Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.

VOLUME 26, NUMBER 4
OCTOBER–DECEMBER 2010
ISSN 0169-2070
ELSEVIER
international journal of forecasting
IIF
International Institute of Forecasters

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Available online at www.sciencedirect.com

ScienceDirect

International Journal of Forecasting 26 (2010) 666–683

international journal of forecasting

www.elsevier.com/locate/ijforecast

Real-time forecasting of online auctions via functional K-nearest neighbors

Shu Zhang[a], Wolfgang Jank[b], Galit Shmueli[b,*]

[a] *Applied Mathematics and Scientific Computation Program, University of Maryland, College Park, MD 20742, USA*
[b] *Robert H. Smith School of Business, University of Maryland, College Park, MD 20742, USA*

Abstract

Forecasting prices in online auctions is important for both buyers and sellers. With good forecasts, bidders can make informed bidding decisions and sellers can select the right time and place to list their products. While information from other auctions can help forecast an ongoing auction, it should be weighted by its relevance to the auction of interest. We propose a novel functional K-nearest neighbor (fKNN) forecaster for real-time forecasting of online auctions. The forecaster uses information from other auctions and weights their contributions by their relevance in terms of auction, seller and product features, and by the similarity of the price paths. We capture an auction's price path by borrowing ideas from functional data analysis. We propose a novel Beta growth model, and then measure the distances between two price paths via the Kullback–Leibler distance. Our resulting fKNN forecaster incorporates a mixture of functional and non-functional distances. We apply the forecaster to several large datasets of eBay auctions, showing an improved predictive performance over several competing models. We also investigate the performance across various levels of data heterogeneity, and find that fKNN is particularly effective for forecasting heterogeneous auction populations.

© 2009 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

Keywords: eBay; Functional forecasting; Functional data; Kullback–Leibler distance; Beta distribution; Dynamics

1. Introduction

The popularity of online auctions, such as those on www.eBay.com, has surged in recent years. This is due in part to their wide accessibility, their low participation barriers, and also the auction mechanism, which engages its participants in stimulating competitive behavior. The popularity of online auctions has lead to a growth in related research, and particularly in the desire to *predict* the outcome of an auction before its close. Knowing the auction's closing price has several advantages for auction participants. Bidders can use this information to make more informed (and perhaps even automated) bidding decisions (e.g. Jank & Zhang, 2008). Sellers can use predictions to identify times when the market is more favorable for selling their products and to better evaluate the value of their inventory.

[*] Corresponding author.
E-mail address: gshmueli@rhsmith.umd.edu (G. Shmueli).

0169-2070/$ - see front matter © 2009 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.ijforecast.2009.08.006

Author's personal copy

Different approaches have been proposed for predicting the price of an ongoing auction. Wang, Jank and Shmueli (2008) used regression-based models to forecast an auction's final price in a dynamic fashion (see also Ghani & Simmons, 2004; Jap & Naik, 2008). Common to all these models is the fact that they use information from a set of past auctions to predict an ongoing auction of interest. Moreover, for the purpose of model estimation, they weight the information from each past auction equally. For instance, if the goal is to predict the end price of a laptop auction based on a sample of historical auctions, then estimating a regression-type model will put equal weight on information from a *Dell* laptop and that from an *IBM* laptop — which may be inappropriate if the goal is to predict an auction for a *Sony* laptop. While some of the brand and product differences can be controlled using appropriate predictor variables, there might still be intrinsic differences that are hard to measure. An alternative to regression-based models which was proposed by Caccetta, Chow, Dixon and Stanton (2005) is a classification and regression tree. However, as the authors point out, the prediction can be poor if the prices in each final tree-node vary significantly. Moreover, while trees, unlike regressions, manage to partition the data in a very flexible way, their predictions, like those of regressions, are also based on the *un-weighted* information in each final node. In this paper, we propose a novel and flexible approach for forecasting online auction prices based on the ideas of *K-Nearest Neighbors* (KNN).

KNN is a forecasting approach that weights the information from each record differently, depending on how similar that record is to the record of interest. For instance, if our goal is to predict the end price of an auction for a *Sony* laptop, then it will put more weight on information from other *Sony* laptops and downweight the information from, say, *Dell* or *IBM* laptops. More specifically, KNN makes a prediction based on the weighted average of the K nearest neighbors of the item being predicted, where the weight is proportional to the proximity of the neighbor to the predicted item. KNN has been shown to converge to the true value for arbitrarily distributed samples (Devroye, 1981; Kulkami & Posner, 1995; Stone, 1977), but studies also show that its effectiveness is greatly affected by the choice of the number of neighbors (K) and the distance metric (Cover & Hart, 1967; Goldstein, 1972; Kulkarni, Lugosi & Venkatesh, 1998; Short & Fukunaga, 1981).

In the context of online auctions, the choice of the distance metric is challenging because auctions vary over many conceptually different dimensions. In particular, online auctions vary in terms of three types of information: *static*, *time-varying* and *dynamic* information. Static information is information that does not change throughout the auction. This includes product characteristics (e.g., brand, product condition) and auction and seller characteristics (e.g., auction length, whether there is a secret reserve price, or whether the seller is a power seller). Time-varying information updates itself during the auction (e.g., the number of bids or bidders). Both static and time-varying information have been shown to be important for forecasting the auction price, because differences in product and bidding characteristics all influence bidders' decisions, and hence the final price. Finally, auctions also vary in terms of their *dynamic* information. Dynamic information refers to the price path and its dynamics. This includes the price-speed and the rate at which this speed changes throughout the auction. Auction dynamics are important for forecasting the final price, because an auction that experiences fast price movements in the earlier stages is likely to see a slow-down in price in later stages; conversely, auctions whose price travels very slowly at the beginning often see price-accelerations toward the end (e.g. Jank & Shmueli, 2009; Shmueli, Jank, Aris, Plaisant & Shneiderman, 2006; Wang et al., 2008).

Auction price dynamics can be captured via functional objects such as curves. This means that bids are viewed as a discrete realization of an underlying smooth price path. This price path is recovered from the discrete observations using smoothing methods (Simonoff, 1996), and the smoothness of the resulting object allows the dynamics to be gauged by taking derivatives. In this paper, we propose a novel *functional KNN* forecaster (*fKNN*), which combines functional and non-functional data, for forecasting prices in online auctions.

One challenge with functional methods is the choice of smoother. Typical smoothers include penalized splines (p-splines) and monotone splines. However, while p-splines cannot guarantee the monotonic nature of the auction price growth, monotone splines can be computationally burdensome. One alternative

Author's personal copy

is to use a flexible parametric approach that can capture different types of price growth patterns. Hyde, Jank and Shmueli (2008, chap. 13) proposed a set of four parametric growth models for capturing the price paths of online auctions. In Section 3, we propose a parsimonious parametric form that generalizes these four growth models. Our parametric model has many appealing features, such as monotonicity and computational efficiency. It is particularly important within the context of fKNN, since it allows us to measure the distance between auctions' dynamics in a very parsimonious way via the Kullback–Leibler distance (Basseville, 1989).

Our fKNN forecaster, which integrates information of various types, uses different distance metrics for each data type. In Section 4 we discuss the different distance measures and how they are combined into a single distance metric. We also discuss another important aspect of KNN forecasters, which is the choice of K. While choosing values of K which are too small eliminates important information, choosing values of K which are too large results in noise that causes the forecast accuracy to deteriorate. The goal is to find the value that best balances the signal and noise. Stone (1977) found that the value of K can depend on the distribution of the data, and that the optimal value of K often increases with the sample size. In this paper, we investigate the optimal value of K as a function of different distance metrics, as well as of data size and heterogeneity.

The paper is organized as follows. In Section 2, we introduce the two sets of eBay data used in this study and discuss their levels of heterogeneity. In Section 3, we discuss the flexible parametric model used for capturing the price path in online auctions. Section 4 investigates the choice of the distance metric (combining distance metrics for static, time-varying and dynamic data) and the optimal choice of K. In Section 5, we describe the results of applying the *f-KNN* forecaster to the two datasets, and compare it to some competing approaches. We conclude and discuss possible extensions in Section 6.

2. Data

We use two datasets from the popular marketplace eBay (www.eBay.com). The datasets vary in terms of their heterogeneity. The first dataset contains auctions selling an identical product, while the second dataset contains auctions for a range of products. Each dataset is described in further detail in what follows.

2.1. *Palm PDA auctions*

Our first dataset includes the complete bidding records for 380 auctions that were transacted on eBay between March and May, 2003 (a sample is available online at http://www.forecasters.org/ijf). Each auction sold the same product, namely, a new Palm M515 handheld device. At the time of data collection, the market price of the product was about USD $220 (based on Amazon.com). Each bidding record includes the auction ID, the starting and closing times and prices, all bids with the associated time stamps, and other information such as auction duration, shipping fee, seller's feedback score, whether the seller is a power seller, whether the product is from an eBay store, and whether the auction descriptions include a picture. All of these variables contain information that can affect the final price of the auction. Table 1 presents summary statistics for these variables.

We now briefly describe the aspects of the auction process that the individual variables measure and how they are related to the final price. The opening price is set by the seller, and is known to influence the number of bidders the auction attracts. Regarding the final price, eBay uses second-price auctions, where the winner is the highest bidder and s/he pays the second highest bid (plus an increment). Hence, the final price is equal to the second highest bid plus an increment. Auctions can vary in their duration (between 3 and 10 days, in our data), with 7-day auctions being the default. In terms of auction competition, the average number of bids is 17.45 and the average number of bidders is almost 9. The average shipping fee, set by the seller, is $15.44. This fee is often perceived as a "hidden cost". Another piece of relevant information is the seller's feedback score, which is approximately the number of transactions that the seller has completed on eBay. A seller's feedback score often proxies for his/her credibility. In our data, the highest seller rating is 27,652.

We can see from the bottom half of Table 1 that over 87% of all auctions featured a picture. Pictures carry visual information about products, and thus enhance the bidders' confidence in the quality of the

Author's personal copy

Table 1
Description of the Palm auctions. The top panel reports statistics for all continuous variables; the bottom panel reports statistics for all discrete variables.

Variable	Mean (Stdev)	Median	Min	Max
OpeningPrice	\$76.67 (92.45)	\$9.99	\$0.01	\$265
ClosingPrice	\$229.45 (22.00)	\$232.50	\$172.50	\$290
AuctionLength	5.74 (1.79)	7	3	10
NumberOfBids	17.45 (11.23)	17.50	1	54
NumberOfBidders	8.92 (5.13)	9	1	23
ShippingFee	\$15.44 (5.51)	\$15	\$0	\$50
SellerFeedback	545.73 (1787.47)	44	0	27652

Variable	Yes	No
PowerSeller	121 (31.84%)	231 (60.79%)
eBayStore	117 (30.79%)	235 (61.84%)
Picture	332 (87.37%)	20 (5.26%)

item. Power sellers are sellers with consistently high volumes of monthly sales, over 98% positive ratings, and PayPal accounts in good financial standing. We can see that 30% of sellers are power sellers. Lastly, sellers with feedback scores of 20 or higher, verified IDs, and PayPal accounts in good financial standing are permitted to open “stores” on eBay. Stores provide an easy management of accounts, as well as improved brand boosting when the sellers have multiple items listed. In our data, approximately 30% of all auctions are associated with an eBay store.

2.2. *Laptop auctions*

While the Palm PDA dataset is very homogenous in terms of the products sold, the second dataset consists of auctions for a range of laptops, featuring products of many different makes and models.

The data contain information on 4965 laptop auctions that took place on eBay between May and June, 2004 (a sample is available online at http://www.forecasters.org/ijf). Table 2 summarizes the data. We can see that while some auction variables are similar to those of the Palm PDA data, others are different. For instance, Buy-It-Now auctions are listings that have the option of a fixed-price transaction, thus foregoing the auction mechanism. Over 20% of the laptop auctions included this feature. Moreover, a secret reserve price is a floor price below which the seller is not required to sell. This feature is particularly popular for high-value auctions. We can see that roughly 30% of all laptop auctions make use of the secret reserve price feature.

The main difference between the Palm PDA data and the laptop data is that the latter include products of a wide variety of makes and models. The bottom three panels of Table 2 show that the data include more than 7 different brands, and for each brand the laptops differ further in terms of their memory size, screen size, processor speed, whether they are a new or used product, and whether or not they include an Intel chip or a DVD player. All in all, the products sold in these auctions are of a wide variety of types, which is reflected in the wide range of closing prices (between \$445 and \$1000).

3. A functional model for capturing price growth patterns

Our fKNN forecaster includes both functional and non-functional data. By functional data we mean a collection of continuous objects such as curves, shapes or images. Examples include measurements of individuals' behaviors over time, digitized 2- or 3-dimensional images of the brain, or recordings of 3- or even 4-dimensional movements of objects traveling through space and time. In our context, we consider the price path of an online auction. Such data, although often recorded in a discrete fashion, can be thought of as continuous objects represented by functional relationships. This gives rise to the field of functional data analysis (Ramsay & Silverman, 2005).

Author's personal copy

Table 2
Summary statistics for the laptop auctions. The top two panels report statistics for auction features, while the bottom three panels report summary statistics on the product characteristics.

Variable	Mean (Stdev)	Median	Min	Max
OpeningPrice	93.31 (159.54)	9.99	0.01	900
ClosingPrice	499.22 (210.26)	445	200	999.99
AuctionLength	5.00 (1.81)	5	3	7
NumberOfBids	21.13 (11.05)	19	6	115
NumberOfBidders	9.94 (4.20)	9	1	30

Variable	Yes	No
BuyItNow	1027 (20.68%)	3938 (79.32%)
ReservePrice	1529 (30.80%)	3436 (69.20%)

Variable	Category
Brand (count)	Dell (1622); Fujitsu (15); Gateway (165); HP (1347); IBM (705); Sony (307);Toshiba (535); Other (229)

Variable	Mean (Stdev)	Median	Min	Max
MemorySize	269.12 (157.78)	256	64	2000
ScreenSize	14.03 (0.92)	14	12	21
ProcessSpeed	1125.05 (728.83)	850	133	3200

Variable	Yes	No
NewProduct	628 (12.65%)	4337 (87.35%)
IntelChip	102 (2.05%)	4863 (97.95%)
DVDPlayer	2992 (60.26%)	1973 (39.74%)

Functional data consist of a collection of continuous objects. Despite their continuous nature, the limitations of human perception and measurement capabilities allow us to observe these objects at discrete time points only. Thus, the first step in functional data analysis is to recover, from the observed data, the underlying continuous functional object. This is usually done with the help of data smoothing. Typical data smoothers include penalized splines and monotone splines (Simonoff, 1996). In this paper, we suggest a novel approach to recovering the functional objects via a Beta model. The main advantage of the Beta model is that it allows us to measure the distance between two functional objects using the Kullback–Leibler distance. In contrast to penalized splines, it guarantees the monotonicity of the resulting functional object, which is important for modeling the monotonic price growth behavior in auctions. Compared to monotone splines (which also result in monotonic representations), the Beta model is computationally much more efficient.[1] Recently, Hyde et al. (2008, chap. 13) proposed a family of four growth models for representing auction price paths. Our approach using the Beta model generalizes this idea and includes the four growth models as special cases.

3.1. The Beta model

We model an auction's price path using the Beta cumulative distribution function (CDF). The Beta distribution is a continuous probability distribution defined on the interval [0, 1], with two shape parameters, α and β, that fully determine the distribution. Its CDF

[1] We use the popular R function smooth.monotone in the fda package. An alternative is to use the pcls function (and the accompanying functions gam, smoothCon, and mono.con) in the mgcv package, which is computationally more efficient; however, in our experience it produces inferior fits.

Author's personal copy

can be written as

$$F(x, \alpha, \beta) = \frac{\int_0^x u^{\alpha-1}(1-u)^{\beta-1}\mathrm{d}u}{B(\alpha, \beta)}, \tag{1}$$

where $B(\alpha, \beta)$ is the *beta* function[2] (Abramowitz & Stegun, 1972), a normalization constant in the CDF to ensure that $F(1, \alpha, \beta)$ is equal to unity.

We model auction price paths using the Beta CDF in the following way. Let p denote the sequence of observed prices with associated time-stamps t. Since auctions can be of varying durations, we normalize the time sequence by $t_n = t/Duration$, which yields time-stamps between 0 and 1. Similarly, auctions close at different prices, so we normalize the observed prices by $p_n = p/ClosingPrice$, which yields values of p_n between 0 and 1. The goal is then to find the values of α and β that satisfy $p_n = \int_0^{t_n} u^{\alpha-1}(1-u)^{\beta-1}\mathrm{d}u/B(\alpha, \beta)$ for every element of p_n and t_n.

In the context of real-time forecasting, we only observe price paths up to some time T (with associated price P). We therefore estimate α and β by normalizing the time and price scales to $[0, T]$ and $[0, P]$, respectively (i.e., $t_n = t/T$ and $p_n = p/P$). Estimation is done by error minimization. (See the Appendix for details of the algorithm. The algorithm is implemented using the R function `fbeta.fun`, available online at http://www.forecasters.org/ijf.)

The top panel in Fig. 1 shows typical paths produced by the Beta model for different values of α and β. The solid black line represents the case of rapid price growth at the beginning and the end, but very little growth in the middle; this is representative of auctions with intense early and last-moment bidding, but very little bidding activity in between — a situation which is pretty common on eBay. The dashed red line represents auctions that experience little bidding activity during most of the auction, with bidding picking up only toward the end. In contrast, the dotted green line corresponds to auctions with high levels of early activity, which level off as the auction progresses. Lastly, the dash–dot blue line corresponds to auctions where most of the bidding occurs during the middle part (not at either the beginning or the end), a case that, though rather uncommon, does occur from time to time on eBay.

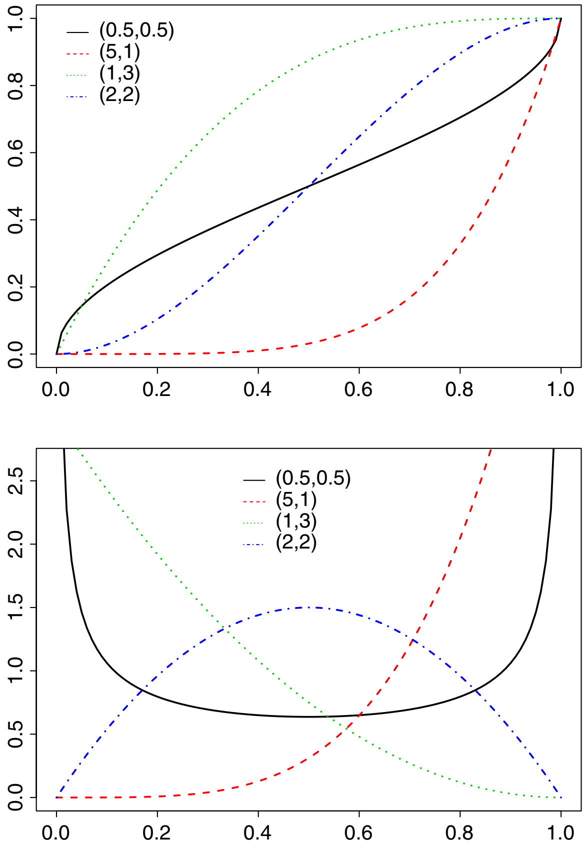

Fig. 1. Top panel: Typical price paths based on the Beta CDF with varying shape parameters (α, β). Bottom panel: corresponding price velocities using the Beta PDF.

3.2. *Representation of price dynamics*

One important consequence of the Beta model is its closed-form representation of the price dynamics. As was pointed out earlier, auction dynamics play an important role in forecasting the auction price. Given the Beta model, the price-velocity can be measured by the first derivative of the price path. Since we model the price path via the Beta CDF (see Eq. (1), its first derivative is given by the Beta probability density function (PDF). Therefore, at any given time t, the price-velocity of an auction with shape parameters α and β is given by

$$\mathrm{Vel}(t, \alpha, \beta) = \frac{t^{\alpha-1}(1-t)^{\beta-1}}{B(\alpha, \beta)}. \tag{2}$$

The bottom panel of Fig. 1 plots the price velocities corresponding to the price paths in the top panel. The solid black line shows rapid dynamics at the beginning and end, but not much price activity in the

[2] $B(\alpha, \beta) = \int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\mathrm{d}u$.

Author's personal copy

middle; while in contrast, the dash–dot blue line signals heightened price dynamics mid-auction. Similarly, the dashed red line captures increased price-velocities toward the end, while the dotted green line captures early price spurts.

Using the representation of the price-velocity in Eq. (2), we can obtain higher-order price-dynamics by taking further derivatives. For example, to measure the price-acceleration we can use the next derivative:

$$\mathrm{Acc}(t,\alpha,\beta)=\frac{t^{\alpha-1}(1-t)^{\beta-1}}{B(\alpha,\beta)}\left[\frac{\alpha-1}{t}-\frac{\beta-1}{1-t}\right].$$

3.3. Model estimation

We estimate the Beta model for our auction data in a way that optimizes the fit in both the x and y directions. In the auction context, the x direction corresponds to time, and a good fit in that direction is necessary in order to accurately capture points of different bidding activity (e.g., early or last-minute bidding). We also require our model to fit well in the y direction, which corresponds to price. A good fit in the y direction will guarantee accurate forecasts of an auction's final price, which is the main goal of this paper. The Appendix explains how one can efficiently fit the Beta model to auction data.

Since we fit the model in both the x and y directions, we measure the goodness-of-fit by examining the residual error in both directions. For the ith auction with n bids, we define the residual as

$$\mathrm{Resid}_i=\frac{1}{n}\sum_{k=1}^{n}\left[0.5(y_k-\hat{y}_k)^2+0.5(x_k-\hat{x}_k)^2\right],$$

which is the average of the sum of squared errors in both the x and y directions. Note that the smaller the residual error, the better our model represents the auction price-path.

In the above definition of a residual, we weight the x and y directions equally (the weight is 0.5) because we have no particular reason to prioritize either direction. Alternatively, one could overweight either the x or y direction if the bidding time or price level, respectively, were of special interest.

Fig. 2 illustrates the model fit for the Palm PDA data.[3] The left panel shows the distribution of residuals for the Beta model, while the other two panels show the corresponding distributions for the growth models (Hyde et al., 2008, chap. 13) and penalized splines, respectively. We can see that the Beta model results in the best model-fit, i.e., the smallest residual error.

3.4. Kullback–Leibler distance

Since the fKNN forecaster uses both functional and non-functional data, we must define distance measures for both data types. While there exist standard measures for the distance between non-functional data (e.g., the Euclidian distance), measuring the distances between functional data (e.g., between two curves) is more involved because of infinite dimensionality. One of the main advantages of the Beta model is that it allows us to measure the distance between two auction price paths in a very parsimonious way via the Kullback–Leibler (KL) distance.

The KL distance (Kullback & Leibler, 1951) is a non-commutative measure of the difference between two probability distributions. For two distributions X and Y, it measures how Y differs from X. The KL distance is widely used in the field of pattern recognition for feature selection (e.g., Basseville, 1989), as well as in physics for determining the states of atoms or other particles (e.g., Nalewajski & Parr, 2000). In our case, X and Y both refer to the Beta distribution, with parameters α,β and α',β', respectively. The KL distance between X and Y is then given by a very simple function of the Beta parameters (Raubera, Braun & Berns, 2008):

$$D_{\mathrm{KL}}(X,Y)=\ln\frac{B(\alpha',\beta')}{B(\alpha,\beta)}-(\alpha'-\alpha)\psi(\alpha)$$
$$-(\beta'-\beta)\psi(\beta)+(\alpha'-\alpha+\beta'-\beta)\psi(\alpha+\beta),\quad(3)$$

where B and ψ denote the *Beta* and *Digamma* functions, respectively (Abramowitz & Stegun, 1972).

Returning to the four auctions in Fig. 1, consider the solid black line ($Beta(0.5,0.5)$) as the focal auction that we want to forecast. Using Eq. (3), the KL distance to the focal auction is 9.69 from the dashed red line ($Beta(5,1)$), 6.40 from the dotted green line ($Beta(1,3)$), and 7.10 from the dotted-dashed blue line ($Beta(2,2)$). While the dotted-dashed blue line may, at least visually, not appear very distant from the focal auction, its distribution is in fact very different, as captured by the KL distance.

[3] The results are very similar for the laptop auctions.

Author's personal copy

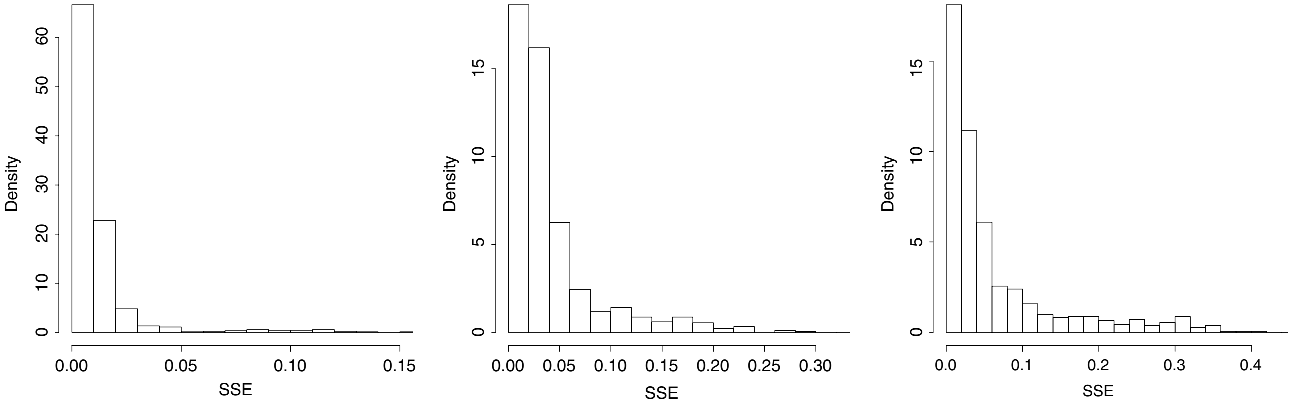

Fig. 2. Residual comparison for fitting three models: the Beta model (left), the growth model (middle), and p-splines (right).

4. Functional K-Nearest Neighbors (fKNN)

In this section we discuss the components of our functional KNN forecaster. We start by explaining the basic forecasting idea, and then discuss the two main elements of our fKNN implementation: the choice of a suitable distance metric and the choice of K.

4.1. Overview

Our goal is to predict the final price of an ongoing auction. Consider Fig. 3. The solid line corresponds to the price-process of an auction that is observed until time T, and the dotted line corresponds to the (future) price path until the close of the auction. Our goal is to predict the closing price. As the closing price is the current price plus the price-increment Δ_f, our forecasting problem is equivalent to predicting Δ_f. We will therefore use fKNN to estimate Δ_f based on a training set of completed auctions.

In order to estimate Δ_f, we look for the K most similar auctions in the training set. Consider Fig. 4 for illustration. In that scenario, we have a training set with 6 auctions, Δ_1–Δ_6. We also have the associated distances, D_1–D_6, between the focal auction and each of the auctions in the training set. If K equals 3, then we will estimate Δ_f using the weighted average of the 3 nearest auctions, in this case Δ_1–Δ_3. More generally, we estimate Δ_f as

$$\Delta_f = \frac{\sum_{i=1}^{K} \frac{1}{D_i} \Delta_i}{\sum_{i=1}^{K} \frac{1}{D_i}}. \tag{4}$$

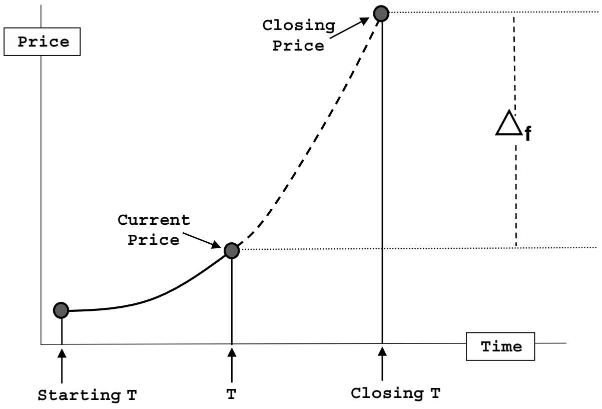

Fig. 3. Illustration of the forecasting idea.

The computation of Δ_f is implemented using the R function `fpred.fun`, available online at http://www.forecasters.org/ijf. As we can see from Eq. (4), the two main elements of this approach are the choice of K and the choice of a distance metric D. We discuss these next.

4.2. Choice of a distance metric

As was pointed out earlier, online auction data consist of three types of information: static information, which captures information that does not change during the course of the auction; time-varying information, which changes during the auction; and auction dynamics, which are captured and represented by functional data. Table 3 summarizes the three types and the specific variables for each data type. We discuss distance metrics for both data types.

Author's personal copy

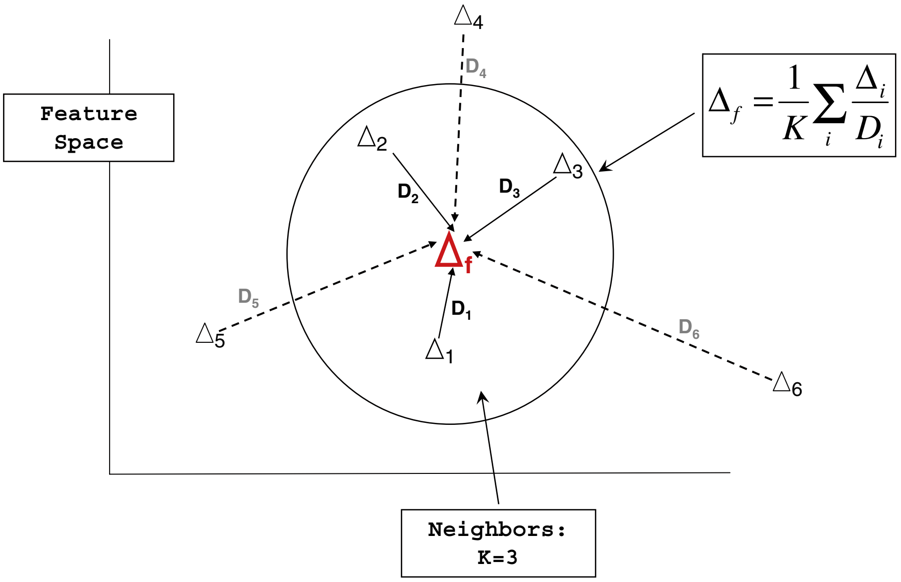

Fig. 4. Illustration of the forecasting scheme.

Table 3
Summary of the information sources characterizing online auctions.

Data type		Measurement scale	Example
Non-functional	Static	Interval Binary Categorical	Opening price, screen size, process speed Buy-it-now, reserve price, condition brand
	Time-varying	Interval	Number of bids, current price
Functional		Functional	Price velocity, price acceleration

4.2.1. Static and time-varying data

Static and time-varying information includes data measured on different scales (interval, binary and categorical). Following Jank and Shmueli (2007), we use separate metrics for each individual scale, and then combine the individual metrics into an overall distance metric for non-functional data.

For binary data x_B and x'_B (e.g., an auction with the buy-it-now option vs. an auction without it), we define the distance as

$$d^B = \mathbf{1}(x_B \neq x'_B), \tag{5}$$

where $\mathbf{1}$ denotes the indicator function, and thus d^B equals 1 if and only if $x_B \neq x'_B$; otherwise it is 0.

We adopt a similar measure for categorical data. For instance, the categorical variable "brand" can assume 8 different levels (Dell, Fujitsu, Gateway, etc.), which can be coded as a vector of 7 different binary variables. Thus, each categorical variable can be represented as a set of binary variables. Let $\mathbf{x}_C$ and $\mathbf{x}'_C$ denote two vectors representing categorical data; we then define their distance, similarly to Eq. (5), as

$$d^C = \mathbf{1}(\mathbf{x}_C \neq \mathbf{x}'_C), \tag{6}$$

which takes the value of 1 if and only if $x_C \neq x'_C$, and is 0 otherwise.

For interval-scaled data x_I and x'_I (e.g., two auctions with different opening prices), we use a scaled version of the Minkowski metric (Jain, Murty & Flynn, 1999):

$$d^I = \frac{\left|\tilde{x}_I - \tilde{x}'_I\right|}{\tilde{R}_I}, \tag{7}$$

where $\tilde{x}$ denotes the standardized value of x, and $\tilde{R}$ denotes the range of $\tilde{x}$. The advantage of the Minkowski

Author's personal copy

metric is that it renders interval-scaled data onto the interval [0, 1]. Note that the maximum and minimum values of d^I are 1 and 0 respectively, which are also the values taken by the binary and categorical distance metrics in Eqs. (5) and (6). Having metrics with comparable magnitudes makes it easier to combine individual distance metrics.

We combine individual distance metrics in the following way. Let $\mathbf{x} = \{x_1, x_2, \ldots, x_p\}$ be a vector of p non-functional features, including binary, categorical and interval data. We compute the overall distance between $\mathbf{x}$ and $\mathbf{x}'$ as

$$d(\mathbf{x}, \mathbf{x}') = \frac{1}{p} \sum_{i=1}^{p} d^*, \tag{8}$$

where d^* denotes the appropriate individual distance metric from Eqs. (5)–(7).

As an example, let x and x' be two three-feature vectors. Specifically, x = {w/buy-it-now, Dell, 1G memory} and x' = {w/o buy-it-now, IBM, 1G memory}. The first, second and third features are binary, categorical, and interval scaled, respectively. Using Eq. (8), $d(x, x') = 1/3(d1 + d2 + d3)$, where $d1 = 1$ is based on Eq. (5), $d2 = 1$ is based on Eq. (6), and $d3 = 0$ is based on Eq. (7). The overall distance between x and x' is therefore $2/3$.

Note that the definition of d in Eq. (8) is flexible, in the sense that one can chose to only use subsets of the available information. For instance, d^{Static} would refer to the distance metric using only static information, while $d^{Time\text{-}Varying}$ would refer to the metric with only time-varying information. One problem with distance metrics of this type is that they may overweight different sources of information, depending on how elaborately each source is recorded. For instance, a data set with 100 different static features and only 10 time-varying features puts 10 times more weight on the information from static features. In order to overcome this potential bias, we follow the ideas of Becker, Chambers, and Wilks (1988) and first scale each individual distance metric by its mean root square (MRS). MRS is a statistical measure of the magnitude of a vector. For a vector $x = \{x_1, \ldots, x_p\}$, MRS is defined as $\sqrt{\frac{1}{p}\sum_{i=1}^{p} x_i^2}$ (Levinson, 1946). We apply the same scaling to each individual distance metric and obtain

$$d_s^{Static} = d^{Static}/\mathrm{MRS}(d^{Static}) \tag{9}$$

$$d_s^{Time\text{-}Varying} = d^{Time\text{-}Varying}/\mathrm{MRS}(d^{Time\text{-}Varying}) \tag{10}$$

$$d_s^{Static\&Time\text{-}Varying} = d_s^{Static} + d_s^{Time\text{-}Varying}. \tag{11}$$

Note that the combined metric $d_s^{Static\&Time\text{-}Varying}$ now puts equal weight on static and time-varying information.

4.2.2. Dynamics (functional data)

As was shown in Section 3.4, we can measure the distance between two functional observations using the KL distance. Let (α, β) and (α', β') denote the Beta parameters for two different auction price paths; their distance (when x is the focal auction) is then defined as

$$d^F = \left| D_{\mathrm{KL}}(x, x') \right|, \tag{12}$$

where $D_{\mathrm{KL}}(x, x')$ is defined as in Eq. (3).

Note that d^F is within the range $[0, +\infty)$, as the KL distance assumes values on the real line. In order to make d^F comparable to the non-functional distance measures, we again scale it using the MRS transformation. Thus, we obtain

$$d_s^{Dynamics} = d^F/\mathrm{MRS}(d^F). \tag{13}$$

4.2.3. Optimal distance metric

To determine which combination of individual distance metrics leads to the best forecasting model, we investigate a series of different distance metrics. In particular, we investigate the performances when using five different metrics: d_s^{Static}, $d_s^{Time\text{-}Varying}$, $d_s^{Dynamics}$, $d_s^{Static\&Time\text{-}Varying}$ and d_s^{All}, where $d_s^{All} = d_s^{Static\&Time\text{-}Varying} + d_s^{Dynamics}$. We first determine the optimal metric based on a validation set, and then investigate the predictive accuracy of the resulting fKNN forecaster on a test set.

4.3. Choice of K

The second important component of fKNN is the choice of K, the number of neighbors from which the forecast is calculated. Too small a value will filter out relevant neighbors; too big a value will introduce noise and weaken the prediction.

Stone (1977) finds that the optimal value of K is data-dependent, and it usually grows with the sample size. In additional, K may also vary depending on the

Author's personal copy

distance metric used. Therefore, we select the optimal value of K separately for each distance metric and data set. To do so, we again select the best value of K based on a validation set, and then apply the resulting model to the test set.

4.4. *Forecasting scheme*

Our complete forecasting process includes determining the optimal distance metric and the optimal value of K. We determine both based on a validation set. Then, using the optimal metric and K, we estimate the fKNN model based on the records in the training set. We investigate the performance of that model by predicting a new focal auction using auctions from a test set.

4.5. *Comparison with alternate methods*

We benchmark our fKNN forecaster against two other very popular prediction methods: parametric regression models and nonparametric regression trees (CART).

In a linear regression model, the closing price is modeled as a linear function of the observed predictor information. This information can include some or all of the three types of data from Table 3. Note that in such models, all auctions from the training set are weighted equally when estimating the model coefficients.

CART forecasting takes a hierarchical approach. It recursively partitions the data into smaller sub-groups, and the focal auction is then forecasted based on the average of the most relevant sub-group. While CART, like KNN, uses neighboring information from similar auctions, it weights each auction equally, which is one major way in which it differs from KNN.

We next discuss differences in prediction performance.

5. Results

We now discuss the predictive performance of our functional KNN forecaster when applied to the two datasets of eBay auctions, and compare it with competing approaches. We also investigate the optimal distance metric and the optimal value of K. The two datasets, Palm PDAs and laptops, are different in their levels of heterogeneity. While the Palm PDA dataset is very homogeneous, the laptop data are very heterogeneous. We also investigate different time horizons; that is, we investigate forecasting at different distances in the future.

We split each of the datasets into a training set (50% of the auctions), a validation set (25%) and a test set (25%). We split the data according to the temporal nature of our prediction task. That is, auctions in the training set are transacted prior to those in the validation set, and auctions in the test set are transacted after those in the validation set. Therefore, our experiments mimic the prediction task that real bidders face.

For the competing models (regression and CART), we train the models on the combined training and validation sets, and then test their predictive performance on the test set.

We evaluate all models using the *mean absolute percentage error* (MAPE)

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y}_i}{y_i}\right|, \tag{14}$$

where y_i and $\widehat{y}_i$ denote the true and estimated final price, respectively, in auction i.

5.1. *Selecting the optimal K and the optimal distance metric*

We select the optimal value of K in the following way. Recall that we have 5 candidate metrics, $D \in \{d_s^{Static}, d_s^{Time\text{-}Varying}, d_s^{Dynamics}, d_s^{Static\&Time\text{-}Varying}, d_s^{All}\}$. For each metric, we select a value of K from the set $K \in \{1, 2, \ldots, 100\}$. For each combination of $(D \times K)$, we estimate the corresponding fKNN model on the training set, then measure its predictive accuracy (in terms of MAPE) on the validation set. Fig. 5 shows the results. The left panel shows the results for the laptop auctions, and the right panel shows the corresponding Palm PDA results. The top panel gives an overview, while the bottom panel zooms in on the most relevant part.

From the left panel of Fig. 5 (laptop auctions) we can see that $d_s^{Dynamics}$ results in the worst model performance, regardless of the value of K. In other words, using only the dynamic information about the price path is not sufficient to achieve good prediction accuracy. We also see that, of the remaining 4 distance

Author's personal copy

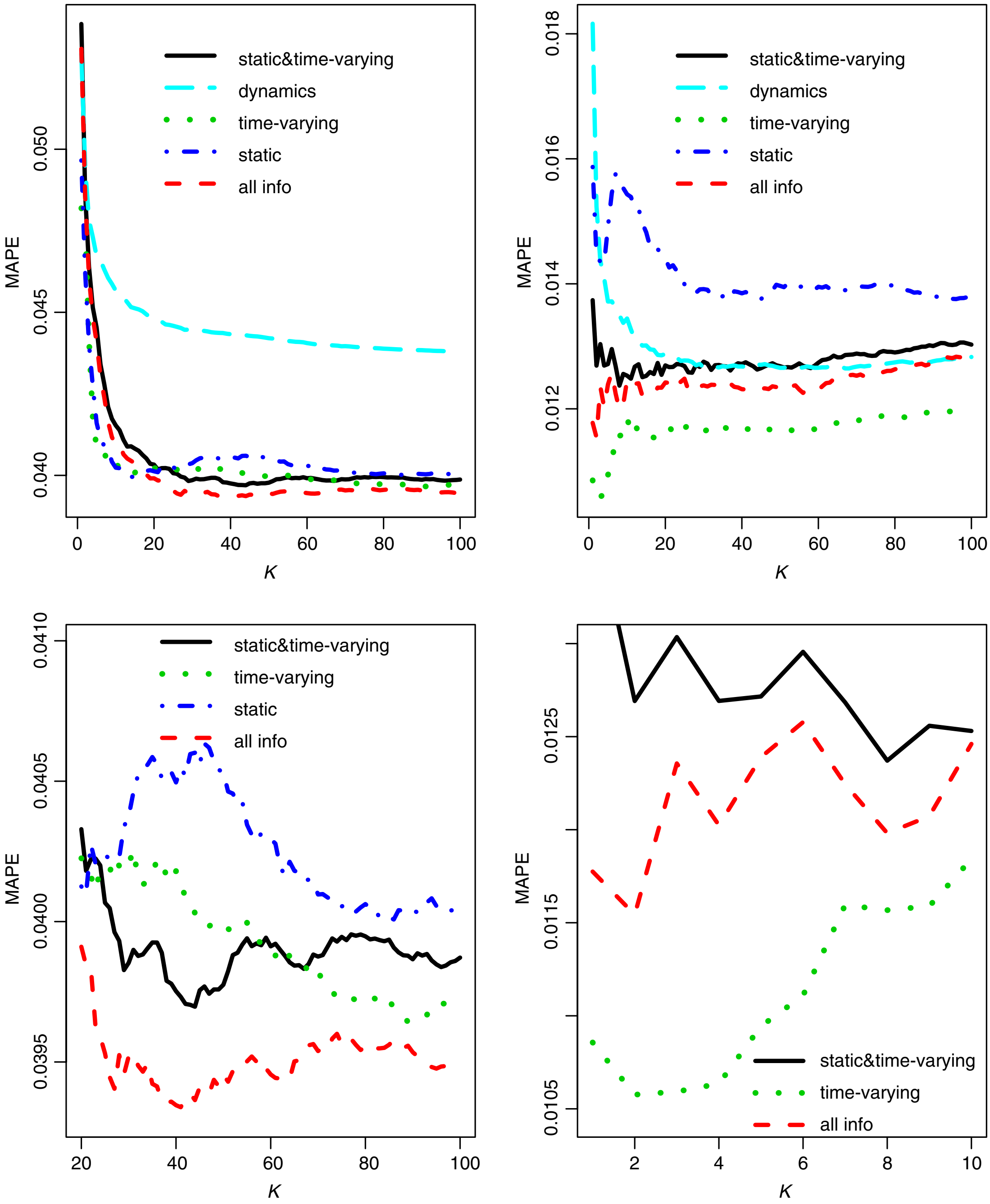

Fig. 5. Optimal values of K and D. The left panel shows the results for the laptop auctions, and the right panel shows the corresponding Palm PDA results. The top panel gives an overview, while the bottom panel zooms in on the most relevant part.

metrics, d_s^{All} uniformly yields the lowest prediction errors. This suggests that for laptop auctions, due to their diversity in makes and models, every single piece of auction information is necessary to achieve good prediction accuracy. Moreover, we note that for d_s^{All}, the lowest prediction error is achieved for $K = 41$. We conclude that the combination of $D = d_s^{All}$ and $K = 41$ results in the best predictions. However, the story is somewhat different for the Palm PDA data (right panel in Fig. 5). For those data, $D = d_s^{Time\text{-}Varying}$ uniformly results in the lowest error (across all distances). Moreover, $K = 2$ is the optimal distance.

Author's personal copy

It is interesting that the two different data sets result in very different choices of K and D. While we need all of the auction information (using the distance metric d_s^{All}) and a very large neighborhood (via $K = 41$) for the laptop data, the Palm PDA auctions require only the time-varying information of the auction process (using $D = d_s^{Time\text{-}Varying}$) and a very small neighborhood (via $K = 2$). One possible explanation for this lies in the difference in heterogeneity between the two data sets. In the homogeneous data (Palm PDA), all products are the same, and any differences in auction outcomes will mostly be due to differences in the current price and level of competition for that product. The competition level is reflected in the number of bids and bidders, which, together with the price level, are captured in $d_s^{Time\text{-}Varying}$. Moreover, since the products are very homogeneous, we only need a very small neighborhood, and thus $K = 2$ is sufficient. However, this is different for the laptop auctions. In that data set, the products are very heterogeneous, and thus the forecaster needs all of the available information (in d_s^{All}) to distinguish between the more relevant samples. Since the products are very different, the method also requires a larger neighborhood, which leads to a larger value of K. This suggests that, as expected, forecasting more heterogeneous auctions is a more difficult task.

5.2. Robustness of the optimal D and K to the time horizon

In the previous section, we investigated the interplay between K and D for a fixed time horizon of 1 min. That is, we assumed that we observe the auction until 1 min before its close. We now investigate the robustness of this choice for different time horizons. Specifically, we investigate the robustness of K and D for different time horizons (δ_T) in the set $\delta_T \in \{2\text{ h}, 1\text{ h}, 30\text{ min}, 15\text{ min}, 5\text{ min}, 1\text{ min}\}$.

5.2.1. Robustness of the optimal K

Fig. 6 investigates the robustness of K to the choice of δ_T. For given values of K ($K \in \{20, 40, 60, 80, 100\}$ for the laptop data and $K \in \{2, 5, 10, 50, 100\}$ for the Palm PDA data), we investigate the predictive accuracy for different values of δ_T. We hold D fixed at $D = d_s^{All}$ for the laptop data and $D = d_s^{Time\text{-}Varying}$ for the Palm data. Fig. 6 shows the *relative* prediction errors $\text{Rel.MAPE}_K = \text{MAPE}_K/\text{MAPE}_{K^*}$, relative to a benchmark value ($K^* = 40$ for the laptop data, $K^* = 5$ for the Palm PDA data).

We can see that for the laptop data (top panel of Fig. 6), lower values of K ($K = 20$) lead to poor performance. We also see that while $K = 40$ generally leads to a good forecasting accuracy, it is outperformed by higher values of K when forecasting time horizons of 30 or 15 min. This suggests that the value of K is not very robust to the time horizon. It is even less robust for the Palm PDA data (bottom panel of Fig. 6), where $K = 5$ only leads to a good forecasting performance for very long time horizons ($\delta_T = 2$ h); in contrast, choosing $K = 2$ leads to the best performance for very short horizons ($\delta_T = 1$ min). This suggests that the choice of K should be a function of δ_T. Table 4 gives the optimal value of K for each combination of δ_T and D.

5.2.2. Robustness of the optimal D

We now investigate the impact of the time horizon δ_T on the choice of the distance metric D. Fig. 7 shows the prediction accuracy as a function of the time horizon δ_T for different choices of D. Note that for each combination of D and δ_T, we use the optimal values of K from Table 4.

The left panel in Fig. 7 corresponds to the laptop data, while the right panel is for the Palm PDA data. Each line corresponds to a distance metric $D \in \{d_s^{Static}, d_s^{Time\text{-}Varying}, d_s^{Dynamics}, d_s^{Static\&Time\text{-}Varying}, d_s^{All}\}$. We can see that, for each data set, a single distance metric consistently yields the best results across all values of the time horizon. That is, d_s^{All} results in the best predictive accuracy for the laptop data, regardless of the value of δ_T. Similarly, $d_s^{Time\text{-}Varying}$ yields the best results for all values of δ_T in the Palm PDA data. This suggests that the choice of the distance metric is very robust to the forecasting horizon, at least for a given data set. We also note that while $d_s^{Time\text{-}Varying}$ significantly outperforms all other distance metrics for the Palm PDA data, for the laptop data most choices of D (except for $d_s^{Dynamics}$) yield largely similar results, at least for short time horizons ($\delta_T \leq 30$ min).

5.3. Comparison with alternative prediction methods

We evaluate the performance of functional KNN by comparing its predictive accuracy with that of more

Author's personal copy

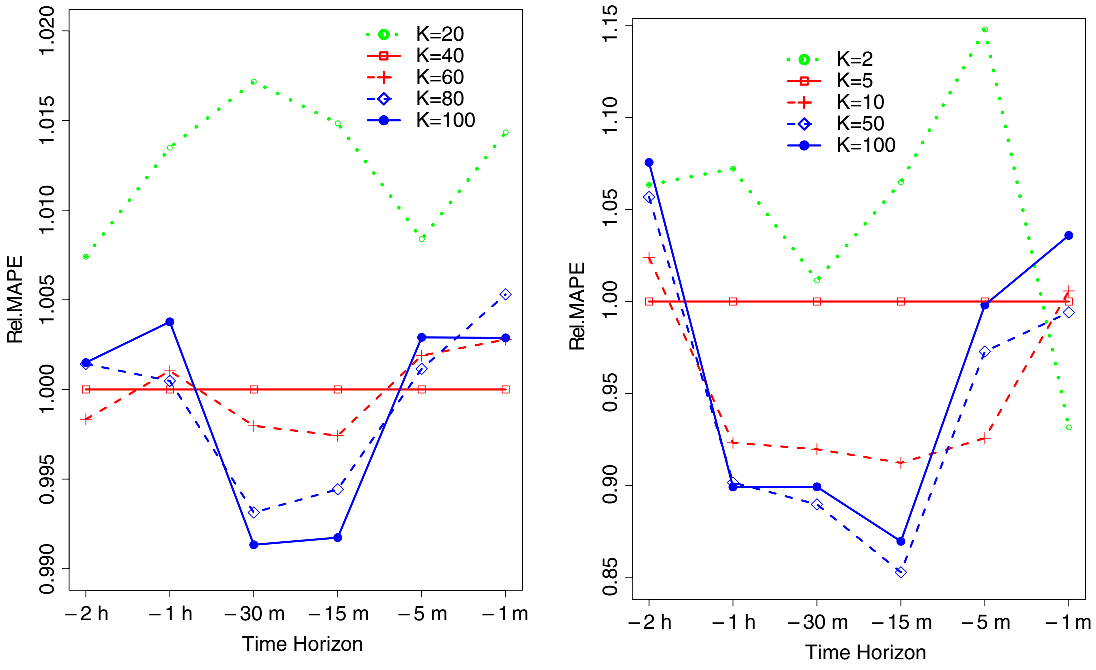

Fig. 6. Relative prediction accuracies for different values of K at different time horizons δ_T. The left panel corresponds to the laptop data, and the right panel to the Palm PDA data.

classical approaches — linear regression models and regression tree (CART) models.[4]

We study the performance of all methods on the test set. Recall that we partitioned our data into a training set (50%), a validation set (25%) and a test set (25%). While we estimated the fKNN forecaster on the training set and optimized its parameters K and D on the validation set, we now compare its performance (using the optimal parameter values) on the test set. That is, for each time horizon δ_T, we use the optimal combination of K and D from the previous section. In order to make fair comparisons, we apply the regression and CART using the same information as for the functional KNN.

Fig. 8 shows the results. We display the *relative* prediction errors between fKNN and the regression model (dotted line) and between fKNN and the tree model (dashed line). It is clear that fKNN generally outperforms its two competitors. In particular, for the laptop data (left panel), fKNN outperforms the tree model by as much as 40%. While the gap between the regression model and fKNN is smaller, fKNN leads to improvements of between 5% and 10%. The picture is similar for the Palm PDA data (right panel). While fKNN leads to general improvements for this data set also, it is curious to see that the two alternative approaches are competitive only for the longest time horizon ($\delta_T = 2$ h).

It is revealing to compare the performances on the two data sets. While both fKNN and regression significantly outperform CART for the laptop data, the gap is not as large for the Palm PDA data; in fact, for the Palm PDA data, CART and regression are comparable for almost all time horizons. The poor performance of CART on the laptop data illustrates the general problem of the method with prediction: while it often fits the training set well, it has a tendency to over-fit the data, and thus perform poorly on the test set, especially in cases like the laptop data where the underlying population is very heterogeneous. On the other hand, functional KNN handles heterogeneous populations well by selecting only those neighbors that are most relevant to the focal auction; in particular, relative to the regression, it performs especially well for forecasting at longer horizons (one–two hours), which is very relevant in practical situations.

[4] We used the software defaults for pruning in CART; that is, we used the defaults in the R package *rpart*.

Author's personal copy

Table 4
The optimal choice of K for different distance metrics D and time horizons δ_T. The top panel corresponds to the laptop data, and the bottom panel to the Palm PDA data.

Time horizon	2 h	1 h	30 min	15 min	5 min	1 min
	Laptop data					
Static	95	94	99	99	81	14
Time-varying	31	27	97	100	91	89
Dynamics	100	100	100	100	100	100
Static& time-varying	40	79	100	96	47	44
All	33	77	98	100	44	41
	Palm PDA data					
Static	52	69	63	63	61	37
Time-varying	3	10	4	1	1	2
Dynamics	94	95	95	29	29	68
Static& time-varying	7	18	32	52	12	8
All	6	40	30	61	11	2

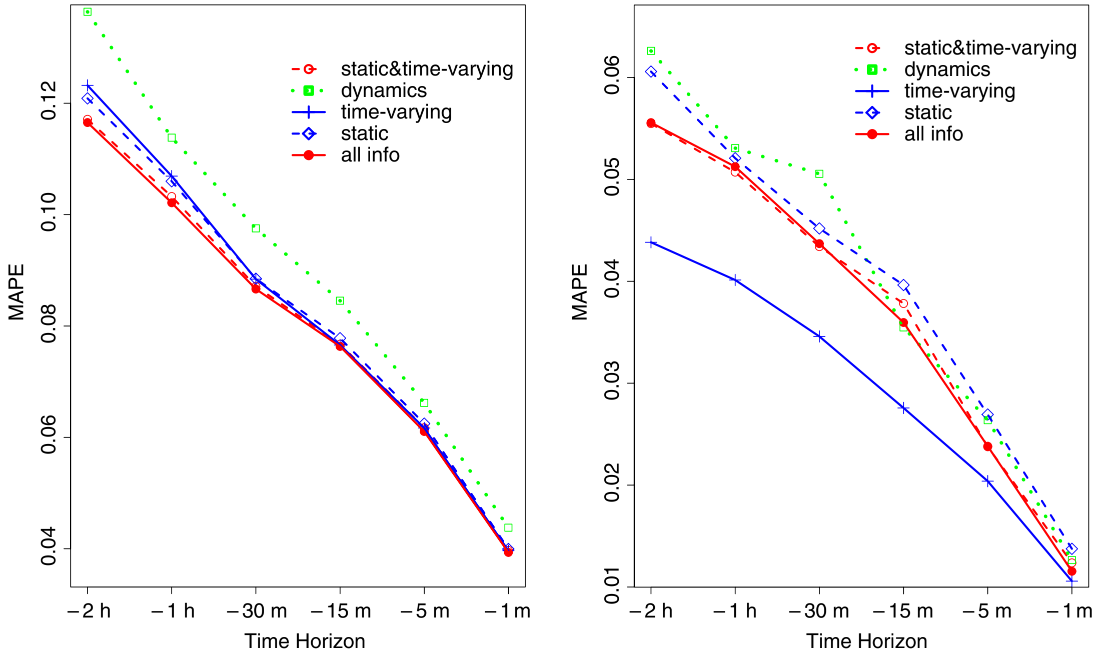

Fig. 7. A comparison of different distance metrics. The left panel is for laptop auctions, and the right panel for Palm PDA auctions.

Functional KNN also leads to improvements for less heterogeneous data sets, such as the Palm PDA data. While the right panel in Fig. 8 suggests that fKNN outperforms both of its competitors for every time horizon, there is a sharp drop for the competitors at $\delta_T \leq 15$ min. At this point, both regression and the tree model perform almost as well as fKNN. A closer investigation of this phenomenon reveals that for this time horizon, the optimal value of K (based on the validation set) equals one (see the left panel of Fig. 9); however, that value leads to very poor performance on the test set (right panel of Fig. 9). This suggests that finding the right value of K is especially difficult for homogeneous data sets (such as the Palm PDA data). While the data homogeneity suggests very small values of K, a slight perturbation of the homogeneity can lead to weaker results. This was also implied by the lack of robustness seen in Section 5.2.1.

Author's personal copy

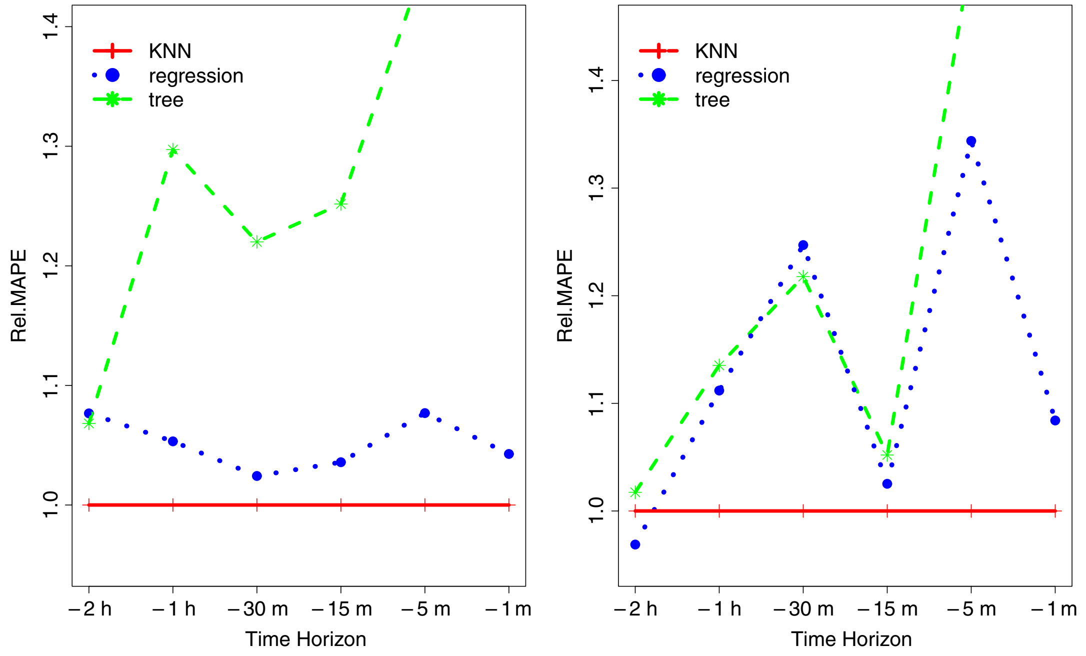

Fig. 8. A comparison of different forecasting methods. The left panel corresponds to the laptop auctions, and the right panel to the Palm PDA auctions.

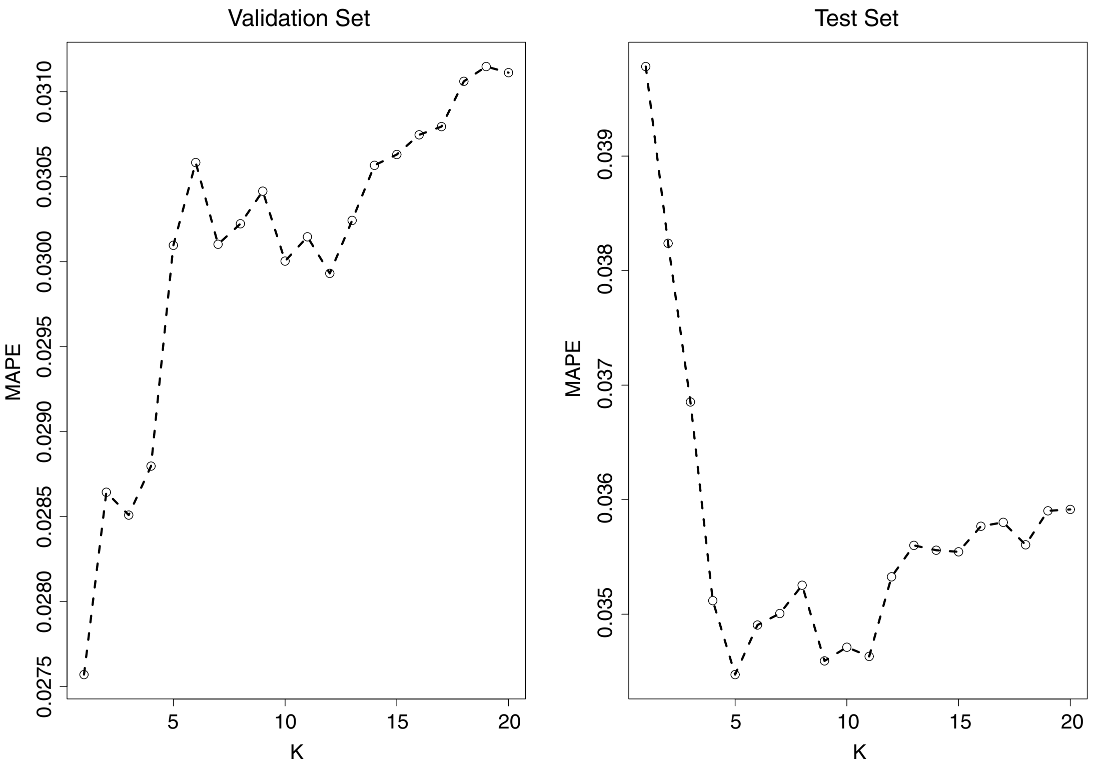

Fig. 9. Optimal values of K for the Palm PDA data at $\delta_T = 15$ min. The left panel corresponds to the validation set, and the right panel to the test set.

Author's personal copy

6. Conclusions

In this paper, we propose a novel functional KNN forecaster for forecasting the final price of an ongoing online auction. Assuming that auctions which are more similar will contain more relevant information for incorporating into forecasting models, we propose a novel dissimilarity measure that takes into account both static and time-varying features, as well as information about the auction's price dynamics. The latter is obtained via a functional representation of the auction's price path. Based on a validation set, we select both the optimal distance metric and the optimal number of neighbors. We find that weighting information unequally yields better forecasts than those from classical methods such as regression models or trees, and this result holds for sets of auctions of varying levels of heterogeneity.

Although we observe that the KNN forecaster improves on regression and CART for sets of auctions of varying levels of heterogeneity, our study shows that the improvement is bigger for heterogeneous data. This means that selecting the most useful information and making use of only the most relevant neighbors is especially crucial for prediction accuracy in situations where the objects are heterogeneous and the information is noisy. This is true not only for forecasting online auctions but also in many other forecasting situations (e.g., weather forecasting). Another finding worth noting is the robustness of the optimal distance to the time horizon. The fact that the same distance metric is optimal regardless of the time horizon implies that the most important information for making price predictions is time-invariant. This insight simplifies the process of decision making. To produce forecasts, we only need to find the optimal distance once, and this distance can then be re-used as the forecasting process proceeds.

There are several ways to extend this study. While we scale the distance metrics for different information sources to achieve equal weighting across all metrics, one could alternatively assign individual weights to individual metrics and then optimize the weights. There are also alternative ways of defining the distances for different data types. For example, for categorical data we can define several levels of category "similarity", such as "US brand". Then, the distance between items can be set to 0.5 for "similar categories" (e.g., laptops of a US brand), or 1 for categories that are more different.

Another way to complement this study would be by investigating alternatives to classical linear regressions and regression trees, e.g., weighted regressions or weighted tree models, which might lead to forecasting advantages, especially for heterogeneous data.

Appendix. Fitting the Beta model

The inputs to our algorithm are the observed discrete bids; the outputs are the estimated shape parameters α and β that represent the auction price-path.

For a given auction, we estimate α and β from the observed bids as follows:

Step 1: Standardize bid levels and bid times. Since both the range (y) and the domain (x) of the Beta CDF are [0, 1], we first standardize the bid levels and times by the following two transformations.

$$y \leftarrow \frac{bid - \min(bid)}{\max(bid) - \min(bid)}$$

and

$$x \leftarrow \frac{time - \min(time)}{\max(time) - \min(time)},$$

where x and y are standardized bid levels and times and lie within [0, 1].

Step 2: Compute α_0 and β_0, the initial values of $\hat{\alpha}$ and $\hat{\beta}$. Since we treat x as a beta distributed random variable, it is reasonable to assume that the empirical average and variance of x are close to their theoretical mean and variance. That is, $mean(x) \simeq \frac{\alpha}{\alpha+\beta}$ and $\mathrm{var}(x) \simeq \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$. Therefore, the initial values of α and β are found by solving the minimization problem:

$$(\alpha_0, \beta_0) = \{(\alpha^*, \beta^*) | DIST^A(\alpha^*, \beta^*) \\ = \min(DIST^A(\alpha, \beta))\},$$

where

$$DIST^A(\alpha, \beta) = \left(mean(x) - \frac{\alpha}{\alpha+\beta}\right)^2 \\ + \left(\mathrm{var}(x) - \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}\right)^2.$$

Step 3: Compute $\hat{\alpha}$ and $\hat{\beta}$. In order to capture both the bid levels and the bidding times, our model

Author's personal copy

minimizes the model error in both the y and x directions simultaneously. Specifically, we choose to minimize the sum of the squared residuals in the y and x directions. With the initial values α_0 and β_0 from Step 2, we solve for $\hat{\alpha}$ and $\hat{\beta}$ through the following minimization problem:

$$(\hat{\alpha}, \hat{\beta}) = \{(\alpha^*, \beta^*) | DIST^B(\alpha^*, \beta^*) \\ = \min(DIST^B(\alpha, \beta))\},$$

where $DIST^B(\alpha, \beta) = \sum(y - pbeta(x, \alpha, \beta))^2 + \sum(x - qbeta(y, \alpha, \beta))^2$; and $pbeta$ and $qbeta$ represent the cumulative distribution function and the inverse of the cumulative distribution function of the beta distribution, respectively.

References

Abramowitz, M., & Stegun, I. E. (1972). *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. New York: Dover.

Basseville, M. (1989). Distance measure for signal processing and pattern recognition. *Signal Processing*, *18*(4), 349–369.

Becker, R., Chambers, J., & Wilks, A. (1988). *The new S language*. Pacific Grove, Ca.: Wadsworth & Brooks.

Caccetta, L., Chow, C., Dixon, T., & Stanton, J. (2005). Modelling the structure of Australian wool auction prices. In A. Zerger, & R. Argent (Eds.), *Modsim 2005 international congress on modelling and simulation* (pp. 1737–1743). Modelling and Simulation Society of Australia and New Zealand.

Cover, T. M., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *IT-13*, 21–27.

Devroye, L. P. (1981). On the almost everywhere convergence of nonparametric regression function estimates. *The Annals of Statistics*, *9*, 1310–1319.

Ghani, R., & Simmons, H. (2004). Predicting the end-prices of online auctions. In *International workshop on data mining and adaptive modelling methods for economics and management, held in conjunction with the 15th European conference on machine learning, ECML/PKDDD 2004*.

Goldstein, M. (1972). K-nearest neighbor classification. *IEEE Transactions on Information Theory*, *IT-18*(5), 627–630.

Hyde, V., Jank, W., & Shmueli, G. (2008). A family of growth models for representing the price process in online auctions. In W. Jank, & G. Shmueli (Eds.), *Statistical methods in e-commerce research* (pp. 291–324). Wiley & Sons, New York.

Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: A review. *ACM Computing Surveys*, *31*(3), 264–323.

Jank, W., & Shmueli, G. (2009). Studying heterogeneity of price evolution in ebay auctions via functional clustering. In G. Adomavicius, & A. Gupta (Eds.), *Handbook of information systems series: Business computing* (pp. 237–261). Emerald.

Jank, W., & Zhang, S. (2008). An automated and data-driven bidding strategy for online auctions. Working Paper, University of Maryland.

Jank, W., & Shmueli, G. (2007). Modelling concurrency of events in online auctions via spatio-temporal semiparametric models. *Journal of the Royal Statistical Society: Series C*, *56*(1), 1–27.

Jap, S., & Naik, P. (2008). Bidanalyzer: A method for estimation and selection of dynamic bidding models. *Marketing Science*, *27*, 949–960.

Kulkami, S. R., & Posner, S. E. (1995). Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Transactions on Information Theory*, *41*(4), 1028–1039.

Kulkarni, S. R., Lugosi, G., & Venkatesh, S. S. (1998). Learning pattern classification survey. *IEEE Transactions on Information Theory*, *44*(6), 2178–2206.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*, 79–86.

Levinson, N. (1946). The Wiener RMS (root mean square) error criterion in filter design and prediction. *Journal of Mathematics and Physics*, *25*, 261–278.

Nalewajski, R. F., & Parr, R. G. (2000). Information theory, atoms in molecules, and molecular similarity. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(16), 8879–8882.

Ramsay, J., & Silverman, B. (2005). *Functional data analysis* (2nd ed.). New York: Springer-Verlag.

Raubera, T., Braun, T., & Berns, K. (2008). Probabilistic distance measures of the dirichlet and beta distributions. *Pattern Recognition*, *41*(2), 637–645.

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

Short, R. D., & Fukunaga, K. (1981). The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, *27*(5), 622–627.

Simonoff, J. S. (1996). *Smoothing methods in statistics*. New York: Springer-Verlag.

Stone, C. J. (1977). Consistent nonparametric regression. *Annals of Statistics*, *5*, 595–645.

Wang, S., Jank, W., & Shmueli, G. (2008). Explaining and forecasting online auction prices and their dynamics using functional data analysis. *Journal of Business and Economic Statistics*, *26*(2), 144–160.