A Data-Mining Approach to Travel Price Forecasting

Till Wohlfarth, Stéphan Clémençon, François Roueff, Xavier Casellato

A Data-Mining Approach to Travel Price Forecasting

Till Wohlfarth
Telecom Paristech / liligo.com
Paris, France
till@liligo.com

Stéphan Clémençon
Telecom Paristech
Paris, France
stephan.clemencon@enst.fr

François Roueff
Telecom Paristech
Paris, France
francois.roueff@enst.fr

Xavier Casellato
liligo.com
Paris, France
xavier.casellato@liligo.com

***Abstract*—With the advent of *yield management* in the air travel industry, a large body of data-mining techniques have been developed over the last two decades for the purpose of increasing profitability of airline companies. The mathematical optimization strategies put in place resulted in *price discrimination*, similar seats in a same flight being often bought at different prices, depending on the time of the transaction, the provider, *etc.* It is the goal of this paper to consider the design of decision-making tools in the context of varying travel prices from the customer's perspective. Based on vast streams of heterogeneous historical data collected through the internet, we describe here two approaches to forecasting travel price changes at a given horizon, taking as input variables a list of descriptive characteristics of the flight, together with possible features of the past evolution of the related price series. Though heterogeneous in many respects (*e.g.* sampling, scale), the collection of historical prices series is here represented in a unified manner, by *marked point processes* (MPP). State-of-the-art supervised learning algorithms, possibly combined with a preliminary clustering stage, grouping flights whose related price series exhibit similar behavior, can be next used in order to help the customer to decide when to purchase her/his ticket.**

***Index Terms*—prediction; machine learning;**

I. Introduction

Since the mid-eighties *yield management* (or *revenue management*) has undoubtedly become a keystone of the travel industry's business policy (see [1], [2] or [3]). Many actors of the tourism industry are now intensively using data analysis and sophisticated mathematical techniques in order to predict the *willingness to pay* of various customer segments and maximize their profits [4] [5]. This naturally resulted in a *price discrimination* phenomenon, the same service (*e.g.* same category of seat in a given flight) being possibly transacted at (very) different prices. By contrast, from the customer side, fluctuation prices are mostly a source of worry. It can be very difficult to ground the decision to buy or not a ticket at a certain price/time in a rational way. Kept unsure and unknowing about the future evolution of the price and about the underlying mechanisms, there are times when the customer is exhorted to buy way in advance his/her ticket and some other times when he/she is encouraged to make a last minute purchase on an impulse, hopping for a possible discount. This naturally opens the way for the supply of possible new services, providing decision-making tools to customers.

Liligo.com[1], a real-time travel search engine, finds among more than 250 travel sites, agencies and tour operators all tickets that are available on the web market place for a particular travel search. As it constantly records the result pages, the information collected allows the visualization of the price time series for every ticket, and can be used to gain insight into the effect of the yield management policy conducted by the travel companies. It is one of the goals of Liligo.com to simplify the user experience, by providing him/her informations about the past price evolution and forecasts eventually, based on historical data.

State of the art. In [6], asks, and answers positively the question "Is it possible to develop data mining techniques that will enable customers to predict price changes?". Precisely, the authors developed a multi-strategy data mining algorithm called HAMLET, which combines *reinforcement learning*, *rule based learning* and *time series analysis*. The predictive rule produces a binary output, corresponding to the advice 'buy' or 'wait', based on a list of attributes of the flight (flight number, route, airline, current price and time left before departure). Though promising results have been reported (apart from the vagueness in the construction of the labels assigned to the training data), one faces computational difficulties when trying to implement this method, due to the growing number of routes, not enough historical data being available for each of them. Since the publication of this paper, airfare data-mining has not received more attention in the literature.

It is the purpose of this paper to propose a novel approach, based on a specific representation of the temporal evolution of historical price series, making them amenable to statistical processing. Among the most salient features of such time series, it is noteworthy that travel prices generally evolve at random times by (eventually large) jumps, the amplitude of the jumps and their "frequency" varying possibly a lot depending on the route, the provider, *etc.* The dedicated description of the historical series we introduce here is based on concepts that arise from the theory of (marked) point processes [7], (M)PP in abbreviated form, and allows direct comparison with each other. Though originally very heterogeneous, when preprocessed this way, the series can next feed (un)supervised state-of-the-art learning algorithms([8]) in order to build predictive

[1]www.liligo.com is a product of Findworks Technologies

rules. Precisely, we considered a diversified database with international, national, long and short haul flights, from low-cost and regular companies. In addition, an attempt is made here to incorporate "path properties" (reflecting the beginning of the price series) to the list of attributes (predictor variables). Equipped with these representation tools, we implement data-mining methods in order to extract patterns in historical data and hopefully build efficient predictive models. Data series are first *clustered* so as to exhibit typical fluctuation behaviors. We next tackle the problem of predicting the behavior (*i.e.* the group to which the flight belongs, among the those identified at the previous stage) based on the attributes available and finally show how forecasts can be deduced from the clusters.

The article is structured as follows. The historical database is described in section II, together with the time series representation that is intensively used in the subsequent analysis. In section III, it is explained at length how we then performed the clustering and classification tasks in order to forecast the evolution of the price series. Section IV displays some experimental results, obtained through a list of variants of the general methodology promoted in this paper. Finally, concluding remarks are collected in section V and some lines of further research are sketched.

II. Preliminaries

A. The Data

The (sampled) price series are observed through liligo.com's historical data repository, where all user search results are stored. It is used to build the training and the testing datasets: price series for specific flights are reconstructed at times corresponding to user searches. In the following, a *database item* is uniquely defined by 6 entries: the departure station, the arrival station, the departure date, the return date, the departure transport code and the return transport code namely, which depend on the provider. Thus, for a given round way trip, there are as many items as providers. Indeed prices may be collected from the websites of direct sellers like low cost companies as well as national airline companies.

Focus is here on 6 routes collected on 9 flight tickets providers. The chosen roundtrip flights represent more than 80% of the demand. We consider trips of 3, 7 or 14 days to cover the most common length of stays based on liligo.com statistics.The routes selected are typical examples of the European flight travel market. They are popular enough to correspond to price time series with high sampling rate. The national route Paris-Toulouse includes both flights and train trips. Paris-Budapest route includes both a low-cost company and a regular company. Barcelona and Marrakesh are short haul touristic and business destination, while Bangkok is a long-haul flight, almost exclusively catering to tourists.

A database item contains a set of features either read from the historical data (departure station, arrival station, departure date, return date, supplier, ...) or computed from additional information such as past searches.

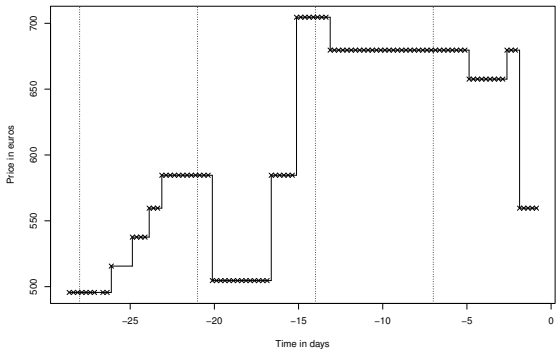
For each database item, prices are sampled every 6 hours over a month before departure which leads to time series with more than one hundred sampling points.

B. Data modelling and preprocessing

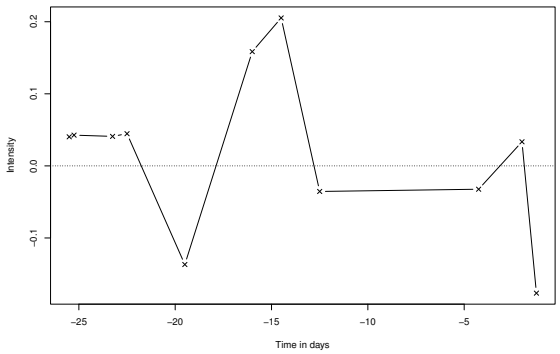
a) Interpolated trajectory: As an illustration, consider the time series displayed in Figure 1(a). It contains the collected prices of a Paris-Marrakesh flight provided by a French low-cost, for a 3 days trip leaving on the 24th of October 2010. The observed trajectory is piecewise constant. The price jumps from one plateau to another, triggered by the yield management system. It is convenient to see the price evolution up to the departure date of the data base item i as a piecewise constant function $p_i(t)$ of the continuous time $t \in [T_0^{(i)} - 28, T_0^{(i)}]$, where $T_0^{(i)}$ denotes the departure time (in days) of item i.

Since we observe a finite sample of this trajectory, the continuous time curve must be approximated by some interpolation scheme. To keep the piecewise constant property, we choose to interpolate $p_i(t)$ between two successively observed prices P and P' respectively at times $T < T'$ as a constant price equal to the firstly observed one, that is, $p_i(t) = P$ for all $t \in (T, T')$. This is an approximation as either $P \neq P'$ and the true instant of jump can be anywhere between T and T' or $P = P'$ and there may have been a successive sequence of jumps between T and T' that have compensated. The error on the jump instant induced in the first case and the probability of the event in the second case are assumed to be small enough to be neglected in our subsequent processing. The $1/(6$ hours) sampling rate of stored price time series indeed seems high enough to neglect these approximations. However, since the observed prices depend on historical searches, a smaller effective sampling rate is achieved. For instance in Figure 1(a), some values are missing in the first week of the series. In this contribution, to avoid taking censorship into account and keep our presentation simple, time series with more than 15% unobserved price samples are discarded. For other time series, we assume that the approximation resulting from the above interpolation scheme keeps being negligible in our data processing. We end up with a set of around 7000 time series from the 23000 series of the complete database.

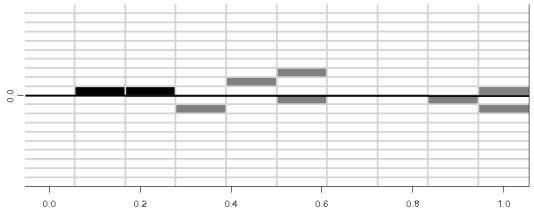
b) Marked point process of returns (MPPR): Piecewise constant trajectory can be entirely reconstructed from the initial price at time $T_0 - 28$ and the sequence of change points in the series. It is in fact sufficient to have the sequence of change points instants with the corresponding *relative jumps* (also called the *price returns*). Let us formalize how this is obtained. From the interpolating scheme described above, we have defined the price $p_i(t)$ for all t between successive observed price instants. We choose the convention that p_i is right continuous. Now, denoting by $p_i(t-)$ the left limit of p_i at time t, a jump instant t is characterized by a discontinuity $p_i(t-) \neq p_i(t)$ and the corresponding relative jumps takes value $\{p_i(t) - p_i(t-)\}/p_i(t-)$. We denote by $(T_k^{(i)}, s_k^{(i)})_{k \leq -1}$ the sequence of couples of successive (jump instants,jump returns) indexed in increasing order so that $T_{-1}^{(i)}$ correspond to the last price jump instant before departure time $T_0^{(i)}$. The



(a) Collected prices of 3 days long Paris-Marrakesh trip, '*' corresponds to observed points, plain lines represent the interpolated trajectory. Vertical dotted lines are displayed every 7 days from the departure date.



(b) Marked point process of returns (MPPR) of the time series displayed in Figure 1(a).



(c) Intensity estimator $\widehat{J_i}$ of the MPPR of Figure 1(b) represented as a grayscale pixelated image with $b_1 = 3\,days$ and $b_2 = 0.1$

Fig. 1. Representation transformations

collection of these points can be represented by a measure that assign a mass 1 to each of these point in the plane $\mathbf{R}^2$,

$$N^{(i)} = \sum_{k \leq -1} \delta_{(T_k^{(i)}, s_k^{(i)})} \ ,$$

where δ_x denotes the Dirac measure at point x. Measure $N^{(i)}$ is called a *marked point process* (MPP), see e.g. [9], with $T_k^{(i)}$, $k \leq -1$, being the *locations* and $s_k^{(i)}$, $k \leq -1$, being the corresponding *marks*. As a consequence each item i in the data base is assigned an *initial price* $P^{(i)}$ and a *marked point process of returns (MPPR)* $N^{(i)}$ that allow to construct the whole interpolated price trajectory with a minimum set of values. We stress moreover that $N^{(i)}$ entirely contains the *relative* price trajectory, that is, it allows to reconstruct the whole trajectory up to a multiplicative constant. This is an important remark for two reasons

1) Our goal is to forecast price decrease events that only depend on the relative trajectory, hence that can be detected from $N^{(i)}$.
2) Aggregating items based on their MPPR's allow to build clusters grouping items with possibly huge difference in their absolute prices.

An example of MPPR is displayed in Figure 1(b). One can see that the whole trajectory is entirely determined by a few points of the plane.

c) Class variable definition: Suppose that a customer is willing to purchase the flight of Item i , t_1 days before departure, that is, at time $T_0^{(i)} - t_1$. Let us set the time origin at the departure time, $T_0^{(i)} = 0$, for convenience, without meaningful loss of generality. We define the class variable φ_i taking values $\{0, 1\}$ respectively corresponding to the classes "buy" and "wait", depending on the preferable choice that should be given to this customer. This choice only depends on the price evolution after $-t_1$. Assuming the customer is able to check for prices on a daily basis up to time $-t_2 \in (-t_1, 0]$, we define $\varphi_i = 1$ if and only if the price $p_i(t)$ remains below $p_i(-t_1)$ over a period of time lasting more than 1 day between $-t_1$ and $-t_2$. It is interesting to note that the value of φ_i only depends on the evolution of the relative price, hence only on the MPPR $N^{(i)}$ restricted to the time interval $[-t_1, -t_2]$. The collected time series are 28 days long. In the following, we set $t_1 = 21\,days$ and $t_2 = 14\,days$. This corresponds to a customer selecting a flight 21 days before its departure time and that is able to delay the definitive purchase up to 14 days before departure time.

d) Intensity image: We will model the MPPR's $N^{(i)}$ as an inhomogeneous Poisson point process, see [9]. Such processes are parameterized by a (non-negative) intensity function J_i on the plane that can be interpreted as follows: the probability of having a jump in the interval $[t, t + \delta t]$ with return value in $[s, s + \delta s]$ is approximatively given by $J_i(s, t)\delta t \delta s$. A simplified way to parameterize $N^{(i)}$ is to use a finite dimensional representation of J_i using a pixelated image of the rectangle $[T_0 - 28, T_0] \times [-1, 1]$. Each pixel corresponds to a constant value of J_i on a rectangle cell/pixel R of size $b_1 \times b_2$. This value can be estimated as the number of jump points $(T_k^{(i)}, s_k)$ $k \leq 1$ lying in R divided by the size of R,

$$\widehat{J}_i(s, t) = \frac{1}{b_1 b_2} \sum_{k \leq -1} \mathbf{1}_R(T_k^{(i)}, s_k), \quad (s, t) \in R \ , \qquad (1)$$

where $\mathbf{1}_R$ denotes the indicator function of the set R. The resulting image is displayed in Figure 1(c) in grayscale.

This representation of the intensity is used as an input and also as an output of the clustering algorithm. That is why it is a key step in our methodology. It is used as an output for items

i on which we wish to forecast prices events. Indeed once J_i is estimated, it is possible to estimate the probability of any events depending only on the relative trajectory. This will be done for the event $\{\varphi_i = 1\}$ using Monte Carlo simulations as described in Section III-D.

e) Final data set: As a result of the data collection and preprocessing described above, we finally obtain a data set containing a collection of items i attached with a set of features $V_i(1), \ldots, V_i(p)$ related to trip characteristics, an initial price P_i, an MMPR $N^{(i)}$ representing the successive jumps instants and relative sizes. For each item i two additional objects are computed from $N^{(i)}$, namely, 1) the estimated image intensity $\widehat{J}_i$ defined as a pixelated image representing an estimator of the intensity of $N^{(i)}$, and 2) the class variable φ_i indicating a decrease event over the period $(T_0^{(i)} - t_1, T_0^{(i)} - t_2)$, where $T_0^{(i)}$ denotes the departure time, and t_1 and t_2 are here set to $21\,days$ and $14\,days$ respectively but can be adapted to the customer's constraints.

III. Prediction method

The final data set described at the end of Section II can now be used for our main goal: providing an efficient, yet adaptable, predictor of decrease events.

The predictor is built in four distinct steps. Step 1 consists in an unsupervised clustering that produces clusters of intensity images. Step 2 is a supervised classification for learning the best cluster from the features only. In Step 3 a common model of the price evolution up to departure time is defined for each cluster. Finally, in Step 4, the predictor is defined, relying on the previous steps. Given a new item i, its corresponding features and partially observed prices, the classifier of Step 2 is used to select a cluster and the corresponding model defined in Step 3 is used to compute the probability of a decrease event of the form $\{\varphi_i = 1\}$ under this model, which can be used as a prediction function of the class variable φ_i. Adaptability is allowed in this step as the model obtained in Step 3 can be used to predict any variable φ_i. Hence the latter can be adapted to the specific purposes of the customer.

As usual the data set is divided into a training set and a testing set with a 2/3–1/3 ratio. The training set is used to build a predictor and the testing set to evaluate the obtained predictor. Steps 1 and 2 are achieved using two different subsets of the training set. Steps 3 and 4 are applied to the testing data set.

A. Step 1. Intensity image clustering

The first step of our approach is to extract groups of similar patterns from the intensity images in the training data set. We use the KMeans clustering algorithm in [10] [11], with an Euclidean distance between intensity images. For two items i and j,

$$d(i,j) = \left(\sum_{\text{all cells } R} |\widehat{J}_i(R) - \widehat{J}_j(R)|^2 \right)^{1/2},$$

where $\widehat{J}_i(R)$ and $\widehat{J}_j(R)$ denote the constant values of the estimated intensity over a cell R, see Eq. (1).

At the end of this step, we obtain a collection of C clusters (or groups) defined as sets of items $I_1, \ldots, I_C$.

B. Step 2. Features based classification

In this step we construct a classifier which, for a given database item i, takes the features $V_i(1), \ldots, V_i(p)$ as input and provides a cluster index $j \in \{1, \ldots, C\}$ as output. We use two different state-of-the-art algorithms : classification tree (CART [12]) and random forest [13].

A benefit of CART is that the created rules can be meaningfully interpreted. As a result, the relative importance of features for the classification problem can be exhibited. Using the random forest [13] , we can also give complementary information on the importance of each attributes in the classification. This will be done in Section IV.

C. Step 3. Price evolution modeling

Each cluster is defined as a set of items I. We consider two approaches for defining a model of price evolution represented by a *cluster MPRR* $N^{(I)}$ of jumps (each jump being again defined as a couple [instant, relative size]).

The first approach is to define an *empirical model* $\overline{N}^{(I)}$ defined as a point process uniformly distributed in the set $\{N^{(i)},\, i \in I\}$ of all MPRR's of the cluster I, that is

$$\overline{\mathbf{P}}_I(\cdot) = \mathbf{P}(\overline{N}^{(I)} \in \cdot) = \frac{1}{\#I} \sum_{i \in I} \mathbf{1}\{N^{(i)} \in \cdot\}\,. \qquad (2)$$

This formula says that the price evolution distribution of the model $\overline{N}^{(I)}$ is defined as an average over all price evolutions that belong the considered cluster I.

The second approach relies on the assumption that all MPRR's $N^{(i)}$, $i \in I$ are realizations of the same inhomogeneous Poisson point process characterized by an intensity function J_I. The estimation of this intensity is given by the mean pixelated image $\widehat{J}_I$ over the cluster, namely $\widehat{J}_I = \frac{1}{\#I} \sum_{i \in I} \widehat{J}_i$. The corresponding model $\widehat{N}^{(I)}$ is defined as the inhomogeneous Poisson process having intensity function $\widehat{J}_I$. Its distribution is not obtained as easily as that of $\overline{N}^{(I)}$, which can be computed directly from (2). The most practical way is to use a Monte Carlo approach: simulate many independent realizations of $\widehat{N}^{(I)}$ over the domain of interest, say $\widehat{N}^{(I,j)}$ for $j = 1, \ldots, M$ and then compute the probability $\mathbf{P}(\widehat{N}^{(I)} \in \cdot)$ as a mean over these simulations

$$\widehat{\mathbf{P}}_I(\cdot) = \mathbf{P}(\widehat{N}^{(I)} \in \cdot) = \frac{1}{M} \sum_{j=1}^{M} \mathbf{1}\{N^{(I,j)} \in \cdot\}\,.$$

To simulate a realization of a 2-dimensional inhomogeneous Poisson process, we refer to [14, Chapter 4]. Observe that $\widehat{J}_I$ can be interpreted as the centroid of the cluster of intensity images J_i, $i \in I$. Therefore we call this model the *centroid model*.

D. Step 4. Price decrease event prediction

For a new item i, the classifier defined in Section III-B selects the most probable cluster corresponding to this item, according to its features and to the available observed prices up to the time $T_0^{(i)} - t_1$. The later is included in the list of features through an incomplete intensity image : each values of $\hat{J}_i$ that can be estimated from the observed prices are included in the list of features used in Step 2. With this method we include a temporal information of the first evolution. We also added a global evolution feature which is the sum of the $\hat{J}_i$ by rows. At the end, we have as new feature as the number of vertical subdivision, and each feature represents the sum $\hat{J}_i$ for the same jump interval.

Now, the selected cluster I and the corresponding model, either the empirical model $\overline{\mathbf{P}}_I$ or the centroid model, $\widehat{\mathbf{P}}_I$ can be used to predict a decrease event $\{\varphi_i = 1\}$.

IV. Experiments and results

Steps 1 and 2 described in Section III have been applied to the training data set (steps). A predictor is then obtained from Steps 3 and 4 and applied to the testing data set. The results are evaluated using the receiver operating characteristic (ROC) curves, as is commonly done in prediction problems, see [15]. A *direct predictor*, described below, is also constructed and compared to the clustering-based one.

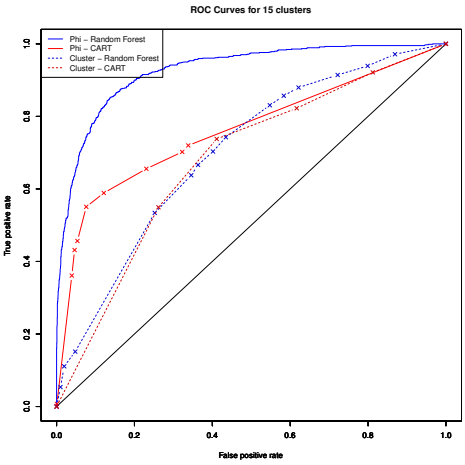
A. Benchmarking based on direct variable prediction

A simpler method than the one described in Section III is to learn φ directly from the set of features. This method is model free. The only goal here is to optimize the prediction (or the ranking) of the decrease event $\varphi = 1$ from the features. Hence we expect better ROC curves than the previous method. Such method requires a definition of φ beforehand (in particular the dates t_1 and t_2) which is a drawback in practice. Nevertheless, we use it here as a benchmark for comparison with the model based approach.

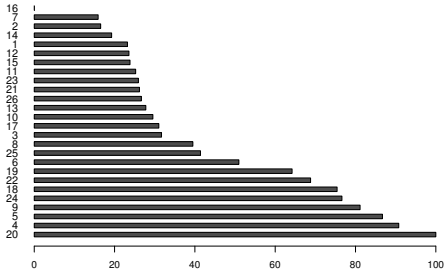
B. Performance results

The ROC curve of the model-based approach is obtained by computing $\mathbf{P}(\varphi = 1)$ for the model of the selected cluster for each item i of the testing data set. This value is used as a Test statistic for choosing $\varphi = 1$. The ROC curve is displayed in Figure 2(a) in dotted lines. Recall that two different classifiers are used in Step 2, resulting in two different ROC curves, namely CART (green) and Random Forest (blue).

The ROC curves obtained from the model free method is displayed on the same graph. As expected the direct method performs much better as its goal is concentrates on a specific event prediction task rather than a complete model of price time series. We believe that the direct method should be used for preregistered purchase periods to give a first coarse advice to the customer. The model based one can then been used to provide a more specific, yet less reliable advice adapted to the needs of the customer.



(a) ROC curve of a model based prediction and a direct prediction using the random forest, CART. The cell size is $(0.1, 3\,days)$ and the number of cluster is 15



(b) Attribute's importance in the φ classification

C. Comparison of features

With the random forest algorithm, we show the importance of each attributes in the classification process. We will be able to give meaningful interpretation of the discriminant attributes. On Figure 2(b), the most important attributes are the the temporal ones (departure day (9), return day (4), day of the year (20), day of the week (19)...).

As we add the first evolutions information, we notice the importance of the first cells of the grayscale in the cluster classification. Obviously the majority of the cells (always equal to zero) will not be useful, but the low variation cells have a significant importance that will give us better result in the attribute based classification. The attribute day, which is the day of the month, is usually very important in the classification, though it seems unlikely that there is a monthly seasonality. But according to liligo.com's trends, the searches for a departure date in the first 10 days of every month is always inferior to the rest of the month. On the other hand, the $day\,of\,week$, $day\,of\,year$ and $demand$ are known as a relevant attributes both in our algorithm and from a marketing point of view, which gives us good hope in our methodology.

D. Comparison with HAMLET

In the paper [6], the performance measure is the earning made with the prediction compared to the optimal decision. The prediction process begins at $T0 - 21$ and every 3 hours HAMLET gives a prediction "buy" or "wait". The total ticket price is the price when the prediction is "buy" and the earn or loss is the difference between this price and the $T0 - 21$ price. The optimal price is the lowest between $T0 - 21$ and $T0$. With this procedure they managed to reach 61.8% of the optimal price. Our prediction give the evolution of the prices for a 7 days period. Because we consider that the user will not go to liligo.com every 3 hours, we considered the relevant decrease as a more than 24 hours decrease. With these parameters we reached 55% of the optimal price. We are almost as efficient as HAMLET, with a more useful prediction, a flexibility (evolution in the newt 7 days, or next 14 days), a confidence (based on the grayscale) and an interpretation of the prediction (trees generated).

V. Conclusion

In this article, we introduced a novel representation of (heterogeneously sampled) price series, that allows for grouping trajectories into clusters, depending on the similarity of their behavior. Equipped with this representation, the clustering stage has been followed by a statistical analysis of the groups thus formed. In particular, the probability distribution on the path space governing the observed trajectories has been estimated for each cluster. Based on historical data, we next learned, via a decision tree method, how to assign to a new flight (viewed through its attributes) a group. A variant, taking into account the first points of the trajectory in the list of predictor variables, has also been considered, in order to obtain more accurate predictions. The clustering approach gives us a great flexibility in regards to the answers we can provide to a customer who would be interested in the price evolution at a larger horizon than the next 7 days. Indeed, this method yields excellent results, not only when trying to directly predict the price evolution at a specific horizon, but also when inferring more general features of the price distribution. We highlight the fact that this approach can be applied to any perishable resources, such as hotel rooms or car rental, and will give to the customers a way to respond to yield management techniques.

The prediction application is designed to be deployed as a standalone web service. Any internet user accessing this web service through a client website or application will be able to request a prediction for a given flight. The prediction service will then provide a set of predefined requests, from a simple "buy"/"wait" advice to a more precise one. According to the request, the service will compute and return the appropriate response by querying the prediction model. The prediction model is built through a training application that systematically updates the historical database several times per day. As it needs to be continuously adjusted, a supervisor application, built on the top of the web service and the training application, monitors the system: it automatically compares predicted and observed data, evaluates the accuracy of the prediction and eventually launches a new training stage when the prediction performance is significantly below the expected level (estimated by means of validation/test samples at the end of the training step).

The method promoted here can be improved in several ways. Regarding the clustering stage, the choice of the distance used in the K-Means algorithm should be discussed and its impact on the partition thus obtained deserves to be investigated. For instance, in order to obtain more interpretable results, instead of comparing two vectors of grayscales through their euclidean distance, it would be more relevant to compare the probability distributions they define using the
Kullback-Leibler distance. All companies with a standing inventory are using dynamic pricing strategies in order to optimize their revenue. This is why it is necessary to use the temporal evolution of the previous prices in the learning process. However, the key information to predict price evolution is undoubtedly the number of seats left. The claim of liligo.com is that the demand curve is almost proportional to the purchase curve.It is thus not hopeless to estimate the number of seats left and improve the learning process by incorporating it to the predictive model.

Finally, the data can be enriched by including the flights with stops, prices found on online travel agencies websites, and by increasing the length of the window over which prices are collected from 28 days to 90 days.

References

[1] B. Smith, J. Leimkuhler, R. Darrow, and Samuels, "Yield management at american airlines," *Interfaces*, vol. 22, no. 1, pp. 8–31, 1992.

[2] S. Daudel and G. Vialle, *Yield management : applications to air transport and other service industries*. P.I.T.A., 1994.

[3] J. I. McGill and G. J. van Ryzin, "Revenue management: Research overview and prospects," *transportation science*, vol. 33, no. 2, pp. 233–256, 1999.

[4] R. E. Curry, "Optimal airline seat allocation with fare classes nested by origins and destinations," *transportation science*, vol. 24, no. 3, pp. 193–204, 1990.

[5] P. P. Belobaba, "Application of a Probabilistic Decision Model to Airline Seat Inventory Control," *Operations Research*, vol. 37, no. 2, pp. 183–197, 1989.

[6] O. Etzioni, R. Tuchinda, C. A. Knoblock, and A. Yates, "Mining airfare data to minimize ticket purchase price," in *ACM SIGKDD*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 119–128.

[7] M. Jacobsen, *Point Process Theory and Applications: Marked Point and Piecewise Deterministic Processes (Probability and its Applications)*, 1st ed. Birkhäuser Boston, December 2005.

[8] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, corrected ed. Springer, July 2003.

[9] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes. Vol. I*, 2nd ed., ser. Probability and its Applications (New York). Springer-Verlag, 2003.

[10] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 2005.

[11] J. Kogan, *Introduction to Clustering Large and High-Dimensional Data*. New York, NY, USA: Cambridge University Press, 2007.

[12] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[13] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, October 2001.

[14] S. Resnick, *Adventures in stochastic processes*. Boston, MA: Birkhäuser Boston Inc., 1992.

[15] H. van Trees, *Detection, Estimation, and Modulation Theory*. Wiley, New York, 1968, vol. 1.