to start and stop the auction, or bidders who decide when and where to place their bids. The situation is similar in Web logs (‘blogs’) where the arrival of new postings depends on the arrival (and the importance) of news. Similarly, information on a patient’s medical status becomes available only when the patient decides to visit a doctor. Either way, the result is irregularly spaced data which pose a challenge to traditional smoothing methods.

It is important to obtain accurate representations of the underlying functional object. Just as measurement error leads to an (unwanted) source of variation in classical statistics, poor curve representation can lead to an (additional) error source in functional data analysis. Moreover, in functional data analysis one often analyses derivatives of the functional objects to study, say, the dynamics of a process (Jank and Shmueli, 2007a). Then, if the curve already contains error, this error will be propagated (and magnified) to the curve derivative. Another important area is curve forecasting to obtain dynamic, realtime predictions of on-line auctions (Wang et al., 2007; Jank et al., 2006). There, if the functional object is poorly represented, then the prediction (together with the ensuing conclusions) can be far off. In this paper we propose a method that can overcome sparse and unevenly distributed data by borrowing information from neighbouring functional objects. The underlying idea is very similar to that of mixed models (McCulloch and Searle, 2001). A related idea in that context is that of James and Sugar (2003) who considered clustering of sparsely sampled functional objects.

One additional advantage of our modelling approach is that it results in conceptually more meaningful functional objects compared with previous approaches. Much of the extant literature that studies on-line auctions assumes independence between auctions (Lucking-Reiley et al., 1999; Kauffman and Wood, 2003; Bapna et al., 2003, 2005; Roth and Ockenfels, 2002).
This assumption, however, is difficult to justify from a practical point of view given that it is very easy for a bidder to monitor 10 or more auctions simultaneously. For instance, if a bidder participates in two auctions simultaneously, then prices in these two auctions are no longer independent of one another. Also, the independence assumption implies that two auctions for the same (or similar) items transacting during the same period of time have no effect on one another (see Jank and Shmueli (2007b) for evidence against this assumption). This assumption is typically not made out of ignorance of the fact, but rather because of the lack of models that are sufficiently flexible to account for the different types of correlation structure that arise. Clearly, there is room for statistical thought and innovation. The method that is proposed in this paper is one attempt in that direction.

We focus here on methods that can overcome irregularly spaced data and that can also incorporate dependences between functional objects. These methods are derived from the mixed regression model framework. In the context of regression models, much work has been done to extend the strict parametric form to include more flexible semiparametric and non-parametric approaches. For details see Hastie and Tibshirani (1990), Green and Silverman (1994) or Schimek (2000). For example, the $P$-spline (e.g. Eilers and Marx (1996)) is very versatile and requires only an a priori decision about a few basic smoothing parameter settings such as the location and number of knots, the order of the spline and the magnitude of the smoothing penalty. However, one of the disadvantages of $P$-splines (and also of other smoothing methods) is the manual (or semimanual) selection of the smoothing parameters. Another disadvantage is that they require relatively large sample sizes to produce reliable results. Furthermore, not only the sample size is important but also the sample variability. For instance, if we wish to estimate a function over a particular region, then the results that are returned by $P$-splines can be very poor if that region is sampled only very locally. Thus, traditional smoothing methods can be problematic if the data are sparse and very unevenly distributed.

We overcome this problem by using semiparametric mixed regression models. Our boosting approach also results in automated selection of the smoothing parameters. Our approach is different from standard mixed (or random-) regression models in that we not only allow random variations around a common (population) slope but also random variations around a population slope function. The resulting model is thus a hybrid between a penalized smoothing spline and a mixed regression model, which allows an enormous amount of flexibility for the modeller.

This paper is organized as follows. In Section 2 we review the basics of eBay’s auction mechanism and describe the data challenges that it produces. In Section 3 we describe two approaches for modelling sparse and unevenly spaced data. The first approach is the more traditional approach that is based on penalized smoothing splines, and we demonstrate situations in which it becomes unreliable. The second approach uses the ideas of mixed models. We describe the general semiparametric mixed model for estimating sparse and unevenly spaced data and describe boosting strategies to estimate the model parameters. We apply the method to a set of eBay auctions in Section 4. In Section 5, we compare our approach with a random-regression model and discuss advantages and disadvantages. We conclude with final remarks and future directions.

The data that are analysed can be obtained from

http://www.blackwellpublishing.com/rss

2. Recovering the price curve in on-line auctions

In what follows we motivate the problem of recovering sparse and unevenly sampled curves by considering eBay’s on-line auctions (see http://www.ebay.com). We describe eBay’s auction
mechanism, the data that it generates and the challenges that are involved in taking a functional approach to analysing on-line auction data.

2.1. eBay’s auction mechanism

eBay is one of the biggest and most popular on-line marketplaces. In 2005, eBay had 180.6 million registered users, of whom over 76.8 million bid, bought or sold an item, resulting in over 1.9 billion listings for the year. Part of its success can be attributed to the way in which items are sold on eBay. The dominant form of sale is the auction and eBay’s auction format is a variant of the second-price sealed bid auction (‘Vickrey auctions’; see for example Krishna (2002)) 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. For example, suppose that bidder A is the first bidder to submit a proxy bid on an item with a minimum bid of $10 and a minimum bid increment of $0.50. Suppose that bidder A places a proxy bid of $25. Then eBay’s Web page automatically displays A as the highest bidder, with a bid of $10. Next, suppose that bidder B enters the auction with a proxy bid of $13. eBay still displays A as the highest bidder; however, it raises the displayed high bid to $13.50, one bid increment above the second-highest bid. If another bidder submits a proxy bid above $25.50, bidder A is no longer in the lead. However, if bidder A wishes, he or she can submit a new proxy bid. This process continues until the auction ends. Unlike some other auctions, eBay has strict ending times, ranging between 1 and 10 days from the opening of the auction, as determined by the seller.

2.2. eBay’s data

eBay is a rich source of high quality—and publicly available—bidding data. eBay posts complete bid histories of closed auctions for at least 15 days on its Web site. See http://listings.ebay.com/pool1/listings/list/completed.html. One implication of this is that eBay data do not arrive in the traditional form of tables or spreadsheets; rather, they arrive in the form of hypertext pages.

Fig. 1 shows an example of eBay’s auction data. Fig. 1(a) displays a summary of the auction attributes such as information about the item for sale, the seller, the opening bid, the duration of the auction and the winner. Fig. 1(b) displays the bid history, i.e. the temporal sequence of bids placed by the individual bidders. Fig. 2 shows the scatter of these bids over the duration of the auction (a 7-day auction in this example). We can see that only six bids were placed in this auction and that most bids were placed towards the end of the auction, with the earlier part of the auction only receiving one bid. If we conceptualize the evolution of price as a continuous curve between the start and the end of the auction, then Fig. 2 shows an example of a very sparse and unevenly sampled price curve.

2.3. Price curve and data challenges

Studying and modelling the price curve can help in finding answers to questions such as ‘Does price in a typical on-line auction increase sharply at first and then level off towards the end?’ Or, conversely, ‘Does price remain low throughout most of the auction only to experience sharp increases at the end?’ And, if so, ‘Is this price pattern the same for auctions of all types? Or do patterns differ between different product categories?’ Jank and Shmueli (2007a) showed that answers to these questions can help in characterizing auction dynamics and lead to more
Fig. 1. Bid history for a completed eBay auction: (a) auction attributes and information on the format of the auction, the seller and the item that was sold; (b) detailed history of the bidders and their bids.
One way of modelling the price curve is via functional models. However, this modelling task is complicated owing to the data structure that is found in on-line auctions. Consider again the example in Fig. 2. The first step in functional data analysis is to recover, from the observed bids, the continuous price curve. Note, however, that only six bids are observed, most of them at the end of the auction. Using traditional smoothing methods to recover a continuous curve from only six data points of which five are located at the end is not particularly meaningful or reasonable and will not lead to very representative estimates of the price curve.

3. Modelling sparse and unevenly sampled data

One solution is to borrow information from other similar auctions. Fig. 3 shows the bid histories for three similar auctions for the same item, labelled 6, 121 and 173. We can see that the price curve in auction 6 is only sampled at the end. Conversely, in auction 121 the price is sampled predominantly at the beginning, with no information from the middle of the auction. And finally, auction 173 contains plenty of price information from the middle of the auction but only little from its start and end. Although every auction by itself contains only partial information about the entire price curve, if we combine the information from all three auctions, we obtain a more complete picture. This is shown in Fig. 3(d). The idea of semiparametric mixed model smoothing is now as follows: whenever an individual auction contains incomplete information, we borrow from the combined information of all similar auctions. The resulting approach is different from standard mixed (or random-) regression models in that we not only allow random
variations around a common (population) slope but also random variations around a flexible population slope function. Our model is thus a hybrid between a penalized smoothing spline and a mixed regression model and it thus achieves an enormous amount of flexibility. We describe this method more formally next.

3.1. Penalized splines: the challenge

The basic model for price that we consider has the form

$$\text{Price}_i(t) = \alpha_{i0} + \alpha_{i}(t) + \varepsilon_i(t), \quad (1)$$

where $t$ stands for time and $\text{Price}_i(t)$ denotes the price of the $i$th auction at time $t$. (We can also model log(Price) for auctions that have extreme price jumps, e.g. Jank and Shmueli (2007a).) $\alpha_{i0}$ is the intercept, $\alpha_{i}(t)$ denotes a suitable function of time, centred at zero for identifiability reasons, and $\varepsilon_i(t)$ is a noise process with zero mean and variance $\sigma^2$.

A common approach for obtaining estimates of $\alpha_{i}(t)$ is via basis function expansion. One of the simplest basis functions is the truncated power series basis of degree $d$, which yields

$$\alpha_{i}(t) = \gamma_0^{(i)} + \gamma_1^{(i)} t + \ldots + \gamma_d^{(i)} t^d + \sum_{s=1}^{M} \alpha_s^{(i)} (t-k_s)_+, \quad (2)$$

where $k_1 < \ldots < k_M$ are distinct knots and $\gamma_j^{(i)}$ and $\alpha_s^{(i)}$ are parameters to be estimated from the data. More generally, one uses a function of the form

$$\alpha_{i}(t) = \sum_{m=1}^{M} \phi_{m}^{(i)}(t^*) = \phi^T(t^*) \alpha_i \quad (2)$$
where \( \phi^{(i)}_n \) denotes the \( n \)th basis function, \( \phi^T_i(t) = (\phi^{(i)}_1(t), \ldots, \phi^{(i)}_M(t)) \) and \( \alpha^T_i = (\alpha^{(i)}_1, \ldots, \alpha^{(i)}_M) \) are unknown parameters.

Let the data be given by the pairs \((y_{is}, t_{is})\), \(i = 1, \ldots, n\), \(s = 1, \ldots, S_i\), where \(y_{is}\) is the price of auction \(i\) at bid number \(s\) which occurs at time \(t_{is}\). The number of bids and their timing varies across auctions. The additive model that we consider has the general form

\[
y_{is} = \alpha^{(i)0} + \alpha^{(i)}(t_{is}) + \epsilon_{is}, \quad i = 1, \ldots, n, \quad s = 1, \ldots, S_i,
\]

where \(E(\epsilon_{is}) = 0\) and \(\text{var}(\epsilon_{is}) = \sigma^2_i\). This approach models each auction separately, resulting in \(n\) different function estimates \(\hat{\alpha}^{(i)}(\cdot)\) and \(n\) different parameter estimates \(\hat{\alpha}^{(i)0}, i = 1, \ldots, n\). For semiparametric and non-parametric regression models, Marx and Eilers (1998) proposed the \(B\)-splines which have also been used by Hastie and Tibshirani (2000) and Wood (2004). For further properties of basis functions see also Wand (2000) and Ruppert and Carroll (1999).

Using equation (2) and writing \(x^T_{is} = (1, \phi^T(t_{is}))\) and \(\delta^T_i = (\alpha^{(i)0}, \alpha^T_i)\), model (3) becomes

\[
y_{is} = \alpha^{(i)0} + \phi(t_{is})^T \alpha_i + \epsilon_{is} = x^T_{is} \delta_i + \epsilon_{is},
\]

or in matrix form

\[
y_i = \alpha^{(i)0} + \Phi^T_i \alpha_i + \epsilon_i = X_i \delta_i + \epsilon_i,
\]

with \(y^T_{i} = (y_{i1}, \ldots, y_{iS_i})\), \(\Phi_i\) has rows \(\phi_i(t_{is})\), \(\epsilon^T_i = (\epsilon_{i1}, \ldots, \epsilon_{iS_i})\) and \(X_i = (1, \Phi_i)\). Estimates for \(\alpha_i\) may be based on the penalized log-likelihood for auction \(i\),

\[
l_p^{(i)}(\delta_i) = -\frac{1}{2\sigma_i^2} (y_i - X_i \delta_i)^T (y_i - X_i \delta_i) - \frac{1}{2} \lambda \delta^T_i K_i \delta_i,
\]

where \(\lambda \delta^T_i K_i \delta_i\) is a penalty term which penalizes the coefficients \(\alpha_i\). For the truncated power series an appropriate penalty is given by

\[
K = \text{diag}(0, I)
\]

where \(I\) denotes the identity matrix and \(\lambda\) determines the smoothness of the function \(\alpha\). For \(\lambda \to \infty\), a polynomial of degree \(d\) is fitted. \(P\)-splines use \(K_i = D_i^T D_i\) where \(D_i\) is a matrix of the difference between adjacent parameters, yielding the penalty \(\lambda \alpha_i^T K_i \alpha_i\).

From the derivative of \(l_p^{(i)}(\delta_i)\) we obtain the estimation equation \(\partial l_p^{(i)}(\delta_i)/\partial \delta_i = 0\), which yields

\[
\hat{\delta}_i = \left( \frac{1}{\sigma_i^2} X^T_i X_i + \lambda K_i \right)^{-1} \frac{1}{\sigma_i^2} X^T_i y_i.
\]

The tuning parameter \(\lambda\) may be optimized by using the generalized cross-validation criterion that was described in Wood (2000).

Fig. 4 illustrates the performance of the penalized smoothing spline for four sample auctions. For each auction, we investigate four different smoothing scenarios: a low order (grey curve) versus a high order (black curve) smoothing spline (i.e. second order versus fourth order), coupled with a low (full curve) versus a high (broken curve) smoothing parameter (\(\lambda = 0.1\) versus \(\lambda = 1\)). We chose four very representative auctions out of the set of all 183 auctions.

We can see in Fig. 4(a) that for auction 51 all four smoothers deliver very similar curves, which all approximate the observed data very well. In contrast, for auction 121 (Fig. 4(b)), the performance of the four smoothers differs greatly, especially in the middle of the auction (between days 3 and 6) where no observations are available. Moreover, note that the higher order smoother (coupled with the lower penalty term) results in a locally variable curve where
Fig. 4. Performance of penalized splines by using various smoothing parameters (○, actual live bids observed during the auction; ——, order 2, penalty 0.1; ——, order 2, penalty 1; ——, order 4, penalty 0.1; ——, order 4, penalty 1): (a) auction 51; (b) auction 121; (c) auction 141; (d) auction 165
the curve increases up to day 3 but then decreases to day 6. This curve decrease is difficult to justify from a conceptual point of view since auction prices, by nature of the ascending auction mechanism, should be monotonically increasing. In fact, note that the actual observations do increase over the same time period. However, the sparsity of the data between day 3 and day 6 causes the high order–low penalty smoothing spline to exhibit too much local variability. Consequently, it appears as if the lower order spline (together with the higher penalty term) is the better choice for auction 121, at least from a conceptual viewpoint. Now consider auction 141 (Fig. 4(c)), which has a strong surge in price at the end of the auction. Whereas the price changes only a little throughout most of the auction, it jumps dramatically during the last day. Not surprisingly, only the smoother with the highest flexibility (order 4 and $\lambda = 0.1$) manages to capture this last moment surge in price activity. Although the other smoothers produce a reasonable approximation for most of the duration of the auction, they all fail at the end of the auction owing to the high bidding intensity. And finally, consider auction 165 (Fig. 4(d)). Interestingly, for this auction all four smoothers vary quite significantly in their fit and, more importantly, none captures the price activity at the last moments of the auction.

In summary, although in some cases smoothing splines can produce very reasonable functional objects regardless of the smoothing parameters, in other cases the choice of the parameters can have a significant effect. In particular, whereas some data scenarios call for smoother objects of lower order and higher penalty term, other scenarios require more flexible objects of higher order. And yet, the data challenges that are presented in on-line auctions are so vast that even four different smoothers are not sufficient for accounting for all scenarios as seen in auction 165 above. There are alternative types of smoother that may offer some relief such as monotone smoothing splines (Ramsay, 1998); however, they can be more expensive to compute. In our applications of the mixed model approach to these auction data, all estimates turned out to be monotonic without using monotonic smoothers, i.e. without explicitly enforcing monotonicity. We describe that approach (and our findings) next.

### 3.2. A solution: semiparametric mixed models

Mixed models have been around in the statistics literature for quite a while. Yet, to date, they have found only little use in the context of functional data analysis. The basic idea of mixed effect models (or random-effect models) is the presence of several data clusters and repeat observations within each cluster (see for example Henderson (1953), Laird and Ware (1982) and Harville (1977)). Overviews including more recent work can be found in Verbeke and Molenberghs (2001). Concepts for estimating semiparametric mixed models with an implicit estimation of the smoothing parameters are described in Verbyla et al. (1999), Parise et al. (2001), Lin and Zhang (1999), Brumback and Rice (1998), Zhang et al. (1998) and Wand (2003). Bayesian approaches have been considered by, for example, Fahrmeir and Lang (2001). A different concept is the use of boosting techniques. Boosting allows fitting of additive models with many covariates. One of the major advantages of boosting is the automated selection of the smoothing parameters. Moreover, boosting techniques may be used to incorporate subject-specific variation of smooth influence functions by specifying ‘random slopes’ on smooth effects. This results in flexible semiparametric mixed models which are appropriate in cases where a simple random intercept cannot capture the variation of effects across subjects.

Recall that in equation (1) we model each auction separately, assuming independence across all $n$ auctions. A more parsimonious (and conceptually more appealing) approach is based on semiparametric mixed model methodology. Assume that the price curve is modelled as

$$\text{Price}_i(t) = \alpha_0 + \alpha(t) + b_{i0} + \varepsilon_i(t),$$  \hspace{1cm} (7)
where $b_{i0}$ is a random effect with $b_{i0} \sim N(0, \sigma^2_{b})$ and $\alpha_0$ is the (fixed) intercept for the model. Note that in equation (7) we assume a common slope function $\alpha(t)$ for all auctions. We also assume that the intercepts of all auctions vary randomly with mean $\alpha_0$ and variance $\sigma^2_{b}$. The residual error $\varepsilon_i$ is assumed to be $N(0, \sigma^2_{R})$, where $R$ is a known residual structure, e.g. auto-regressive first order. (For simplicity we use an independence structure in this application.) In that sense, all auctions are conditionally independent given the level (the random intercept). In this fashion, we can model all auctions within one parsimonious model; yet, we obtain a price curve estimate for each auction individually.

We can obtain further modelling flexibility by also assuming random-slope functions. In other words, we assume that there is a common underlying slope which is a random variable, and that each individual auction draws a realization from this random-slope function. For this, we extend the model by using flexible splines of the form

$$\text{Price}_i(t) = \alpha_0 + \alpha(t) + b_{i0} + b_{i1} \alpha(t) + \varepsilon_i(t)$$

(8)

where $b_{i0}$ and $b_{i1}$ are again random effects with $b := (b_{i0}, b_{i1}) \sim N(0, Q)$. The model implies a common intercept and slope function for all auctions. Individual heterogeneity is induced by an auction-specific random intercept $b_{i0}$ and an auction-specific random ‘slope’. (The model formulation is reminiscent of interaction terms: if we think about a parametric alternative to the term $\alpha(t)$, e.g. $\alpha \times t$, then the corresponding parametric equivalent of $b_{i1} \alpha(t)$ would be the interaction term that is given by $\alpha \times b_{i1} \times t$. In other words, $b_{i1}$ modifies the common slope $\alpha$ by a random amount.) The covariance $Q$ can be parameterized by $\rho$, a vector of parameters to be optimized. For an unstructured covariance matrix with elements $q_{11}, q_{21}$ and $q_{22}$,

$$Q(\rho) = \begin{pmatrix} q_{11} & q_{21} \\ q_{21} & q_{22} \end{pmatrix},$$

the vector $\rho$ is then the set of the elements in the lower triangular matrix of the Cholesky root $Q^{1/2}$. In a more technical sense, $\rho$ is the symmetric diagonal operator of $Q^{1/2}$.

What we obtain via model (8) is an estimated price curve for auction $i$ that is characterized by the level $b_{i0}$, the common slope $\alpha(t)$ and the auction-specific modification $b_{i1} \alpha(t)$. Model (8) can be regarded as a restricted version of equation (1) using the information of the other auctions in the form

$$\alpha_{(i)}(t) \approx b_{i0} + \alpha(t) + b_{i1} \alpha(t).$$

A few comments on our model specification are in order. Our modelling approach defines an individual slope on the smooth underlying function. This has strong conceptual advantages in monotonic settings, as is the case here, but also in other settings. Take, for example, a sine function. In that case, the random slope allows some of the individual functions to be very non-linear whereas others can be totally flat. This cannot be modelled by a sum of a spline function and a random linear time effect. The same occurs, although to a lesser extent, in the monotonic case. Individual flat functions cannot be fitted unless the monotonic function itself is linear. The limitation of a spline population model with random effects is therefore that, even if quadratic (or higher order polynomial) effects are included, the individual functions are restricted to linear and quadratic deviations from the population mean. In the auction context, this disadvantage is key, because of the nature of the price curves: auction dynamics are not well approximated by additive quadratic time effects.

Another advantage of our model formulation is that the random-spline model allows a simple interpretation: the common spline function varies across auctions in the steepness of the bid-
ding process and the initial bidding structure. So the auction dynamics are completely specified by the common spline function, the random intercept and the modification to the common spline. Of course, the random slope on the population function is also restrictive, assuming a distinct structure in the underlying model. But it is very simple and allows simple interpretation of estimated effects.

In what follows, we describe an estimation approach via boosting. Boosting allows us to estimate jointly not only the model parameters but also the smoothing terms. Moreover, boosting as a stepwise procedure allows us to include multiplicative effects, which is not possible by using restricted maximum likelihood.

### 3.3. Boosting and the mixed model approach

Boosting originates in the machine learning community where it has been proposed as a technique for improving classification procedures by combining estimates with reweighted observations. Since it has been shown that reweighting corresponds to minimizing iteratively a loss function (Breiman, 1999; Friedman, 2001), boosting has been extended to regression problems in an $L_2$-estimation framework by Bühlmann and Yu (2003). In what follows, boosting is used to obtain estimates for the semiparametric mixed model. Instead of using restricted maximum likelihood estimates for the choice of smoothing parameters (Wand, 2000; Ruppert et al., 2003), the estimates of the smooth components are obtained by using ‘weak learners’ iteratively. For observations $(y_{is}, t_{is})$, $i = 1, \ldots, n$, $s = 1, \ldots, S_i$, we write

$$y_{is} = \alpha_0 + \phi(t_{is})^T \alpha + b_{i0} + b_{i1} \phi(t_{is})^T \alpha + \varepsilon_{is},$$

or, in familiar mixed model matrix form,

$$y_i = X_i \delta + Z_i b + \varepsilon_i,$$

where

$$\begin{pmatrix} b \\ \varepsilon_i \end{pmatrix} \sim N \left\{ 0, \begin{pmatrix} Q(\rho) & 0 \\ 0 & \sigma^2 \rho \end{pmatrix} \right\} ,$$

and where we write $X_i = (1, \Phi_i)$, $\delta^T = (\alpha_0, \alpha)$ and $Z_i = (1, \Phi_i \alpha)$. Let $V_i = V_i(\sigma^2, \rho)$ denote the covariance matrix of the marginal model $V_i = Z_i Q(\rho) Z_i^T + \sigma^2 \rho$. Penalizing $(\alpha_0, \alpha)$ by $\delta$ is based on the penalty matrix which for the truncated power series has the form $K = \text{diag}(0, \lambda I)$.

The weak learner for $\delta$ is based on an initially fixed and very large smoothing parameter $\lambda$. By iteratively fitting the residuals, the procedure adapts automatically to the possibly varying smoothness of the individual components. The algorithm is initialized by using an appropriate weak learner. The basic concept in boosting is that in one step the refitting of $\alpha(t_{is})$ is done by using a weak learner which in our case corresponds to large and fixed $\lambda$ in the penalization term.

The algorithm works in the following way. Let $\eta^{(r-1)}_i$ denote the estimate from the previous step. Then the refitting of residuals (without selection) is done by fitting the model

$$y_i - \eta^{(r-1)}_i \sim N \left\{ \eta_i, V_i(\theta) \right\}$$

with

$$\eta_i = 1 \alpha_0 + \Phi_i \alpha + (1, \Phi_i \hat{\alpha}^{(r-1)}) \begin{pmatrix} b_{i0} \\ b_{i1} \end{pmatrix}$$

(9)

where $\alpha_0$ and $\alpha$ are the parameters to be estimated and $\hat{\alpha}^{(r-1)}$ is known from the previous step. Using the resulting estimates $\hat{\alpha}_0$ and $\hat{\alpha}$, the next update takes the form
\begin{align*}
\hat{\alpha}^{(r)} &= \hat{\alpha}^{(r-1)} + \alpha_s \\
\hat{\alpha}_0^{(r)} &= \hat{\alpha}_0^{(r-1)} + \alpha_0.
\end{align*}

The algorithm is stopped if enough complexity is in the model. Since boosting is an iterative way of fitting data the complexity of the model increases from step to step. In the beginning we fit a very robust model which adapts stepwise to the data. The complexity of the model is measured via the Bayes information criterion BIC. Therefore in every boosting step the projection matrix of \( y_i \) on the new estimates \( \alpha^{(r)} \) and \( \alpha_0^{(r)} \) is computed. Then, the trace of this matrix is used to compute the BIC in the \( r \)th step by setting

\[ \text{BIC}^{(r)} = -2 l(\hat{\alpha}_0^{(r)}, \hat{\alpha}^{(r)}) + \text{df} \log(n), \]

where df is the trace of the projection matrix, \( l(\hat{\alpha}_0^{(r)}, \hat{\alpha}^{(r)}) \) is the log-likelihood in the \( r \)th step and \( n \) is the number of different auctions in the data set.

The basic idea behind the refitting is that forward iterative fitting procedures like boosting are weak learners. In that sense, the previous estimate is always considered to be known in the last term of equation (9). Of course, in every step the variance components corresponding to \( (b_{0i}, b_{1i}) \) must be re-estimated. For the complete algorithmic detail see Appendix A.

We want to point out that in the present algorithm monotonicity is not enforced. However, note that monotonicity can be easily incorporated in the algorithm as follows. If the basis functions in \( \phi_i \) are chosen as B-splines, then a sufficient condition for monotonicity is that the coefficients \( \alpha^T_i = (\alpha_{i1}, \ldots, \alpha_{im}) \) in the product \( \phi^T_i (t) \alpha_i \) fulfill \( \alpha_{i,j+1} \geq \alpha_{i,j} \) for all \( j \). This property of B-splines has been exploited by Kelly and Rice (1990) and Leitenstorfer and Tutz (2007). The latter used a boosting technique that could also be applied here. The basic idea is to split the coefficients into two parts, e.g. \( (\alpha_{i1}, \ldots, \alpha_{ir}) \) and \( (\alpha_{i,r+1}, \ldots, \alpha_{im}) \), and then to update the blocks together in boosting steps. For simplicity, we give the boosting algorithm here without monotonicity constraints (but see Leitenstorfer and Tutz (2007) for more details).

4. Application to eBay’s price evolution

4.1. Data description

Our data consist of 183 closed auctions for Palm M515 personal digital assistants that took place between March 14th and May 25th of 2003. In an effort to reduce as many external sources of variability as possible, we included data on only 7-day auctions, transacted in US dollars, for completely new (not used) items with no added-on features, and where the seller did not set a secret reserve price. These data are publicly available at [http://www.smith.umd.edu/ceme/statistics/](http://www.smith.umd.edu/ceme/statistics/).

The data for each auction include its opening price, closing price and the entire series of bids (bid amounts and timestamps) that were placed during the auction. This information is found in the bid history, as shown in Fig. 1.

Note that the series of bids that appear in the bid history are not the actual prices that are shown by eBay during the live auction; rather, they are the proxy bids that were placed by individual bidders (which become available only after the auction closes). eBay uses a second-price mechanism, where the highest bidder wins but pays only the second highest bid. Therefore, at each point in time, the price that is displayed during the live auction is the second highest bid. For this reason, we converted the bids into ‘current price’ values that capture the evolution of price during the live auction. Note that the current price data are indeed monotone increasing.
This adds the extra requirement on our smoothing method that the recovered functional object be monotone. Only a few standard smoothing methods meet this requirement (Ramsay, 1998); moreover, monotonicity constraints typically also increase the computational complexity of the smoother. In what follows we show that our approach, without explicitly adding any such constraints, automatically estimates the underlying monotonicity from the pooled data and imposes it on each auction estimate individually.

4.2. Model fit

We fit the following mixed effects model to all 183 auctions:

\[ s(\text{Price}_{is}) = \alpha_0 + \alpha(t_{is}) + b_{i0} + b_{i1} \alpha(t_{is}) + \varepsilon_{is}, \]

where \( s(\cdot) \) is the square root. Note that \( \alpha_0 \) and \( \alpha(t) \) denote again the intercept and slope func-

![Fig. 5. Mean spline function (-----) and ±1 standard pointwise confidence band (--------), estimated across all 183 auctions: we use 18 equally spaced knots (note that there is no smoothing parameter since we achieve smoothness via boosting)](image)

<table>
<thead>
<tr>
<th>( b_0 )</th>
<th>( b_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_0 )</td>
<td>4.536 (1)</td>
</tr>
<tr>
<td>( b_1 )</td>
<td>-0.619 (−0.847)</td>
</tr>
</tbody>
</table>

†The correlation is given in parentheses.
tion, which are common across all auctions. The random effects $b_{i0}$ and $b_{i1}$ capture auction individual variation. We estimate the parameters $\alpha_0$, $\alpha(t)$, $q_{11}$, $q_{21}$, $q_{22}$ and $\sigma^2$ by using the algorithm ‘BoostMixed’ that is outlined Appendix A. Fig. 5 and Table 1 show the results.

Table 1 shows the estimated covariance matrix for the random intercept and slope. We can see that there is a strong negative correlation between the intercept and the slope. This implies that the system has a tendency to self-correct: auctions that are high initially (large intercept) will receive a smaller slope and vice versa.

Fig. 5 shows the estimated mean (together with pointwise confidence bounds) of $\alpha(t)$. Recall that $\alpha(t)$ is the mean slope function, which is common to all 183 auctions. The mean, as expected, is monotone, showing two inflection points around days 1 and 5. Note that the mean slope is monotonically increasing, as expected from an ascending auction. Moreover, the slope is steepest at the beginning and at the end of the auction, which is consistent with the phenomena of early bidding and bid sniping that are observed in the on-line auction literature (Bapna et al., 2003; Shmueli et al., 2007). It is also intriguing that, in contrast with most smoothers, the width of the confidence interval is decreasing towards the end. This is because most bids arrive towards the end of the auction, thus providing greater accuracy in the estimation of the population trend.

Figs 6 and 7 show the resulting curve estimates for 24 auctions. The full curves correspond to the mixed model fit; the broken curves correspond to the ordinary penalized smoothing spline fit. We can see that the penalized smoothing splines can result in poor curve representations: in some auctions there is a lack of curvature (e.g. auctions 3, 15 and 27), whereas in others there is excess curvature (e.g. auctions 23, 24 and 34), yet in other auctions they do not produce any estimates at all owing to data sparseness (e.g. auctions 16 and 35), or data unevenness may result in very unrepresentative curves (e.g. auctions 23, 24 and 34). Moreover, many of the curves that are produced by penalized splines are unsatisfactory from a conceptual point of view: for instance, in auctions 4, 24 or 34, penalized splines result in an estimated price path that is not strictly monotonic increasing, which violates the assumption underlying ascending auction formats.

This is very different for the estimates that are produced by the mixed model approach. Mixed model smoothing takes the mean slope as a blueprint for all auctions and allows for variation from the mean through the random effect $b$. Indeed, although the full curves in Figs 6 and 7 all resemble the mean slope, they differ in steepness and the timing of early and late bidding. These differences from the mean are driven by the amount (and distribution) of the observed data. For instance, auction 10 has a considerable number of bids that are distributed evenly across the entire length of the auction. As a result, the estimated curve is quite different from the mean slope function. In contrast, auction 16 has only one observation. Whereas penalized smoothers break down with only so little information available (and do not produce any curve estimates at all), the mixed model approach can still produce a reliable (and conceptually meaningful) result by borrowing information from the mean slope.

All in all, our results show that a penalized spline can sometimes do better than our mixed model. However, the times that a penalized spline does better, its improvements are typically only slight (e.g. in auctions 2 and 4). However, when the penalized spline fails, it can fail quite miserably (e.g. in auctions 23 and 34). Moreover, for some auctions it does not produce any curve estimates at all (e.g. auctions 16 and 35). So, in some sense the mixed model approach provides peace of mind in that, although it may not produce optimal results across all auctions, it provides a good curve approximation throughout. This is especially important in large databases where (visual) inspection of all individual auctions is difficult or impossible.

Moreover, the quality of the model is determined not only by the number of bids but also by their distribution over the duration of the auction. Indeed, in instances where there are sufficient bids and where the bids are evenly distributed across the entire auction, penalized splines can
Fig. 6. Smoothed time—12 auctions with their specific behaviour regarding price and time (----, mixed model approach; - - - - -, separately fitted penalized splines): (a) auction 1; (b) auction 2; (c) auction 3; (d) auction 4; (e) auction 7; (f) auction 8; (g) auction 9; (h) auction 10; (i) auction 13; (j) auction 14; (k) auction 15; (l) auction 16
Fig. 7. Smoothed time—another 12 auctions with their specific behaviour regarding price and time (——, mixed model approach; - - - - -, separately fitted penalized splines): (a) auction 21; (b) auction 22; (c) auction 23; (d) auction 24; (e) auction 27; (f) auction 28; (g) auction 29; (h) auction 30; (i) auction 33; (j) auction 34; (k) auction 35; (l) auction 36
fit individual auctions better than our mixed model. However, in sparse or unevenly distributed cases (e.g. auctions 16, 23, 24, 34 and 35), penalized splines can break down.

It is interesting that all the price curves that are created by the mixed model approach are monotonically increasing. This is intriguing since the mixed model that we implemented here does not incorporate any monotonicity constraints. However, it appears to have ‘learned’ this feature from the pooled data. This makes it a very flexible and powerful approach, which is suitable for many different data scenarios.

We want to re-emphasize that our model formulation does not enforce (or guarantee) monotonicity. Monotonicity is merely a (fortunate) result of this particular kind of model (which models the trend in the population with random variation around the trend) applied to this particular kind of data (whose population trend happens to be monotonic). Our excitement though stems from the fact that the result is much different from what we obtain by using individual smoothing splines (which do not recover monotonicity at all, despite the underlying monotonic population trend in the data). Moreover, we pointed out earlier that monotonicity constraints could be included in the boosting step without too much extra effort (see also Leitenstorfer and Tutz (2007)).

4.3. Forecasting with the mixed model

Another way to evaluate the quality of a smoother is via its ability to forecast the continuation of the curve. Specifically, in the auction setting we are interested in how well the estimated price curve can predict the final price of an auction. Price predictions for on-line auctions are becoming an increasingly important topic (Wang et al., 2007; Jank et al., 2006; Ghani, 2005; Ghani and Simmons, 2004). On eBay, an identical (or nearly identical) product is often sold in numerous, often simultaneous auctions. For instance, a simple search under the keywords ‘iPod shuffle 512MB MP3 player’ returned over 300 hits for auctions that closed within the next 7 days. A more general search under the less restrictive keywords ‘iPod MP3 player’ returned over 3000 hits. Clearly, it would be challenging, even for a very dedicated eBay user, to make a purchasing decision that takes into account all of these 3000 auctions. The decision-making process can be supported via price forecasts. Given a method to predict the outcome of an auction ahead of time, one could create an auction ranking (from lowest to highest predicted price) and select only those auctions for further inspection with the lowest predicted price. In what follows we investigate the ability of the mixed model approach to predict the final price of an auction.

We do this in the following way. We split each auction into a training set and a validation set. Specifically, we assume that the first two-thirds of the auction are observed; we estimate our model on the price that is observed during this time interval. Then, using the estimated model, we investigate how well it predicts price for the last third of the auction, i.e. for the validation set. In other words, for auction $i$ let

$$T_i := \{(t_{is}, \text{Price}_{is}^{(1)}) \mid t_{is} < \frac{2}{3} \times 7 \text{ days}\}$$

be the time–price pairs that are observed during the first two-thirds of the 7-day auction. These are the training data. Similarly, let

$$V_i := \{(t_{is}, \text{Price}_{is}^{(2)}) \mid t_{is} \geq \frac{2}{3} \times 7 \text{ days}\}$$

denote the validation data from the last third of the auctions. For comparison, we also investigate the performance of the penalized smoothing splines by using the same approach. Since we cannot fit a penalized smoothing spline to auctions with fewer than three bids, we removed those auctions. This reduces the total set to 132 auctions.
Table 2. Mean-squared prediction error for penalized spline and mixed model forecasting

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean-squared error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penalized spline</td>
<td>1701507</td>
</tr>
<tr>
<td>Mixed model</td>
<td>28352</td>
</tr>
</tbody>
</table>

We estimate both penalized splines and mixed model splines from the training data and compute the mean-squared prediction error based on the validation data. Specifically, for the penalized splines we estimate the model

\[ s(\text{Price}_{is}) = \alpha_0 + \phi^T(t_{is})\alpha_i \]

whereas in the case of the mixed models we estimate

\[ s(\text{Price}_{is}) = \tilde{\alpha}_0 + \phi^T(t_{is})\alpha + b_{i0} + \phi^T(t_{is})\tilde{\alpha}b_i. \]

The mean-squared prediction error is shown in Table 2. We can see that the penalized splines result in a mean-squared error that is almost 60 times larger than that of the mixed model approach. This implies that taking a traditional smoothing approach can result in forecasts that are severely off.

This result is not too surprising since \(P\)-splines are known to be unreliable at the end of their trajectories. Moreover, simply using semiparametric mixed models to forecast an auction may also not be satisfying in practice. However, what Table 2 shows is that, if we have the choice between a \(P\)-spline and a semiparametric mixed model, then the latter will provide more reliable forecasts. This is important since the result can be used in more elaborate on-line auction forecasting models such as those which were developed in Wang et al. (2007) or Jank et al. (2006).

5. Conclusion

Functional data analysis often arrives with many data-related problems and challenges. One such challenge is sparse and unevenly distributed data. Traditional smoothing approaches often break down and/or produce conceptually not very meaningful results when the functional objects are sampled sparsely and unevenly. We propose a new approach which is based on the concept of mixed model methodology. In particular, we propose semiparametric mixed models together with boosting for parameter estimation. Our approach has several appealing features. First, by borrowing information from similar functional objects, we can overcome challenging sparse data situations with as little as only one sample point per functional object. Moreover, our approach also allows dependences across functional objects to be captured. This is especially appealing in situations like ours where different processes (i.e. auctions) are hardly independent of one another. And, lastly, by assuming a common underlying trend for all functional objects and by estimating this trend from all the data, our approach manages to capture the underlying data monotonicity without explicitly assuming any model constraints. Our boosting approach allows for a convenient joint estimation of all model and smoothing parameters under one roof. The resulting model is parsimonious in that it adds only two additional parameters: the
variance of the slope and the covariance between slope and intercept. It is very flexible, yet easy to interpret, which may make it an uncomplicated and pragmatic model for functional data.

On the substantive side, we contribute to the literature on on-line auctions by suggesting a new way of accounting for dependences across different auctions. Much of the current on-line auction literature assumes independence, which is typically not out of ignorance of the facts, but due to the lack of appropriate statistical models. On-line auction data feature complicated dependence structures: auctions for the same (or similar) product may be correlated because they are competing for the same set of bidders. Moreover, repeat auctions by the same seller may be similar in terms of auction design (e.g., length of the auction, opening bid, usage of a secret reserve price, the number of pictures and the quality of descriptions). This similarity in turn may lead to similar auction outcomes. And, lastly, bidders have the freedom to participate in more than one auction. As a consequence, events in one auction (e.g., stark price increases) may cause bidders to update their strategies in other auctions and thus the bids that a bidder places in one auction are no longer independent from the bids that she or he places in another auction. All of this means that on-line auction data can feature complicated dependences. Our approach is one step towards capturing some of these dependence structures.

Alternatives to our model formulation exist. One alternative could be to fit a random-regression model with a spline as the primary underlying population curve and random slopes and intercepts for time for the individual profiles. Although this model is easily fitted we think that our approach, namely defining an individual slope on the smooth underlying function, has strong conceptual advantages. The model allows the steepness of an unknown function to vary across auctions such that some may be very steep whereas others may be almost flat. A limitation of the separation of spline and random effects, even if quadratic effects are included, is that the individual functions are restricted to linear and quadratic deviations from the population mean, and the auction dynamics are completely separated from the additive quadratic time effects. In our approach the common spline function varies across auctions in the steepness of the bidding process and the initial bidding structure. So the dynamic of auctions is completely specified by the common spline function, the random intercept and the modification on the common spline. Of course the random slope on population function is also restrictive, assuming a distinct structure in the underlying model.

Another limitation to our current approach is that we focus only on auctions of the same duration (7 days). Although it is possible to model a wider range of durations by transforming each time axis to the [0, 1] scale, we must be careful in choosing the right transformation. A simple rescaling from [0, 7] to [0, 1] could lose much valuable information. The reason is that, for two auctions of different length, the information is not distributed symmetrically. What we mean by that is that, for a 7-day auction, there is typically some bidding activity at the beginning (say the first day), not much activity during the middle and plenty of activity in the final moment (say the last 6 or 12 h). This is similar for a shorter 3-day auction: some activity during the first day, not much during the middle and plenty of activity in the final 6 or 12 h. (Note that the spans of the early and last moment activity are typically relatively unaffected by the length of the auction). So, a simple scaling to [0, 1] will miss the change from no to high activity. In fact, finding an appropriate transformation for on-line auctions is a topic of current research and involves estimation of a suitable ‘auction clock’.

Acknowledgements

We thank the Joint Editor, the Associate Editor and two referees for their many helpful comments and suggestions which significantly improved the manuscript.
Appendix A: Algorithmic details

The algorithmic details of boosting are given below. For additional details, in particular on how to obtain good starting values, see Tutz and Reithinger (2007).

A.1. Algorithm BoostMixed

Step 1: initialization—compute starting values $\hat{\alpha}_0^{(0)}$ and $\hat{\alpha}^{(0)}$ and set $\hat{\eta}_i^{(0)} = X, \delta^{(0)}$ and $\hat{Z}_i^{(0)} = (1, \Phi \hat{\alpha}^{(0)})$.
Step 2: iteration—for $r = 1, 2, \ldots$, perform the following cycle.

(a) Refitting of residuals:
(i) computation of parameters—fit the model for residuals

$$y_i - \eta_i^{(r-1)} \sim N(\eta_i, V_i^{(r-1)})$$

with

$$V_i^{(r-1)} = V_i \{ \hat{\rho}^{(r-1)}, (\hat{\sigma}_v^2)^{(r-1)} \} = (\hat{Z}_i^{(r-1)})^T Q(\hat{\rho}^{(r-1)}) \hat{Z}_i^{(r-1)} + (\hat{\sigma}_v^2)^{(r-1)} I$$

and $\eta_i = X, \delta$ yielding $\hat{\delta}$.
(ii) stopping step—stop if BIC$^{(r-1)}$ is smaller than BIC$^{(r)}$.
(iii) update—update for $i = 1, \ldots, n$ using $\hat{\delta} = (\hat{\alpha}_0, \hat{\alpha})$

$$\eta_i^{(r)} = \eta_i^{(r-1)} + X, \hat{\delta},$$
$$\hat{\alpha}^{(r)} = \hat{\alpha}^{(r-1)} + \hat{\alpha},$$
$$\hat{\alpha}_0^{(r)} = \hat{\alpha}_0^{(r-1)} + \hat{\alpha}_0$$

and set $\hat{Z}_i^{(r)} = (1, \Phi_i \alpha^{(r)})$.

(b) Computation of variance components: the computation is based on the penalized log-likelihood

$$l_p(\theta | \eta^{(r)}; \hat{\delta}^{(r)}) = -\frac{1}{2} \sum_{i=1}^{n} \log(|V_i^{(r)}|) + \sum_{i=1}^{n} (y_i - \eta_i^{(r)})^T V_i^{(r)} (\rho, \sigma_v^2)^{-1} (y_i - \eta_i^{(r)}) - \frac{1}{2} (\hat{\delta}^{(r)})^T K \hat{\delta}^{(r)};$$

maximization yields $\rho^{(r)}$ and $\sigma_v^2^{(r)}$.

References


