Internship Project

Unconstraining demand applied in the hospitality industry

Vrije Universiteit Amsterdam
MSc Business Analytics - Internship report

Supervisor:

prof.dr. G. M. Koole
ger.koole@vu.nl

dr. E. N. Belitser
e.n.belitser@vu.nl

Author:

R. van Leeuwen
rik.van.leeuwen@student.vu.nl

January 29, 2019

Abstract. Historical bookings in Revenue Management (RM) are seen as constrained demand due to the restriction of capacity or due to various rates for the same product. Research shows that relatively simple unconstraining methods can result in a considerable revenue increase due to a more accurate forecast performance. The goal is **to obtain unconstrained daily historical demand**, which can be used as input for a forecasting algorithm. This graduation project, part of the Master's program of Business Analytics at the VU Amsterdam, describes the steps from raw data until implementation applied in the hospitality industry. Expectation Maximization algorithm is implemented with simulated data and real data. The outcome of EM algorithm is a non-decreasing demand function per choice set per historical day.

Keywords: revenue management (RM), demand unconstraining, Expectation Maximization (EM), hospitality

Table of Contents

1 Introduction

This internship report is part of the graduation project for the Master's program of Business Analytics at the VU Amsterdam. This content of this Masters program is the application of mathematics, computer science and business administration. During this internship prof. dr. G. M. Koole is the main supervisor and dr. E. N. Belitser is the second supervisor. The goal is to validate and implement mathematical methods in order to solve practical problems. Often the issue occurs that theoretical methods described in literature do not solve the reality due the occurrences of random events. Whereas the theoretical methods are mostly tested with simulated data. A problem in the hospitality industry will be solved with simulated and real data.

One of the main challenges in the hospitality industry is selling the *right* room to the *right* guest at the *right* moment for the *right* price. By applying Revenue Management (RM) strategies, hotels attempt to optimize their revenue with, for example, dynamic pricing and allocation according to Talluri and van Ryzin [25]. The common way of RM is selling a fixed number of rooms (the capacity), which are perishable, at a fixed deadline, also known as the check-in date. Based on, among others, historical reservations, market information and guest behavior, hotels choose optimal controls in the form of dynamic pricing and capacity allocation in order to maximize revenue. RM is mainly associated with the airline industry and hospitality industry stated by Talluri and van Ryzin [25].

This internship contributes to the forecasting module of the RM system created by Irevenu. Irevenu, originally a software company, was founded in January 2017 by Jan Jaap van Roon, CEO of Ireckonu, prof. dr. Ger Koole, Jeroen de Korte (MSc Business Analytics student) and Rik van Leeuwen (MSc Business Analytics student). Ireckonu is a software vendor from Amsterdam that connects several software and hardware systems from hotels into one platform. JJvR is the investor behind Irevenu, and GK helps with the statistics and validation of the models. The goal of this start-up is to build an easy-to-use and affordable RM solution for the hospitality industry.

Irevenu has access to millions of records of reservation and guest data of several hospitality companies through the software of IreckonU. Due to the amount of data, extensive analysis is executed and mathematical models are validated. IreckonU and one of their customers agreed to cooperate with this internship. This customer owns 12 properties globally and have a single type of hotel room. An external dataset of competitor rates is available besides the data of IreckonU.

The remaining part of the report is structured as follows: in Section 2, the problem statement is explained. An overview of existing literature about the problem is discussed in Section 3. In Section 4, the available data, explanatory data analysis and statistical data analysis are presented. The methodology is described in Section 5. In Section 6, the results are presented. Section 7 is dedicated to the implementation of the model. Finally, Section 8 contains the conclusion and discussion.

2 Problem Statement

2.1 Background and Context

Revenue Management (RM) systems have proven to have a positive impact on revenue of hospitality companies by allowing revenue managers to carry out strategies more efficiently and effectively, according to Lee [11] and Rajopadhye et al. [20]. However, less than 10% of the industry has implemented a RM system. The other part of the industry do RM based on human gut and reports about historical data using spreadsheets. Gökşen [5] stated that the lack of technical structure or the cost of such a system are two of the main reasons why companies do not have a RM system in use.

From a mathematical point of view, a RM system consists of two main components: a demand forecasting algorithm and an optimization algorithm which determines the room rate given the demand. The interaction between these two components is key to a successful system. However, before any of these algorithms are chosen, validated and implemented, extensive data analysis needs to be executed in order to verify the assumptions of a model.

Zooming in on forecasting algorithms, there are roughly three techniques categories: time series models, the econometric approach, and other emerging methods such as AI techniques according to Song & Li [23]. For all these forecast techniques, the input is one of the crucial factors in order to obtain an accurate demand forecast. And Weatherford & Kimes [29] state that one of the key components for a successful RM system is an accurate forecast. Industry wide, companies have a database of historical bookings which is often used as input for a forecasting algorithm.

A clear distinction must be made between the input possibilities of a forecasting algorithm: bookings and demand. Bookings are defined as the records that are stored in a database and demand is the number of guests who are interested in a room. In literature, regularly these two terms are confused with each other. The number of records that are stored is an inaccurate indicator of demand because this number is influenced by rate and capacity of a property. The rate, generally set by revenue managers, is based on a combination of historical reporting spreadsheets, the number of bookings made so far (also known as on-the-books) and human experience. However, the optimal rate can only be set when there is information available of true demand and the on-the-books.

Historical bookings are seen as constrained demand due to the restriction of capacity of a property or due to various rates for the same room night, and therefore, not the appropriate input for a forecasting algorithm. Reconstructing the true demand from reservation data is called "unconstraining" in RM and is the only appropriate input form for a forecasting algorithm. Figure 1 illustrates an example of pace for a single day of a single property with a single rate for a booking horizon of 100 days. The solid line represents the true demand until the booking limit is reached. Unconstraining methods estimate the total demand that would have been observed in the absence of any booking limits (dashed part of the line).

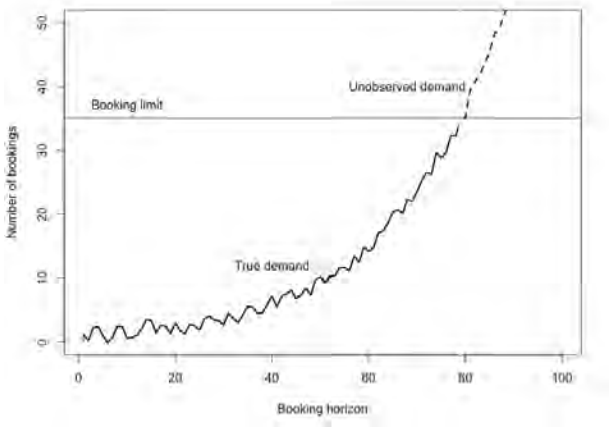


Fig. 1: Sample of pace

Weatherford & Pölt [28] show that relatively simple unconstraining methods can result in a considerable revenue increase due to a more accurate forecast, which is used as input for the optimization module of a RM system. Changing the input has advantages for the rest of a RM system. Despite these benefits, Guo, Xiao and Li [6] stated that demand unconstraining has not received as much attention as other components of a RM system.

2.2 Problem Statement

Reservation data is constrained and therefore not the appropriate input for a demand forecasting algorithm. At this moment there is no statistical unconstraining method implemented into the RM solution of Irevenu. Therefore, the goal is **to obtain unconstrained daily historical demand**, which can be used for forecasting purposes. And so, increase the accuracy of the forecasting module of this RM system. This report describes the steps from raw data until implementation applied in the hospitality industry.

2.3 Approach and Validation

The next section discusses literature about the different types of models and gives an overview of the spectrum regarding unconstraining techniques. The outcome of this review provides direction for the exploratory analysis, where data is visualized and analyzed. Several variables of both datasets are scrutinized in order to explore the dynamics of specific segments. After the exploratory analysis, statistical data analysis is executed to prove assumptions. The next step is the implementation of the chosen model and the validation of this model. After validation with simulated data, the algorithm is applied to real data.

Validation is crucial when revenue managers depend increasingly on predictions by mathematical models to justify their decisions or even let a system makes decisions. According to James et al. [9], separating data into a training set and a testing set is an important step of evaluating an implemented algorithm. Generally, the training set is around 80% of the entire set and the other 20% is used as testing set.

To reduce variability in the outcome of the model, cross validation [10] is implemented. The goal of cross validation is to test the ability to adapt to new data in order to flag overfitting or underfitting and see the performance on new datasets (test set). The data is split into 5 groups, because of the 80-20 split in training/test set, and therefore 5-fold cross validation is applied.

The RM department at the hospitality company, who agreed to cooperate with this internship, steer the revenue in such a way that it is ultimately as close to budget as possible. Industry wide, part of a budget is the revenue over a given period of time. However, the issue with such a measurement is that negative errors and positive errors cancel each other out. Which is in the advantage of revenue managers because they need to be as close as possible to the budget. From a validation perspective it is too simple and can indicate an accurate average but insufficient fit when zoomed in into single scenarios.

The performance of the model is validated by an accuracy measurement between the outcome of EM and the observations and between the outcome of EM and simulation. A detailed explanation of measurements is described in Section 5 Methodology.

2.4 Scope and Relevance

The scope of this internship is to implement and validate a demand unconstraining algorithm for hotel reservation data as stand alone. Components of the RM solution of Irevenu will not be included or used for validation. By the usage of components of Irevenu, the report could include certain preferences instead of an unbiased research. The results are relevant to researchers in the industry of RM who want to increase their forecasting accuracy, which in turn, the revenue will increase since the optimization module deals with true demand. The main focus is about room revenue, and associated costs are not included in any way.

3 Literature Review

As mentioned in the previous chapter, a Revenue Management (RM) system consists of two main pillars: a demand forecasting algorithm and an optimization algorithm. The forecasting pillar can be seen as the heart of the system. Without an accurate demand forecast, the system provides rate recommendations which can be highly inaccurate. Weatherford and Belobaba [27] states that an underestimation of 12.5% - 25% in demand can hurt the revenue by 1% - 3% on high demand days. In 1990, Lee [11] concludes that an increase of 10% in forecast accuracy can result in a revenue increase of 0.5% - 3% on high demand days. The majority of research about forecasting in RM is based on the airline industry (e.g., L'Heureux [8]; Lee [11]; Wickham [30]), however the same techniques are applied in other industries such as hospitality and retail.

The majority of these techniques use historical (observed) booking data as input. However, observed data is considered to be censored data, as mentioned in Chapter 2: Problem Statement. Weatherford and Pölt [28] report that the unconstraining process results in 2% - 12% increase in revenue. According to Guo et al. [6], unconstrained demand can contribute to the determination of allocation and booking limits. Moreover, Cooper et al. [2] prove that the underestimation of demand has a spiral down effect on the revenue. This implies that the revenue will decrease monotonically when the data remains constrained.

Guo et al. [6] categorize different strategies to unconstrain demand. Each of these strategies has advantages in specific situations. In general, a hospitality company has five possible strategies to tackle the problem of constrained demand: (1) directly observe all incoming demand, (2) leave data constrained, ignoring the fact that censorship occurs, (3) use unconstrained data only and discard censored data, (4) replace censored data using imputation methods, or (5) statistically unconstrain the data.

Strategy (1) is impossible since a hospitality company deals with multiple distribution channels through which a room night can be sold, for example Booking.com and Expedia. Monitoring all demand is not possible with these distribution channels. Zeni [31] argues that the importance of unconstraining and simply leave the data censored, results in revenue loss, which makes strategy (2) not an option. Strategy (3) discards many observations, and high demand days are important for hospitality companies because extra revenue can be achieved. Replacing censored data, strategy (4), has not been extensively researched and simple techniques are applied such as replacing by mean or median. Statistical methods, strategy (5), have proven their effectiveness in terms of revenue according to Zeni [31]: *"These models avoid the ad hoc nature of imputation methods and are built on a foundation of statistics theory. This is done at the cost of additional complexity and model assumptions that must be validated"*. With the application of statistical methods, a variety of optimization and heuristic techniques are covered that rely only on observed bookings and rate availability, addressed by Queenan et al. [17]. From the five possible strategies, the statistical strategy (5) seems to be the most promising, even though it is the most complex one of all strategies. Possible statistical options are discussed next.

Guo et al. [6] listed, in an extensive literature review, an overview of the statistical unconstraining methods, which can be divided into three categories: single-class methods, multi-class methods and multi-flight methods. The categories are based on the airline industry, however these can easily be translated to the hospitality industry. Single-class methods assume independent rate classes, which is a potentially problematic assumption, because in reality, different rate classes are dependent on each other, as explained by Talluri and Ryzin [25]. Multi-class methods and multi-flight methods do take interaction between rate classes into account. In this review, only multi-class will be discussed further because of its relevancy of the research. The multi-flight category takes into account different options to reach the same destination. The equivalent in the hospitality industry is taking another room type for the same night.

Expectation Maximization (EM) is a method, which optimize a maximum likelihood estimator, that is used in multi-class problems as well as multi-flight problems. According to Zeni [31], Weatherford [26] and Pölt [16], EM is one of the most robust statistical methods to unconstrain demand, and hence, to deal with missing data. EM was first published by Dempster et al. [3] and McLachlan and Krishnan [13] were the first with a detailed description. It has been proven that EM was successfully applied in various incomplete data problems. However, these models require knowledge of the demand distribution which can be seen as a disadvantage. This maximum likelihood estimator consists of two main steps: expectation and maximization. It is an iterative process where the parameters of the demand distribution are estimated.

Another statistical unconstraining method for multi-class problems is the Spill Model by Belobaba and Farkas [1]. This model estimates the number of persons that could not make a reservation per rate class, which is zero when the capacity is not reached (which is a limitation of the model). When the capacity is reached, the model estimates the spill by a probability distribution. No (published) implementation is found applied to the hospitality industry.

Queenan et al. [17] consider using the Double Exponential Smoothing (DES). DES uses two smoothing constants: one for smoothing the base component of the demand pattern and a second for smoothing the trend component, in order to unconstrain total demand for a point on the booking horizon. In their paper, EM, DES and not unconstraining at all are compared with each other. In two of the three datasets, DES outperformed EM and no unconstraining method. However, the assumption of independent demand is taken into account, but a method is suggested to take dependent demand into account in their paper.

Another possibility is Censored Demand Expectation-Maximization (CDEM) model introduced by McGill [12]. Demand is unconstrained by estimating untruncated distribution parameters, which is based on the EM algorithm. However a drawback of CDEM is the computational intensity. When there is a high degree of censorship, CDEM takes still around 65 iterations to to converge according to McGill. However, numerical experiments result in good estimates, even when 75% or more of the demand is censored.

Expected arrivals are underestimated or overestimated by external factors, one them is competition. Revenue managers of hospitality companies where no RM system is implemented, have difficulties with price setting due to influence of competition. Two reasons are that it is an extra variable to take into account and competition is uncertain. Every property has a competitor set, these are hotels who want to attract the same customers, also known as the segment pool. Enz and Canina [4] concluded effective pricing on low-demand days became more challenging and effective revenue management more important.

Enz and Canina [4] researched that rates of hotel rooms are more dependent on the competition than before the rise of comparison websites. There are strong positive correlations between the average daily rate and occupancy when the rate of a hotel room is just below the rates of the segment pool. Another conclusion is the weaker relationship between rate and occupancy when hotels priced substantially lower than their competitors. Aberate (2016) also concluded that rates of competition are of strong influence on the revenue management strategies of airlines. In these types of industries, where dynamic pricing is crucial, the influence of competition should be taken into account.

4 Data

Reservation data and competitor data of an international hospitality company is used. The first section of this chapter describes the available data and basic statistics in order to obtain a general overview. The second section is devoted to exploratory data analysis to obtain extra insights. The third section is about statistical data analysis and statistical tests. In that section assumptions that are made during exploratory data analysis, are verified statistically.

4.1 Available Data

Data of a single property, located in Amsterdam, The Netherlands, is selected as use case. The dataset contains 92,925 reservations, arriving from 2016-07-01 until 2018-07-01 for this property, which has a capacity of 215 rooms, and only one room type. The sum of the room nights of the reservations is equal to 200,988, which results in an average length of stay (LOS) of 2.16 nights. A reservation record contains the following information: reservation date, check-in date, check-out date, status, cancellation date, source, market code, total price, group code. The distribution of the attribute *status* is as follows: 70.86% completed stays, 26.44% cancellations and 2.70% no-shows. The average room rate per night is €146.40. The range of the attribute *rate* for a single night is bounded between €89 and €239. One downside of this dataset is the fact that if there was no sale on a certain day, the price that was set is unknown. On average, guests cancel their reservation 43.5 days before check-in, with an average length of stay 2.40 nights and the average rate per canceled night is €149.59. Finally, 7.46% of the reservations is part of a group and has a value for *group code*.

The second data source originates from OTA Insight, which contains rates that have been set by the hospitality company and their competitor set. The competitor set, containing six hotels, is established by revenue managers after market research. A competitor will be included based on several criteria, e.g. location, brand power, reviews, price/quality relationship and facilities. Due to confidentiality, competitors are numbered 1 to 6. The dataset contains over ten million records about rates per rooms type, of each competitor, from 2015-01-01 to 2017-01-01. In comparison with the reservation dataset, there are records of the rate that was set, even though there is no sales record. An analysis is executed about the behavior of the rate of the used hospitality company, and all these rates have been set manually by the revenue managers. Extra insights are gained about the behavior of the competitors. However, no information is available about the way competitors price their rooms, manually or automatically by a RM system.

Next to the reservation dataset and the dataset of OTA Insight, a dataset of demand is created based on parameters and assumptions, derived on the findings of explanatory and statistical data analysis. By applying simulated data, the statistical unconstraining of methods will be understood into more detail and it will help with the implementation, evaluation and verification of these methods. The setup of the simulated demand is explained in more detail in Chapter 5: Methodology.

4.2 Exploratory Data Analysis

By describing the datasets and presenting initial statistics, useful insights are gained. However, extra insights are necessary to create assumptions. These findings are presented below by summarizing the main characteristics per subject and help to understand the data better.

4.2.1 KPI The hospitality industry uses several Key Performance Indicators (KPI) that monitor the performance of a property and allows to compare properties with each other. KPIs that indicate the revenue performance are mainly based on the room revenue, the number of occupied beds and the capacity of a property. Next to revenue indicators, there are also indicators about costs such as energy, water consumption and cleaning, however costs is not in the scope of this internship and therefore not included.

Figure 2 shows the rate distribution in bins of €10. The average rate of a reservation is used, since individual rates per room nights are not available. For example, the average rate per room night is €100 when the total price is €300, and the reservation has a LOS of 3. The rate is rounded to the nearest €5 or €10. The rate around €140 is the most frequent one. The rates are bounded by the minimum and maximum rate, which are set up by the hospitality company itself, as mentioned in the previous section Available Data. Around $\sim 1\%$ of the reservations have a rate lower than the minimum rate, these reservations are transferred to the €90 bin. The $\sim 1\%$ increase of bin €240 with respect to bin €230 can be explained by the limitation of the maximum rate, all reservations which has a higher rate than €240 are put into the bin of €240.

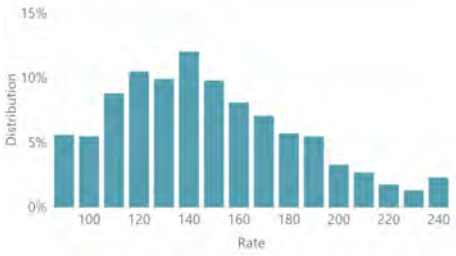
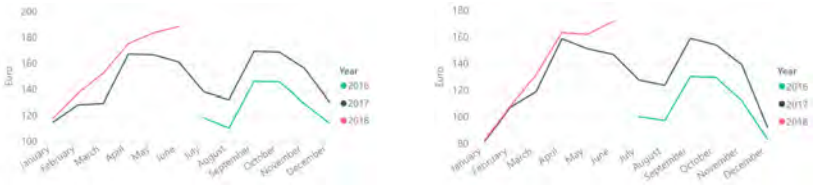


Fig. 2: Rate distribution

Important KPIs concerning price are average daily rate (ADR) and revenue per available room (RevPAR). The formula of ADR is room revenue divided by the number of occupied rooms. RevPAR is calculated by room revenue divided by the capacity. Note that the ADR is always lower or equal to RevPAR. With the available reservations data, at least every month can be compared once with the same month in the previous year. Figure 3(a) represents ADR by month and Figure 3(b) presents RevPAR by month.



(a) ADR (b) RevPar

Fig. 3: KPI by month

These metrics increase over time. The 2017 line is above 2016 line and the 2018 line is above 2017. It is desirable that these KPIs are consistent or increase over the years, otherwise this may imply that tourism is decreasing in the area or competition is taking over, since they are fishing in the same guest pool. However, this increase in ADR and RevPAR has affected the occupancy, which is the third important KPI in hospitality. The occupancy did increase in 2017, but decreases in 2018, see Figure 4.

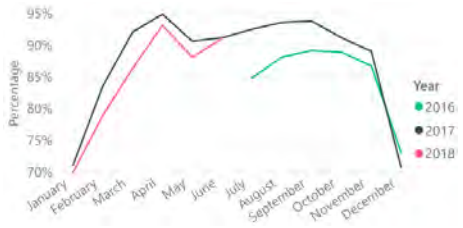


Fig. 4: Occupancy by month

A change in KPIs can have a significant impact on the total room revenue, both positive and negative. Revenue managers quickly notice if the performance of a property is off track compared to budget or forecast. Figure 5 shows the total room revenue in the same way as the previous metrics. It shows that this metric increased over the past two years which has the same seasonal pattern as ADR and RevPAR. The composition of these three metrics changed over time, higher ADR/RevPAR and lower occupancy, which have a positive impact on room revenue. However, this metric does not indicate if the net revenue increases since it does not include any costs or additional revenue. For example cleaning costs: when a new guest is assigned to a room, a more thorough cleaning job is required than when a guest is staying for second night. Room revenue also does not take extra revenue into account, for example at the bar. Since this internship is focused on unconstraining of demand, costs and additional revenue are beyond scope of the project, and the focus is entirely on room revenue.

Fig. 5: Total room revenue by month

Seasonality influences KPIs significantly, see Figure 3 and Figure 4. Other seasonality components can be derived from the arrival date, such as year, quarter and weekday. In Table 1, ADR, RevPAR and occupancy are presented by weekday. As mentioned in section Available Data, ADR is equal to €146.40 and RevPAR is equal to €126.43. The differences in ADR are smaller in comparison with the differences in RevPAR, which is influenced by occupancy: when the difference between ADR and RevPAR is higher, the occupancy is lower. For example, the difference between Friday and Sunday is equal to €10.93 in terms of ADR and for RevPAR this difference is equal to €20.30. Seasonality components month and weekday have impact on KPIs and so on demand. Seasonality should therefore be considered as an influence that should be taken into account.

Metric	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
ADR	145.01	148.10	146.98	146.02	149.13	149.21	138.20
RevPAR	122.12	134.50	134.67	128.64	130.01	136.74	98.47
Occupancy	84.21%	90.81%	91.63%	88.10%	87.18%	91.58%	71.25%

Table 1: KPIs by weekday

4.2.2 Source The source of a reservation is a interesting variable to explore. Table 2 presents an overview of the four sources with several statistics. The four sources are Online Travel Agencies (OTA), e.g. Expedia.com or Booking.com, website (WEB), direct (DIR) and Global Distribution System (GDS), which are cooperate bookings. Channel WEB contains reservations made via the hotel website. The reservations from direct channel are guest who called or emailed the hospitality company. One of the reasons to contact via phone or email is to make a group booking, because guests are, for example, not able to make a reservations for 10 rooms at once online. The desire is to keep the share of OTA reservations as low as possible since a commission is paid to the OTA. These commissions can be as high as 40% of the total price a guest paid, which depends on and/or influences the rank of the hospitality company on the OTA website.

The OTA channel is highly represented with 57.72%. Another disadvantage, besides the high commissions, is the relatively high cancellation rate of 37.51%. The is no information available regarding the commission the hospitality company paid. Even though the ADR of OTA is higher than WEB, the net profit is higher of a reservation made via WEB since the commission needs to be subtracted from the ADR of OTA. Guests who book via direct seem to be the most profitable since the cancellation rate is the lowest, ADR the highest and LOS the longest.

Source	**% grand total**	**Cancellations**	**ADR**	**LOS**
OTA	57.72%	37.51%	146.69	2.33
WEB	23.92%	20.77%	143.78	1.88
DIR	9.18%	8.05%	161.08	2.45
GDS	9.18%	21.51%	147.33	1.85

Table 2: Sources and statistics

4.2.3 Lead Time Several guest types can be identified by a combination of variables. For example, business guests usually book via GDS channels. One of these key metrics is the number of days a guest books the reservation in advance, which is the difference in days between the reservation date and check-in date. This metric is also known as lead time or booking pace, the average over all observations of this metric is equal to 49.8 days.

The number of reservations made per lead time is displayed in Figure 6. This figure is in line with the thoughts of revenue managers, who argue that more guests make a reservation when check-in date becomes closer. In this graph, canceled reservations and no-shows are included. Guests can make a reservation a year in advance. Around 15% of the reservations are made with a lead time between 100 and 365.

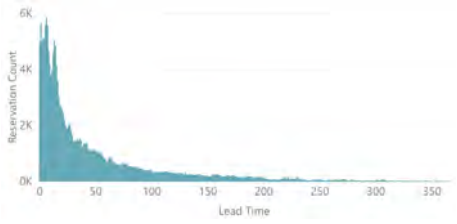
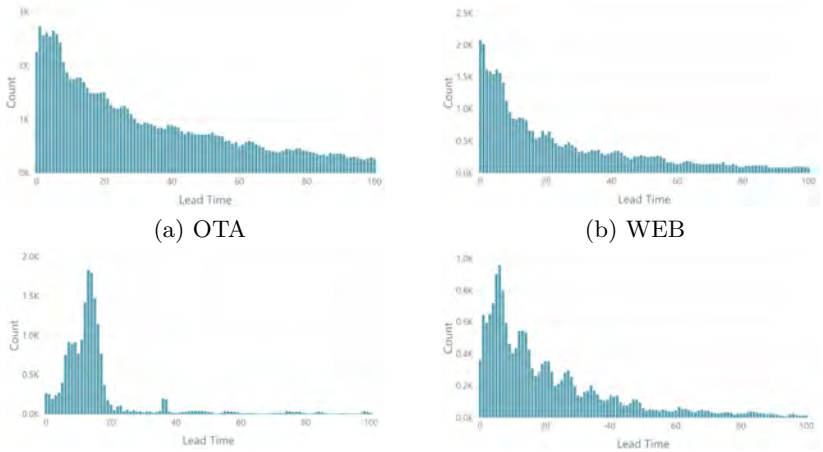


Fig. 6: Count of reservations by lead time

Figure 6 is split into the four channels, see Figure 7. The maximum lead time of 100 is chosen to emphasize the difference between the channels. The average lead time for OTA, WEB, DIR and GDS is respectively 59.4, 48.3, 16.6, and 26.1 days. The shape of Figure 7(c) (DIR) is completely different compared to the other channels. This can be explained because DIR reservations are mostly made via phone, and include group bookings since it is not possible to make a group reservation via the website.



(a) OTA (b) WEB

(c) DIR (d) GDS

Fig. 7: Count of reservations by lead time per channel

The other three channels, Figure 7(a), Figure 7(b) and Figure 7(d), have a similar shape, few observations for high lead time and more observations for lower lead time. However, there is a more pronounced weekly pattern in Figure 7(d) GDS compared to the other two sources. This can be explained because business travelers book during weekdays instead of weekend days. There is also a increase in Figure 7(b) WEB the last two days before check-in date. An explanation can be that guests want to make sure that there is availability at the hotel and therefore make a reservation via the hotel's own website instead of an OTA, such as Booking.com or Expedia.

Figure 8 presents the average room rate per lead time, where lead time is grouped by week. The rate is derived from the reservations dataset. Interesting to see is when the lowest rate is paid. For this specific property, the lowest average rate is paid the week before check-in date. The pattern of this graph is seen as disloyal to guests who book way in advance. The guests who are willing to make their reservations way in advance should pay less than a guest who book a few days in advance. Note that the y-axis starts at 120 to emphasize the differences between the weeks.

The rate for a room night differs over time for a specific check-in date, and that can be seen when the average price per lead time is calculated. Guests are able to make a reservation one year in advance. However, since there are little to no bookings for large lead times, only the last three months before check-in are taken into account. The lead time is grouped per week, see Figure 8. Note that the y-axis starts at 120 to emphasize the differences between the weeks. Around two months in advance, there is a price increase which decreases when the lead time also decreases. There is a spike two weeks before check-in which may be influenced by the type of guest who make their reservation typically two weeks in advance. When looking at Figure 7(c) and 2, these seem to be the guests that book through DIR.

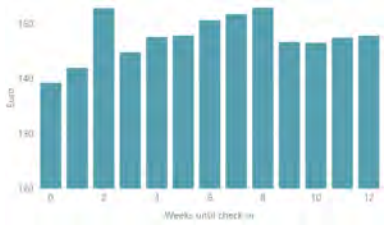


Fig. 8: Average price per lead time grouped by week

In Table 3, the average lead time by weekday and source is presented. There is a difference of roughly 10 days lead time between weekdays, Monday to Thursday, and weekend days, Friday to Sunday. As mentioned before, channel has an impact on rate and length of stay, and this also holds for lead time. The reason why the lead time is higher during weekdays can be explained by the type of guest. During the weekends more leisure guests stay in the hotel, and leisure guests tend to plan their trip more in advance than guests who travel for business. Guest who come in through OTAs make their reservation further in advance than guests that come in through the website. This can be explained by the type of guests who book via this channel.

Source/ Weekday	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**	**Total**
OTA	58.6	57.4	53.5	54.4	61.1	62.7	64.33	**59.4**
WEB	45.0	45.8	43.3	44.9	52.5	57.3	51.7	**48.3**
DIR	13.6	13.6	16.6	17.8	22.4	19.6	15.2	**16.6**
GDS	25.3	25.8	25.0	24.2	31.2	32.0	29.5	**26.1**
Total	**45.6**	**44.1**	**42.0**	**44.8**	**55.5**	**59.0**	**56.3**	**49.8**

Table 3: Lead time by weekday and source

4.2.4 Competitors After looking into the reservations data, the next subsection is devoted to the other data source that is available: OTA Insight. Song and Wong [24] mentioned that possessing competitive advantages could be key to success. Data analysis can be used to gain competitive advantages. Song and Wong [24] also mentioned that competition is still not precisely defined. Nevertheless, competitiveness is obviously seen as an external factor to investigate. As mentioned in a previous section, the dataset contains over 10 million records that only contain rates that have been set on OTA channels. This may cause deviation in rates compared to early mentioned averages which are derived from the reservation dataset.

Each competitor has multiple room types, some of which are not comparable, and rates for different LOS. Therefore, a selection is made per competitor to determine which rate is suitable for comparison. LOS has to be equal to 1 because competitors may have an automated discount if a guest stays multiple nights. The maximum lead time is set on 90 days, because from that point in time rates are actively stored. The best available rate is selected. This basic selection resulted into a dataset containing over ∼2.6 million rows. In Table 4, an overview per competitor is given, whereas Brand is the hotel used for this research.

Competitor	**% grand total**	**Average rate**	**Room types**
Brand	1.99%	123.87	1
1	25.25%	160.52	5
2	6.87%	145.71	3
3	24.68%	171.54	5
4	17.11%	151.25	2
5	11.98%	189.59	2
6	12.10%	129.25	3

Table 4: Competitors and statistics

Hospitality companies switch rates more often when the check-in date comes closer, because competition becomes heavier and every competitor is fighting for the same last minute guest. It is interesting to know who is a leader and who is a follower. This is explored in the next subsection. A single historical day is highlighted in Figure 9, using a lead time of 90 days. This figure shows constantly changing rates of each competitor, increasing as well as decreasing. This is a typical pattern that occurs here, first increase significantly and a couple days later, decrease until even lower than the initial rate.

In Figure 10, the average prices over the last 30 days are shown per competitor. This figure gives insight in what competitors do when the check-in date becomes closer. The rank between the competitors may depend on the number of sales, this assumption is tested in the next subsection. In general, hospitality companies decrease their price on average when the check-in date comes closer, and increase it just a few days before the check-in.

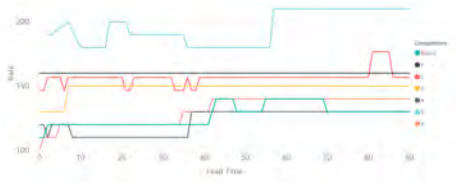


Fig. 9: Single day of rates in competitor set

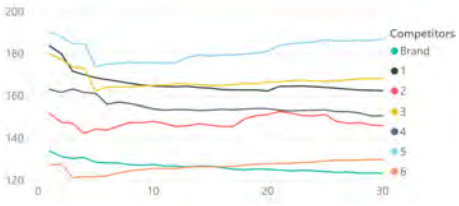


Fig. 10: Average rate by lead time for each competitor

The reservation dataset only contains rate records when there was an arrival. But with the OTA Insight dataset, the behavior of the Brand can be analyzed. Only the days are analyzed which have complete rate information over the final 30 days until check-in. It is not likely that the rate for a room night remains constant in the last 30 days until check-in due to the heavy competition these days. Therefore, days are considered where the rate did change at least once.

The rate changes on average 4.7 times in the last 30 days until check-in, with a variance of 10.2. This indicates that the spread of the number of rate changes is high. Of these 4.7 times of rate change, 2.4 were positive and 2.3 were negative. This indicates that there is no real pricing strategy active since the rate changes is as much up as going down.

In Figure 11, the percentage of rate change is given by lead time. It shows that on lead time 29, 8.10% of the time the rate has been changed. Lead time 1 shows that 24.31% of the time the rate is changed. RM activity increases when the check-in date is coming. When no RM strategy is implemented, it is getting in the last potential guests who search for a room night.

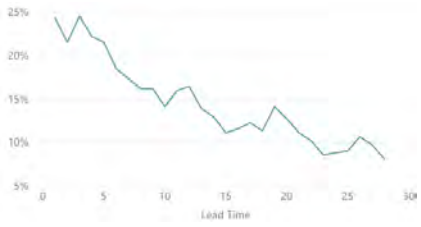


Fig. 11: Percentage of rate change over lead time

4.3 Statistical Data Analysis

From the subsection Explanatory Data Analysis, valuable insights are gained and multiple assumptions where made. In this section, assumptions are validated with statistical evidence. Certain selections of the datasets are made. These selections are explicitly mentioned before validation of assumptions. The term lead time is often used in the previous sections, however from this point on the term booking horizon will only be used, defined as t from $1, ..., T$ where T represents the check-in date.

4.3.1 Rank Dependency By combining the reservations dataset and the OTA Insight dataset, a general assumption is validated among the hospitality industry: when a property is ranked higher on an OTA website, it leads to more sales. This might be true since a guest is searching using specific criteria such as neighborhood and public transport accessibility, and the next logical criteria is room rate. Rank among a competitor set is based on the daily rate that was set on a point on the booking horizon for a certain date. The assumption regarding more sales with higher is tested in this subsection. This is valuable since high commissions, up to 40% of the room rate, are paid for a high rank.

The first step in this validation process is data selection, because not all rates of each point on the booking horizon are stored in the OTA Insight dataset. As mentioned in the previous section Exploratory Data Analysis, rates are actively stored 30 days before check-in. Therefore, rates are considered 30 days before check-in. Looking at this specific selection of the OTA Insight dataset, it still occurs that no rate is stored for a room type. Then this room type is not taken into account. It may occur that two competitors have an equal rate, than the rank is set equally.

A small ranking example is given in Table 5 among three hospitality brands. It is possible that other brands occur based on the search criteria, however these are not observed and therefore the assumption is made that the OTA Insight data is the complete dataset. Regarding the number of sales, only the reservations are considered which came through an OTA distribution channel. Since there is no complete overlap between the two datasets, the rank and reservations are taken into account between 2016-01-01 and 2016-07-01. In the end, this selection led to ∼11 thousand room nights. Only the rows of the hospitality brand are taken into account because the reservation database only contains sales from the hospitality brand itself. The booking horizon and rate are considered as variables which influence the number of sales. Booking horizon is taken into account as categorical value because booking horizon of 1 is not better or worse than booking horizon 2.

A two-way ANOVA test is used to validate the assumption that there is significant difference in sales. This statistical test is chosen because it compares levels of two or more factors for mean differences on a single continuous response variable. The mathematical model is defined as $Sales \sim Rank + BookingHorizon$, where $Sales$ is an numerical explanatory variable and $Rank$ and $BookingHorizon$ is considered as categorical response variables.

Hotel	Date	Booking Horizon	Rate	Rank
1	2018-01-01	1	100	1
2	2018-01-01	1	150	2
3	2018-01-01	1	200	3
1	2018-01-01	2	200	2
2	2018-01-01	2	150	1
3	2018-01-01	2	200	2
1	2018-01-01	3	250	3
2	2018-01-01	3	100	1
3	2018-01-01	3	200	2

Table 5: Ranking example between three hotels

However, booking patterns per weekday are different because more bookings are made on Monday through Friday ($\sim$ 16% of total reservations per day) compared to Saturday and Sunday ($\sim$ 8% of total reservations per day). For example, booking horizon 29 (1 day before check-in) on a Sunday has less sales because Saturday on a booking horizon of 29. Therefore, a two-way ANOVA test is performed per weekday since booking horizon is a response variable.

The outcome of the test does reject the null hypothesis for each of the weekdays because all p-values regarding *Rank* and *BookingHorizon* are below the significance level of 0.05, see Table 6. The F-statistic is the ratio of mean squares, which implies the population variance that takes into account the degrees of freedom. If this number is close to 1 it is likely that the null hypothesis is not rejected. The F-statistic is greater than 1 in all cases, which confirms (again) the rejection of the null hypotheses.

Weekday	**Booking Horizon**		**Rank**	
	F-statistic	Rank	F-statistic	Rank
Monday	2.339	$1.15e^{-04}$	3.751	$3.31e^{-07}$
Tuesday	6.355	$2e^{-16}$	4.278	$2.74e^{-07}$
Wednesday	1.404	$9.18e^{-03}$	2.832	$3.48e^{-04}$
Thursday	1.985	$1.71e^{-03}$	2.631	$9.74e^{-04}$
Friday	5.196	$2.62e^{-16}$	6.769	$1.72e^{-14}$
Saturday	6.366	$2e^{-16}$	6.346	$2.35e^{-11}$
Sunday	6.199	$2e^{-16}$	4.781	$5.53e^{-08}$

Table 6: One-way ANOVA test about dependency of *Rank* and *BookingHorizon*

4.3.2 Dynamic Time Warping The results of the Exploratory Data Analysis pointed out that competitors change their rates continuously, which influences RM strategies as mentioned in Section 3 Literature Review. Schwartz et al. [21] researched the challenges of including competition into a forecasting module and concluded that an implementation using competition is difficult but not impossible. Their results were promising and could potentially increase the forecast accuracy. A first step of understanding the competition is to identify if there are leaders and followers in the market. The rate changes can be seen as movements or actions, and such a set of movements can be analyzed. Silva et al. [22] pointed out movements are analyzed in all sorts of industries, from medicine to astronomy to sensors, all in order to gain extra insights.

By adapting Dynamic Time Warping (DTW), these leaders can be detected in a set of competitors. DTW is a time series analysis technique which identifies similarities between two sequences of values by calculating DTW distance between values over time. In this context, the sequence has a daily time dimension and the values are the rates. The lowest available rate per day is chosen as value. The sequences of competitors are extracted from the dataset of OTA Insight. No papers are found of the application of DTW in the hospitality industry, however, research exists about implementation in the financial market, which has a similar setup and goal as this subsection.

The definition of a time series is a sequence that is equally spaced in time (t) with ordered values (n) such that $x = (x_1, x_2, ..., x_n)$ and $x \in R$ for any $t \in [1, n]$. DTW algorithm computes a non-linear alignment between two time series values, defined as x and y. A dynamic programming algorithm is applied in order to calculate the DTW distances. Equation (1) calculates $dtw(x, y)$ where $dtw(n, n)$ represents the total DTW distance. And $c(x_i, y_i)$ represents the cost, defined as the squared Euclidean distance between them, of matching two observations. The lower the $dtw(n, n)$ value is, the more similar two time series are.

$$dtw(i,j) = c(x_i, y_i) + min \begin{cases} dtw(i-1,j) \\ dtw(i,j-1) \\ dtw(i-1,j-1) \end{cases} \tag{1}$$

The time complexity is $O(n^2)$ because the two time series have length n and all DTW distances between time points are calculated in the matrix. The length is set on 30 days until check-in date. In order to reduce the complexity, the warping constraint is added. This constraint limits the time difference that the algorithm is allowed to use match the observations. Now the algorithm calculates DTW distances closer to its main diagonal instead of all values. Ratanamahatana et al. [19] argues that a warping parameter of 10% is sufficient choice for nearest neighbor classification. In this setup, this results in a range of six days: three days in advance and three days in history.

The DTW distance between time series is calculated between the competitors set, including the used hospitality company. Because there is also interest to discover if competitors follow other competitors. Competitors have different room types, and to reduce the number of calculations a single room type per

competitor is chosen based on the closest average rate. Because the DTW distance between competitor 1 and competitor 2 is equal to the distance between competitor 2 and competitor 1, only the right triangular matrix is calculated without the diagonal, since it does not add any value if DTW distance is calculated between the same time series. Competitor data between 2015-01-01 and 2016-01-01 is selected. One criterion in order to calculate the DTW distance, is that there should at least be two price changes during the booking horizon per time series. Otherwise, if two competitors did not change their price, these time series have DTW distance of zero. Only the 90% best DTW distances are selected because in some of the days it is not realistic to watch the competitor set.

The median DTW distance over the selected date period is given in Table 7, as mentioned only the right triangular matrix is filled. The variance of DTW distance over the selected date period is given in Table 8, rounded as integer since the numbers are distinctive enough. Based on the median and variance, three combinations are close to each other: brand - 1, brand - 6 and 5 - 6. Based on the median, these combinations are respectively ranked as 3, 2, 1. However based on the variance, these combinations are respectively ranked as 1, 2, 3.

Competitor	Brand	1	2	3	4	5	6
Brand	-	92.2	141.3	99.0	113.6	221.3	91.8
1	-	-	144.2	87.5	135.9	179.9	90.9
2	-	-	-	148.6	161.4	233.7	123.7
3	-	-	-	-	116.4	216.7	91.9
4	-	-	-	-	-	270.2	98.2
5	-	-	-	-	-	-	218.3
6	-	-	-	-	-	-	-

Table 7: Median DTW distance between competitors

Competitor	Brand	1	2	3	4	5	6
Brand	-	1303	6012	2316	3175	13913	1345
1	-	-	4219	1698	4981	8023	2285
2	-	-	-	7803	8644	16774	5850
3	-	-	-	-	4227	10102	2034
4	-	-	-	-	-	14023	2865
5	-	-	-	-	-	-	9358
6	-	-	-	-	-	-	-

Table 8: Variance DTW distance between competitors

The DTW algorithm does not define who is a leader and who is a follower, the outcome is the DTW distance. Therefore, the paths of between brand - 1 and brand - 6 are analyzed visually. A single example between brand - 1 can be found in Figure 18 in the Appendix. This particular figure visualizes the result of the algorithm with a distance of 12.2. There is no specific leader between brand - 1 taking into account all of the figures. Whereas between brand - 6, brand can be seen as the leader among the two. The DTW distance, as well as the variance of competitor 5, is way higher than other competitors. This may imply that this competitor is not suitable in the competitor set regarding price. However, due to its location it should be included into the competitor set.

5 Methodology

In the Literature Review, several unconstraining methods are described and considered. Based on robustness and the number of papers found, Expectation Maximization is the most suitable unconstraining method. The algorithm can be expanded into a multi-flight category, implying multiple room types in hospitality, which is one of the major advantages. This might be in the interest for potential future clients of Irevenu. Several papers describe the algorithm applied in the airline industry, but also in the hospitality industry. Other methods were also considered, such as Gaussian process regression by Price et al. [14], however these methods are novel methods and no prove is presented that these methods work for real data.

The rest of this chapter is divided into six sections: the first section introduces the concept of choice sets. The second section discusses the unconstraining algorithm and the third is about the parameters that are estimated. The fourth section is devoted to validation and the fifth is about the booking horizon. The last section describes the setup of simulated data and the results of it. Recreating algorithms and methods used in researches can be challenging to implement, therefore source code of EM can be found in the Appendix.

5.1 Choice sets

As shown in the Exploratory Data Analysis, rates are influenced by several components such as weekday and source. By setting a different rate, guests are willing or not willing to accept that rate. However, everyone is willing to pay less for the same product. For example, when a guest paid 139, it was also willing to pay 129 and so on until the minimum rate. This implies an order in preference, meaning a guest prefers 129 above 139.

A general demand rate is not accurate enough since guests act differently when a different rate plan is set, seen in Exploratory Data Analysis. Therefore, it is assumed that arrival rates per rate class are required. Haensel and Koole [7] discussed the concept of choice sets, which is defined as the sets of substitutable products or choice alternatives with a strict preference order. This is the case with rates for hotel rooms since every guest prefers the best available rate or has a personal maximum rate for a single room night.

A small example to illustrate the concept of choice sets. Imagine a property with a room that has a view to the ocean. There are two possible rates for this room, say A and say B. Possible choice sets are: {A}, {B}, {A,B} and {B,A}; C will denote the set of all choice sets. Choice sets are written with a decreasing preference order from left to right. Therefore, the choice set {A,B} states that customers being represented by this choice set are strictly preferring A over B.

This concept is transformed to rates for a room night for this specific hospitality company, see Table 11 in the Appendix. The rate of a room night is rounded to the nearest 9. If the rate is not adjusted, there are not enough data points per rate bucket and too many choice sets. There are 16 steps of 10 between 89 and 239 and therefore 16 choice sets, defined A to P. The number of

guests decrease when the rate increase since guests have a maximum amount they are willing to spend. This implies that choice sets influence the equation for the arrival rate. Equation (2) represents a non-decreasing demand rate which includes choice sets, as described in Haensel and Koole [7].

$$\lambda_c(t) = \beta_c \cdot e^{\alpha_c \cdot t} \tag{2}$$

5.2 Expectation Maximization

In order to overcome the problem of constrained demand, Expectation Maximization (EM) with interaction between choice sets will be applied. This algorithm is an iterative method that finds, with the use of maximum likelihood, estimations of parameters. Haensel and Koole (2011b) described the setup and pros and cons.

The previous subsection, Choice sets, introduced a function for the demand rate. The demand rate per choice set follows an exponential curve, see Equation (3), for the booking horizon t, from $1, ..., T$. Parameters α and β will be estimated and both should be positive. Reservations data is used as input, which are the observable sales. Unfortunately, not all information is available of which choice sets were open during the booking horizon. Therefore, this information is extracted from the observed sales data.

$$\lambda_c(t) = \beta_c \cdot e^{\alpha_c \cdot t} \tag{3}$$

As mentioned in the Literature Review, there are two main steps that need to be executed each iteration i, $i = 1, ...,$, until the stopping criteria is reached to obtain the optimum. In each iteration, new α and β parameters will be estimated separately for all choice sets c, denoted as α_c^i and β_c^i. The new estimated parameters are updated and the E-step can be executed again. Equation (4) represents the M-step for each choice set c for iteration i. In words, this equation represents the maximum likelihood function that estimates, for each choice set c, parameters α and β, which are obtained by minimizing the negative log-likelihood function, see Equation (4).

$$(\alpha_c^i, \beta_c^i) = \arg\min_{\alpha,\beta>0} - \boldsymbol{L}_c(i) \tag{4}$$

The log-likelihood function at iteration i for choice set c is described by Equation (5), known as the E-step. Note that these functions need to be created and optimized separately for each choice set c. The variable $S(t, f)$ denotes the observed sales in class f at time t.

$$\boldsymbol{L}_c(i) = \sum_{t=1}^{T} \log P_c^i(S(t, f)) \tag{5}$$

The probability of a sale is defined by Equation (6), which takes into account interaction between choice sets where $X \sim \text{Poisson}(\lambda_c^i(t) = \beta_c^i \cdot e^{a_c^i \cdot t})$. The $\lceil \text{x} \rceil$ operator returns the closest integer greater than or equal to x. The parameter

$\lambda^{i-1}_{overlap}(c,t)$ sums the estimated rates from the previous iteration for which the preferred available classes are included in choice set c. The indicator function returns whether the preferred available class f in choice set c at time t was open.

$$P^i_c(S(t,f) = \begin{cases} P\left[X = \lceil \frac{\lambda^{i-1}_c(t)}{\lambda^{i-1}_{overlap}(c,t)} \cdot S(t,f) \rceil\right] \cdot \mathbb{1}_{\{U(c,t)=f\}} & \text{, if } f > 0 \\ 1 & \text{, otherwise} \end{cases} \tag{6}$$

Furthermore, there are two possible stopping criteria. Either reaching the maximum number of iterations, which has been set to 100. Or numerical convergence, which is when the difference in both parameters between two iterations for all choice sets is smaller than 10^{-6}. Numerical tests indicate if the maximum number of iterations is sufficient, since numerical convergence is preferred.

The initialization step is introduced in order to obtain initial estimates, $\lambda^0_c(t)$. This step does not take interaction between choice sets into account. The difference between Equation (6) and the initial probability is the faction with the λ's. Therefore, the iteration loop starts at $i = 1$. Pseudo code of EM algorithm for unconstraining demand per choice set:

Initialization step:

For all $c \in C$

$(\alpha^i_c, \beta^i_c) = \arg\min_{\alpha,\beta>0} -\sum_{t=1}^{T} \log \begin{cases} P(X_t = S(t, U(c,t)) & \text{, if } U(c,t) > 0 \\ 1 & \text{, otherwise} \end{cases}$

where $X_t \sim \text{Poisson}(\lambda_c(t) = \beta_c \cdot e^{a_c \cdot t})$

end

Iteration loop: $i = 1, ...$

$E - step$:

For all $c \in C$

For all $t = 1, ..., T$

$$P^i_c(S(t,f) = \begin{cases} P\left[X = \lceil \frac{\lambda^{i-1}_c(t)}{\lambda^{i-1}_{overlap}(c,t)} \cdot S(t,f) \rceil\right] \cdot \mathbb{1}_{\{U(c,t)=f\}} & \text{, if } f > 0 \\ 1 & \text{, otherwise} \end{cases}$$

where $X =$Poisson$(\lambda^i_c(t) = \beta^i_c \cdot e^{\alpha^i_c \cdot t})$

end

$\boldsymbol{L}_c(i) = \sum_{t=1}^{T} \log P^i_c(S(t,f)$

end

$M - step$:

For all $c \in C$

$(\alpha^i_c, \beta^i_c) = \arg\min_{\alpha,\beta>0} - \boldsymbol{L}_c(i)$

end

Until stopping criteria reached.

5.3 Parameters

This section is devoted to the parameters of the non-decreasing demand function. The parameters of this function are estimated using the EM algorithm as described in the previous section. The goal is to obtain unconstrained daily historical demand, so the parameters for each choice set are estimated for each single day of the historical demand. One of the disadvantages of estimating parameters per day for each choice set is the lack of data points for higher choice sets. This becomes a problem when there is only a single sale for a high choice set registered. The algorithm fits an α and β parameter for a choice set where there is only a single sale involved.

Each parameter of this non-decreasing function has their own influence on the exponential line. The α parameters determines the steepness of the line and can be interpreted as the behavior, the β determines the height. Therefore, an α and β parameter per weekday per source per choice set is estimated so the weekday and source influence is captured, now referred as $\alpha_c^{source,weekday}$ and $\beta_c^{source,weekday}$. This seasonality factor is chosen because the guests arriving on a weekday have a different booking pattern over the booking horizon. Guests with business traveling purposes stay mostly on Monday through Thursday, whereas leisure guests mostly stay in the weekends when there is not a specific holiday. The section Exploratory Data Analysis shows sufficient proof of the source and weekday influence in the performance of the hotel.

These parameters are estimated over multiple demand scenarios. For example, there could be 100 historical demand scenarios for a Tuesday, and so all sales with source equal to WEB on Tuesdays are used to estimate the parameters for all of the choice sets regarding weekday Tuesday and source WEB. These demand scenarios are limited by a maximum booking horizon. The method to choose the optimal booking limit is described in the subsection Booking Horizon of this chapter.

5.4 Validation and Accuracy

Once Expectation Maximization (EM) is implemented, validation is required in order to confirm the correctness of the implementation. Demand is simulated according to known α_c and β_c parameters with a straightforward revenue management strategy. The correctness is tested by estimation of the known parameters. This setup is described in the subsection Simulation.

The variable a represents the observation (actual) and the variable f represents the outcome of EM (fit). The observations are 20% of the total dataset because the outcome of the algorithm is based on 80% of the dataset. This is repeated 5 times, because 5-fold cross validation is applied in order to reduce the variability of the results.

The difference between the observation and fit does not imply error, it represents the unpredictable part of observations. An accuracy measurement is applied which is able to deal with zero values as actuals, because the number of observed sales can be equal to zero. Accuracy measurements such as Average

Percentage Error (APE) and Mean Average Percentage Error (MAPE) cannot deal with zero values as actuals due to the fact that the actuals are in the denominator. Because the numbers are low-volume, a single value may influence the measurement significantly. Therefore, the Weighted Average Percentage Error (WAPE), see Equation (7), is chosen as an appropriate measurement. The difference between actual and fit at t of the booking horizon is labeled as the deviation. The WAPE is less sensitive to large distortions in comparison with MAPE for example.

$$WAPE = \frac{\sum_{t=1}^{n} |a_t - f_t|}{\sum_{t=1}^{n} a_t} \cdot 100 \tag{7}$$

The WAPE indicates the performance for a given day between the actuals and the fit. There is also interest to see the performance or deviation over the entire booking horizon. Therefore, the Mean Absolute Deviation (MAD), see Equation (8), is also calculated over the booking horizon. For example, the MAD for $t = 63$ is the average of absolute deviation over all days. This accuracy measurement is able to handle zero values.

$$MAD = \frac{\sum_{t=1}^{n} |a_t - f_t|}{n} \tag{8}$$

5.5 Booking Horizon

Selecting an alternate booking horizon could have a significant influence on the performance of the unconstraining algorithm due to the number of data points, which decreases when T of the booking horizon increases. One year in advance, $T = 365$, a guest has the possibility to book a room. However, the number of reservations that ware made between 300 and 365 days in advance is less than 1% of the total number of reservations. No papers are found on selecting the optimal booking horizon.

In order to select the optimal booking horizon, a successful implementation of the unconstraining method is required. The next step is to test several ranges of booking horizons and measure the performance of the unconstraining algorithm. By selecting one day less in a loop, every booking horizon is tested. The performance of the unconstraining method is described in the previous subsection Validation.

5.6 Simulation

In this section, the setup of the simulated data is described and unconstrained with the EM algorithm. Introducing the following situation where a single room night is sold for four possible rates: 50, 100, 150 and 200. The corresponding overlapping choice sets are as follows: $Set_1 = \{50\}$, $Set_2 = \{50, 100\}$, $Set_3 = \{50, 100, 150\}$ and $Set_4 = \{50, 100, 150, 200\}$. The capacity of this setup is 100 rooms. With the assumption of arrivals according to an Inhomogeneous Poisson process. 100 different demand scenarios are simulated over a booking

horizon from $1, ..., 56$, which represents a booking window of 8 weeks. The initial parameters of the simulated demand scenarios are presented in Table 12 in the Appendix. The total dataset will be divided into 5 parts of 20 demand scenarios each, due to the implementation of 5 fold-cross validation.

With simulation, a certain strategy is required to open and close choice sets in order to observe demand for a specific choice set. A strategy is booking limits, when a certain number of sales is reached, the choice set will be closed. The booking limits are defined as 50 for Set_1, 85 for Set_2, 95 for Set_3 and 100 for Set_4. This implies that there are overlapping booking limits. When Set_4 is closed, the capacity of this setup is reached and no more room nights can be sold. Results obtained differ when different booking limits are chosen. There are several other strategies that could also be applied, for example is on a specific point of the booking horizon a choice set will be closed. This simulation should be close as the strategy of the revenue managers of the hospitality company.

The first measure that is described in subsection Validation is the Weighted Average Percentage Error (WAPE). The mean WAPE for the simulated sales are presented in the first column of Table 9 per fold, the shows column shows the variance. This means that the WAPE per fold is an average over 20 demand scenarios. Across the five folds, the numbers do not differ much, except fold 4 is higher, caused by an outlier since the variance is also the highest.

Fold	**WAPE**	**Variance**
1	65.89	15.92
2	65.52	24.25
3	65.21	21.86
4	67.60	23.03
5	65.71	20.78

Table 9: WAPE and variance based on simulated sales

The other measure that is described in subsection Validation is the Mean Absolute Deviation (MAD). Figure 12 presents a boxplot per t on the booking horizon. Because there are in total 15-20 outliers, these are not plotted in the figure. All of these outliers are placed above the boxplots. Overall, the average MAD, which is the green line inside of the boxplot, stays between .5 and 1.5. The fluctuations in the boxplots are a consequence of the revenue management strategy implementation described earlier. For example, the average MAD increases in from $t = 0$ until $t = 20$. After that moment, the average MAD jumps up, and decreases until $t = 30$. The rest of the booking horizon varies more in comparison with the beginning of the booking horizon in terms of average MAD. The length of the box represents the middle half of the values. The length of the boxes does not vary much over time. The whiskers of the boxplot indicate the variability. The upper whiskers of the boxplot are overall longer than the bottom whiskers, which indicates more variability towards a higher MAD.

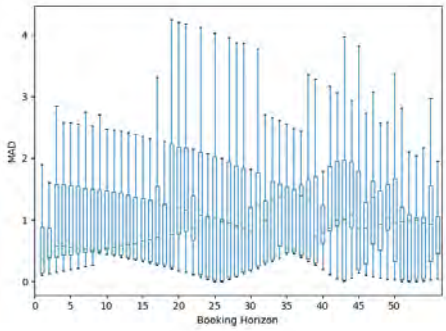


Fig. 12: Boxplot of MAD per t of booking horizon

The parameters of fold 1 are selected and plotted against the initial parameters in order to examine one of the results, see Figure 13. One of the first observations is the fit of Set_2, the estimate underestimates the beginning of the booking horizon and overestimates the end of the booking horizon. Taking a closer look at when Set_2 is open/closed, it is not such a bad fit starting from $t \sim 30$. Around $t \sim 55$, Set_2 is closed again. The estimated curve of Set_1 is overestimated from the start of the booking horizon. The estimate of Set_3 is accurate, however a small underestimation occurs near the end of the booking horizon. Regarding Set_4, the sales that belong to the highest choice set are not significant. The curve does increase at the end, however in the beginning of the booking horizon (0-30) no sales are observed. Therefore, it is hard to estimate whether guests are willing to accept that rate.

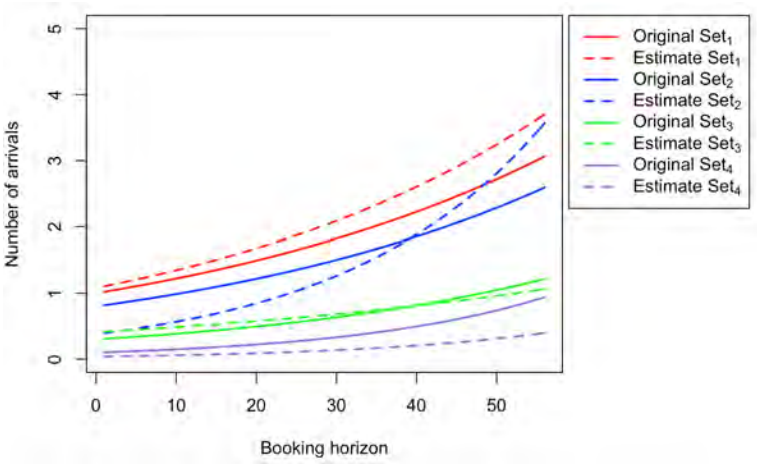


Fig. 13: Example of fit with parameters generated by fold 1

6 Results

This chapter presents the results which are generated by applying the estimation model described in the previous chapter. All of the results are obtained using the reservations dataset. The same approach is used as with the simulated data. The reservation dataset is split by weekday and source, as described in subsection Parameters in the chapter Methodology.

6.1 Single Demand Scenario

In order to obtain a complete overview and understanding of how the results are computed, a single demand scenario is highlighted. In total 100 demand scenarios are used to obtain this result, 99 scenarios as training and 1 scenario as test. Monday is chosen as weekday and OTA is chosen as channel. The code of Expectation Maximization module can be found in Source Code 1 of the Appendix.

The dots in Figure 14 represent the number of sales on time t on the booking horizon. The color represent a specific choice set. This particular scenario contains sales in choice sets B till H, so a rate range of €99 till €159. There are zero sales when there is no dot present on time t of the booking horizon. The rate changes quite often during the entire booking horizon. Compared to other demand scenarios, Figure 14 is not an exceptional case and is in line with the findings in Exploratory Data Analysis regarding rate fluctuations.

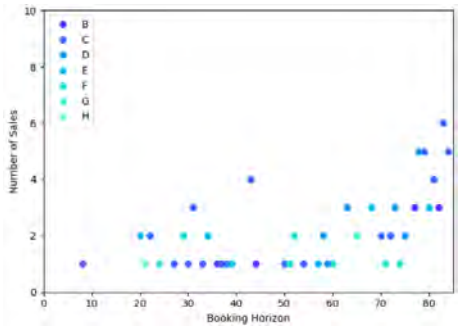


Fig. 14: Single demand scenario of a Monday and OTA

Figure 15 shows the same demand scenario as Figure 14, however the blue line is added which represents the unconstrained demand rate. The demand rate is according to the choice sets which are open on time t. The parameters for the demand rate per choice set is a result of the fit over 99 different demand scenarios. For each scenario, only the parameters are used for the choice sets that were open.

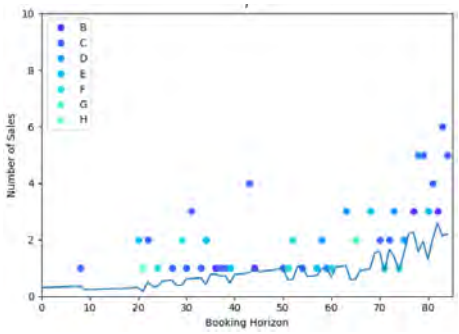


Fig. 15: Single demand scenario of a Monday and OTA with result EM

The WAPE between the observed sales and demand rate is equal to 81.81. As mentioned in the methodology, simulated sales are required to compare the WAPE of the observed sales. The WAPE between the simulated sales and demand rate is 80.05, which is the median of 1000 simulations. This situation can be labelled as almost perfect since the difference between WAPE values is low.

6.2 Cross Validation

The same steps, as described in Single Demand Scenario, are applied with cross validation. There are 100 different demand scenarios selected for each weekday, even though there are $\sim$ 103 scenarios available. For each fold of cross-validation, 80 scenarios are used for training and 20 scenarios are used for testing. As mentioned in the previous section Single Demand Scenario, the maximum booking horizon is equal to 84 days. The reason for setting the booking horizon equal to 84 is because the hospitality company actively starts with revenue management from this moment in time.

The difference between WAPE of sale and demand scenario and WAPE of simulation and demand scenario per weekday per source is displayed in Table 10. The WAPE of the simulation is subtracted from the WAPE of the sale since the interest is in how close the WAPE of the sale is to the WAPE of the simulation.

		Weekday						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Source	OTA	19.42	15.34	23.66	20.71	17.16	20.11	25.65
	WEB	18.09	10.03	8.95	16.77	31.42	36.06	43.59
	DIR	1326.29	680.42	559.99	658.92	732.29	812.00	910.52
	GDS	21.31	8.79	8.31	17.28	31.23	37.29	42.75

Table 10: Difference between WAPE of sale and WAPE of simulation

Source DIR has, by far, the largest difference between WAPE of the four sources. The results of the Exploratory Data Analysis already indicated that source DIR differs significantly from the other three sources. Therefore, unconstraining demand could be challenging since this channel contains phone calls and walk-ins, which is not a non decreasing function over the booking horizon, see Figure 7(c).

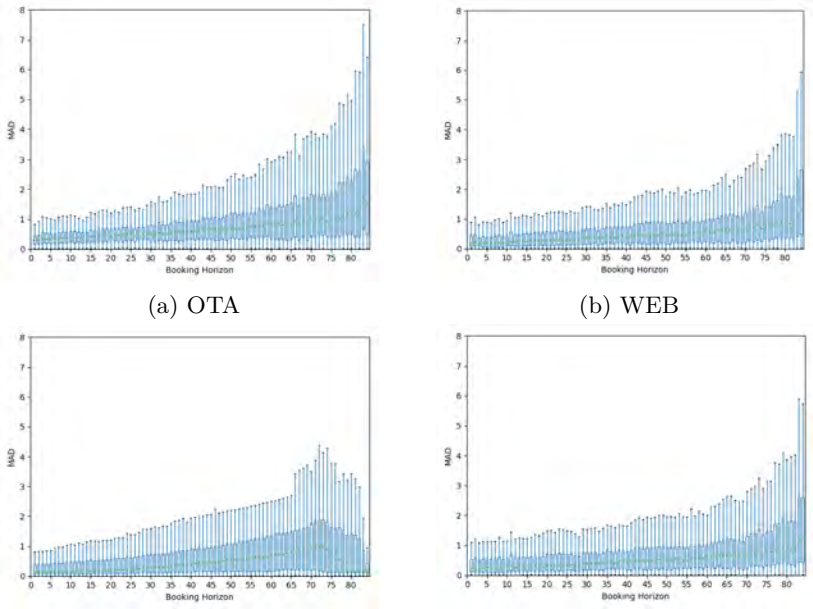
The other thee sources, OTA, WEB and GDS, the results differ from roughly 8 until 43. Source OTA has the smallest deviation among all weekdays and source WEB has the largest deviation, when DIR is not taken into account. Overall the lowest difference is on Tuesday and Wednesday and Sunday has the largest difference. The WAPE of sale and demand scenario can be found in Table 13 in the Appendix. The WAPE of simulation and demand scenario can be found in Table 14 in the Appendix.

The variance of WAPE between sale and demand rate and simulation and demand rate for each combination of weekday and source can be found in Table 15 and Table 16 in the Appendix. If DIR is not taken into account, in $\sim$85% of the time the variance in WAPE of the sale is lower than the variance in WAPE of the simulation Which implies that the results of the unconstraining method is more stable than the results generated by the simulation.

All folds of the cross validation for the each combination of weekday and source terminated due to numerical convergence. In total, Expectation Maximization module successfully converged 140 times. The average number of iterations before numerical convergence for OTA, WEB, DIR and GDS is respectively 7.8, 15.4, 11.8 and 9.8. The average among weekdays is taken because the difference between weekdays was between 1 and 7 iterations. The maximum number of iterations is set on 100.

Figure 16 contains four graphs which represent each of the channels. Each individual graph presents a boxplot of the Median Absolute Deviation (MAD) for each t on the booking horizon. Figure 16 (a), (b) and (d) have similar pattern, which is increasing over the booking horizon. The graph of DIR, 16(c), has a partially similar pattern, however the boxplots are decreasing near the end of the booking horizon, just like the reservations through that channel. All four graphs have a similar pattern as the number of reservations made over the booking horizon, see Figure 7. The outliers are not taken into account in Figure 16. However, the outliers of DIR were in line with the high WAPE which is presented in Table 13 in the Appendix. The whiskers of the boxplot indicate the variability. The upper whiskers of the boxplot are overall longer than the bottom whiskers, which indicates that the there is more variability towards a higher MAD.

For each channel, some of the extreme differences (> 250) in WAPE values are analyzed in order to find an explanation. The specific dates were extracted for these demand scenarios. For some of these dates, little to no guests came in for example. Because of this low demand, the difference increases extremely compared to the simulation generates demand. This low demand can be explained by events which took place, for example public holidays. These events lead to demand which is not visible in the current selected booking horizon.



(a) OTA (b) WEB

(c) DIR (d) GDs

Fig. 16: MAD per channel aggregated over weekdays

6.3 Booking Horizon

Determining the optimal booking horizon is executed as described in section Methodology. For each combination of weekday and source, a range of maximum booking horizon is tested. The range of the booking horizon is between 14 days (2 weeks) and 84 days (12 weeks). The completion for a single weekday and source took almost 24 hours with a single laptop. To illustrate the possible outcome, a single combination is chosen which is weekday Monday and source OTA.

Figure 19 in the appendix illustrates the outcome the range of booking horizon from 70 until 84, which represents a booking horizon of 10 weeks until 12 weeks. The red line is the WAPE between sale and demand rate and the blue line represents the WAPE between simulation and demand rate. Both lines increase in terms of WAPE when a longer booking horizon is chosen. This is due to the increase in deviation, which is higher than the number of sales for a higher maximum of booking horizon. For example Figure 15, the deviation between the sales and demand rate is larger than the number of sales made in range 0 until 20.

The gap between the two lines increases because the WAPE between sales and demand rate increases more than the WAPE between simulation and demand rate. The simulation is more accurate for longer booking ranges due to the fact that the simulation makes less mistakes in the early stages of the booking horizon.

7 System Development Life Cycle

Mathematical problems are, mostly, described in discrete structures and tested with simulated data where access to source code is limited. When there is no access to source code and with the assumption that the reader knows how to translate these structures into code, implementation can be challenging. Due to the lack of explanation regarding this translation, outsiders are frustrated and therefore there is a possibility that there is no willingness to implement a mathematical model, even though it could be a solution to their problem.

A system development life cycle (SDLC) of a Revenue Management system (RM) creates guidelines to a successful implementation. According to Radack [18], a SDLC consists of several steps: analysis, design, development and testing, implementation, documentation, and maintenance. In order to add an extra extension/module to an existing system, the same steps can be applied. Regarding the unconstraining module, the steps analysis until testing are already covered in this report in previous chapters. Therefore the focus of this chapter is on implementation, documentation, and maintenance.

7.1 Overview

As introduced in the Problem Statement chapter, a RM system contains of two main pillars; forecasting and optimization. Figure 17 shows an overview of a RM system in a schematic way. Several other modules could be added in order to complete the overview of a RM system. However, the focus of this figure is on the placement of the unconstraining module. For now, the input is the reservations database, and the outcome of the entire RM system is ideally a rate recommendation. Other sources could be added to increase the performance of the system, event or competitor data for example. Event data may contain information such as the date and the scale of the event.

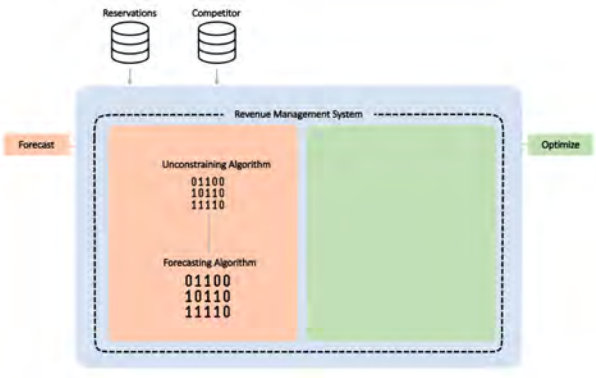


Fig. 17: Schematic overview of placement unconstraining module in RM system

7.2 Implementation

The main goal of implementation ensures that the system is operational. This implies two things: the RM system is still operational and that the module is up and running as a stand alone. The unconstraining module should easily be turned on or off the see the effects and impact of the implementation. Therefore, the the module is structured as stand alone and a single command executes the entire module. Unconstraining reservations data on a daily basis is an ongoing process which needs to be executed every day. Because it is desirable that the foresting algorithm contains the latest data available.

Because the unconstraining module needs to be triggered every day to unconstrain the latest demand, the module can be implemented into an app service. Regarding the implementation of Irevenu, Microsoft Azure is used as app service. This app service has an API which will be triggered according to the desired updating scheme of the end user. This app has two main API's, one which will unconstrain reservations data and the other which will update the parameters of the choice sets.

When the algorithm estimates the new parameters for the choice sets based on the newest demand scenarios, these results need to be saved into a database. When a new demand scenario occurred, these parameters can easily be accessed to unconstrain that specific scenario. Therefore, a database is also part of the implementation and needs to be integrated. Access of the API to the database is therefore crucial. By giving access to a database, security becomes a point of interest, which is discussed further in the subsection Maintenance.

7.3 Documentation

Documentation contributes to the continuity of the module. One can image that the module is maintained by a team which consists of different roles. Another benefit is that documentation provides the ability to onboard new members to the team faster.

Documentation provides the information which is needed to run the system, for example explanation about the required data fields in order to run the module successfully. There are four data fields required in order to unconstrain data: date, booking horizon, type and rate. The type is referred as the product that the guests bought, for example a standard room or suite, since this hotel has one room type, it is left out. The combination of rate and type defines the choice sets.

A part of documentation is data selection from the reservation database. The selected dataset could contain statuses such as stayed, canceled r no-show. Please note that these reservations statuses are based on the available dataset. If there is more selected than stayed reservations, a module should compensate for the initial selection on a later point of the system. For example, a cancellation predictor could be implemented if canceled reservations are taken into account in the unconstraining process. During this internship, only stayed reservations are selected as input. The last two years of data is sufficient to obtain parameters.

7.4 Maintenance

The last step of the SDLC is maintaining and evaluating the module, which should not be underestimated. The completion of the life cycle is essential to quickly solve and adapt to changes that may occur in the future. This step also includes a review of the module from time to time.

A suitable approach to maintain the source code of this module is version control. Version control tracks and provides control over changes to source code. A team of people can concurrently make changes to the same files. For each update, a release note is attached with a description of the most important changes which contributes to documentation. Regarding the implementation of Irevenu, Git is used as version control system. There is a backup of the source code available with version control. When the performance of the new released module is decreased, relative to the previous version of the module, a rollback can be executed and little to no harm is done. For example, when errors occur with saving the new estimated parameters to the database.

Security is also a part of maintenance, and therefore an ongoing process, which can be seen as important as the actual development itself. Security is a point of interest in almost each step of the SDLC. Only people and systems from the organization are provided with access to make changes or make improvements to the system, also known as role based access control. Since there is no interest in third parties making use of the module. Each person has restricted access to parts where they are responsible for. Restricted access by IP address contributes to the security level and therefore necessary to make sure that no third parties are able to access the module.

The evaluation step of the life cycle identifies whether the system meets the initial requirements and objectives. Therefore, every step of the SDLC should have their own objective(s). If one of the requirements is not met, one could take a step back and make sure it is implemented eventually.

8 Conclusion

This chapter is devoted to the findings of this research, and describes the process of from data analysis until implementation with the goal of obtaining unconstrained daily historical demand.

The outcome of the Literature review combined with Exploratory Data Analysis resulted in an implementation of Expectation Maximization (EM) with a non-decreasing demand function (exponential form) depending on three variables: rate, weekday and source. These three variables are chosen based on a general assumption is that guests have a maximum rate they accept. The difference in behaviour is shown among weekdays and the possible booking sources found in the data.

Both for the hotel occupancy and the average paid room rate, the numbers differ significantly among weekdays. The occupancy ranges from almost 92% on average on Saturdays to around 71% on Sundays. Furthermore, the average room rate differs over €10, from €138 on to €149 on Fridays and Saturday. The differences in average room rate between the booking sources are significant as well. Channel DIR has the highest rate of €161, while the other three channels are €15 to €18 lower.

The implementation of EM is validated in a controlled environment where data is simulated based on an Inhomogeneous Poisson process where parameters are known with four choice set. The hospitality company has rate range, which is divided in buckets of 10 which results in 16 choice set since each bucket represents a choice set. Due to the chosen depending variables, this results in estimating 2240 parameters (16 choice sets, 5 fold cross validation, 7 weekdays and 4 channels). The WAPE of the EM implementation is, after five fold cross validation, ~ 65, which is the lowest result possible given the exponential form, with a variance of ~ 21.

The results of the unconstraining methods is a comparison between sales and simulated sales which are generated by the demand rate found by EM. The purpose of simulating sales from the demand rate shows the minimal deviation. The difference in WAPE between the sales and the WAPE of simulation is 20.3 for OTA, 23.6 for WEB and 23.8 for GDS, which are similar, which is positive. Only for the channel DIR, which was an outlier in the data analysis already, which has a difference of over 800. This can be explained by the fact that group bookings are a part of this channel and these do not fit an exponential curve.

The overall conclusion from the internship is that the implementation and usage of EM in the setting of hospitality revenue management are valid methods. Although the results are not perfect, they are promising enough to support the further exploration and research of this method. Further discussion of the methods used and a review of potential improvements can be found in the next section, Discussion.

9 Discussion

In this chapter a number of extensions of this research are discussed.

One of the main improvements could be scaling the values of the β parameters by a seasonal factor. The outcome of Exploratory Data Analysis showed that KPI differs over months, which indicates that the demand is dependent on seasonality. Including such a seasonal factor may increase the accuracy of the current model which can potentially be used for forecasting purposes, since the parameters for each choice set are estimated per weekday and source.

No external influences are taken into account such as nearby events. This will definitely have influence on the demand for these specific days. These dates are announced in advance, and this can even by outside of the predefined booking horizon. Therefore, an deep dive into events could provide clarification on how a hospitality company should handle their events: to handle these days manually, to handle automatically or a combination of these two.

No optimal booking horizon is found due to the lack of computational power. For each combination of weekday and source, it took over 24 hours to estimate parameters for 16 choice sets. An extensive analysis can be done regarding this subject since no papers could be found about this topic. One should take into account the current revenue strategy of the hospitality company due to the success rate of implementation.

The accuracy or error measurement in RM is not uniform across literature. During this research, the Weighted Average Percentage Error (WAPE) and Median Absolute Deviation (MAD) are used as measurements due to the occurrence of zero values. However, Haensel and Koole [7] used different measurements for example, even though the same model is implemented. This leads to results which can not be interpreted in the same way. Other results could be achieved if different measurement were applied. Eventually, since this topic is in the industry of revenue management, all methods should be measured in revenue, at least that is a suggestion.

During this research, an exponential curve is chosen as demand rate. However, this is an assumption which could be investigated. Different functions can be applicable, for example logarithmic functions or a square root function. Even different shape over the entire booking horizon. For example, for early stages of the booking horizon, a linear curve could be sufficient and for later stages an exponential curve. Simulation of arrivals must be adapted if another shapes are tested, since Inhomogeneous Poisson process could be used due to the exponential curve.

The influence of competition on demand unconstraining is an influence which is shortly discussed in Exploratory Data Analysis and Statistical Data Analysis by adapting Dynamic Time Warping. One of the major issues regarding competition is data availability. The majority of hospitality companies do not store their rate for each day, and even less hospitality companies regularly save rates of their competitors.

10 Appendix

10.1 Tables

choice set	Rates included
A	89
B	89, 99
C	89, 99, 109
D	89, 99, 109, 119
E	89, 99, 109, 119, 129
F	89, 99, 109, 119, 129, 139
G	89, 99, 109, 119, 129, 139, 149
H	89, 99, 109, 119, 129, 139, 149, 159
I	89, 99, 109, 119, 129, 139, 149, 159, 169
J	89, 99, 109, 119, 129, 139, 149, 159, 169, 179
K	89, 99, 109, 119, 129, 139, 149, 159, 169, 179, 189
L	89, 99, 109, 119, 129, 139, 149, 159, 169, 179, 189, 199
M	89, 99, 109, 119, 129, 139, 149, 159, 169, 179, 189, 199, 209
N	89, 99, 109, 119, 129, 139, 149, 159, 169, 179, 189, 199, 209, 219
O	89, 99, 109, 119, 129, 139, 149, 159, 169, 179, 189, 199, 209, 219, 229
P	89, 99, 109, 119, 129, 139, 149, 159, 169, 179, 189, 199, 209, 219, 229, 239

Table 11: choice sets

		Choice set			
	Parameters	Set_1	Set_2	Set_3	Set_4
Initial	α_c	.020	.021	.025	.040
	β_c	1.00	.80	.30	.10
Fold 1	α_c	.022	.040	.017	.041
	β_c	1.08	.38	.41	.04
Fold 2	α_c	.022	.041	.018	.041
	β_c	1.10	.37	.40	.04
Fold 3	α_c	.023	.040	.018	.041
	β_c	1.05	.37	.40	.04
Fold 4	α_c	.023	.040	.018	.042
	β_c	1.05	.37	.40	.04
Fold 5	α_c	.021	.040	.018	.042
	β_c	1.09	.37	.40	.04

Table 12: Simulation parameters - inital and estimated

		Weekday						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Source	OTA	108.72	105.69	109.01	103.81	96.21	92.92	111.97
	WEB	117.63	113.47	111.47	117.06	136.76	142.19	145.47
	DIR	1405.71	763.71	650.28	755.76	852.36	920.60	1026.33
	GDS	97.07	113.47	111.47	117.06	136.76	142.19	145.47

Table 13: WAPE between sale and demand rate

		Weekday						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Source	OTA	89.3	90.35	85.35	83.10	79.05	72.81	86.32
	WEB	99.54	103.44	102.52	100.29	105.34	106.13	101.88
	DIR	79.42	83.29	90.29	96.84	120.07	108.60	115.81
	GDS	75.76	104.68	103.16	99.78	105.53	104.90	102.72

Table 14: WAPE between simulation and demand rate

		Weekday						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Source	OTA	5.08	3.15	5.32	5.57	3.62	3.45	8.25
	WEB	3.81	2.86	2.61	4.71	14.33	8.50	10.42
	DIR	59235.64	10730.16	5031.45	7405.07	13970.11	21713.81	22018.09
	GDS	3.24	2.42	2.81	5.01	13.79	7.31	11.22

Table 15: Variance of WAPE between sale and demand rate

		Weekday						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Source	OTA	7.53	11.15	13.51	11.19	6.80	4.35	4.75
	WEB	8.30	12.45	13.51	11.52	9.20	9.83	7.48
	DIR	8.01	6.90	8.00	7.34	9.10	12.95	10.18
	GDS	7.19	12.29	12.50	11.96	8.82	9.60	7.71

Table 16: Variance of WAPE between simulation and demand rate

10.2 Figures

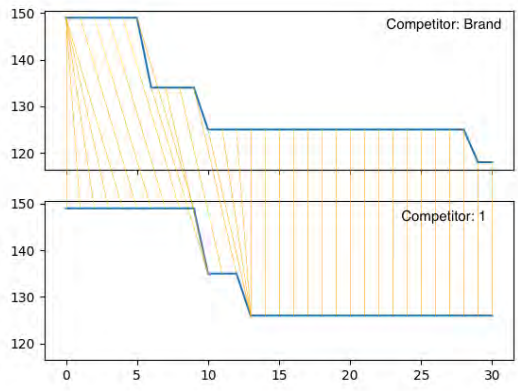


Fig. 18: An example of DTW algorithm

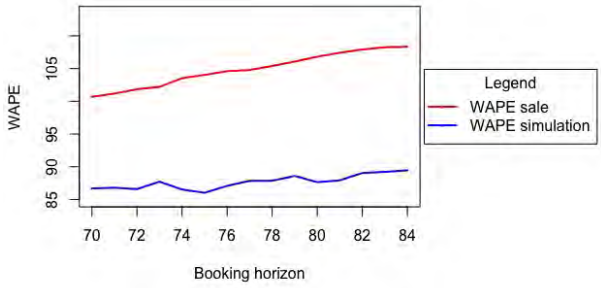


Fig. 19: WAPE over multiple maximum booking horizons

10.3 Source Code

```
import numpy as np
import scipy.optimize as optimize
import scipy
import sys


class ExpectationMaximisation:

  def __init__(self, choice_set, booking_horizon, db_sales):
      # Assign input to variables
      self.c_s = choice_set
      self.b_h = booking_horizon
      self.db = db_sales
      self.dates = np.empty(len(np.unique(self.db[:, 0])),
                      dtype=object)

      # Create frames for algorithm
      self.u_cs = np.empty(len(np.unique(self.db[:, 0])),
                      dtype=object)
      self.s_o_cs = np.empty(len(np.unique(self.db[:, 0])),
                      dtype=object)
      self.alpha = np.zeros(len(self.c_s))
      self.beta = np.zeros(len(self.c_s))

      # Define stopping criteria
      self.iter_max: int = 50
      self.diff_tol: float = 1e-06
      self.stopping_criteria_reached = None

  def initialization(self):
      # Set minimum and maximum rate
      self.db[:, 2] = np.clip(self.db[:, 2], np.min(self.c_s),
                      np.max(self.c_s))

      # Create u_cs and s_o_cs frames for each day in given dataset
      for idx, date in enumerate(np.unique(self.db[:, 0])):
          self.dates[idx] = date
          date_sales = self.db[np.where(self.db[:, 0] == date), 1::][0]
          u_cs_temp = np.zeros((len(self.c_s), len(self.b_h)))
          s_o_cs_temp = np.zeros((len(self.c_s), len(self.b_h)))

          for sale in date_sales:
              u_cs_temp[np.where(self.c_s == sale[1])[0][0]::,
```

```
                                        sale[0]] = 1
            if np.where(self.c_s == sale[1])[0] == 0:
                s_o_cs_temp[np.where(self.c_s == sale[1])[0][0],
                                        sale[0]] += 1
            else:
                s_o_cs_temp[0:(np.where(self.c_s == sale[1])[0][0]
                                        + 1), sale[0]] += 1
        self.u_cs[idx] = u_cs_temp
        self.s_o_cs[idx] = s_o_cs_temp

def algorithm_init(self):
    for c in np.arange(0, len(self.c_s)):
        #  Define log-likelihood function
        def log_likelihood_init(params):
            return_value = 0
            lambda_t = params[1] * np.exp(params[0] * self.b_h)
            for day in np.arange(0, len(self.u_cs)):
                p_array = np.ones(len(self.b_h))
                for t in self.b_h:
                    if self.u_cs[day][c, t] > 0:
                        p_array[t] = max(sys.float_info.min,
                                (np.exp(-lambda_t[t]) *
                                min(sys.float_info.max,
                                    np.power(lambda_t[t],
                                        self.s_o_cs[day][c, t])) /
                                        scipy.special.factorial(
                                                self.s_o_cs[day][c, t]))
                return_value += -np.sum(np.log(p_array))
            return return_value

        # Execute optimization
        result = optimize.minimize(log_likelihood_init,
                                   np.array([.01, .01]),
                                   bounds=[(.01, .01), (.01, .01)])

        # Save initialization parameters
        self.alpha[c] = result.x[0]
        self.beta[c] = result.x[1]

def algorithm(self):
    self.algorithm_init()

    # Iteration loop
    for i in np.arange(0, self.iter_max):
        log_likelihood_array = []
```

```
89
90          # E-step:
91          for c in np.arange(0, len(self.c_s)):
92              lambda_c = self.beta[c] * np.exp(self.alpha[c] * self.b_h)
93              lambda_overlap = np.zeros(len(self.b_h))
94              for c_j in np.arange(c, len(self.c_s)):
95                  lambda_overlap += self.beta[c_j] * np.exp(self.alpha[c_j] *
96                                                            self.b_h)
97
98              # Define log-likelihood function
99              def create_log_likelihood_i(c):
100                 def log_likelihood_i(params_i):
101                     return_value = 0
102                     lambda_t = params_i[1] * np.exp(params_i[0] *
103                                                     self.b_h)
104                     for day in np.arange(0, len(self.u_cs)):
105                         x = np.ceil(lambda_c / lambda_overlap *
106                                     self.s_o_cs[day][c, :])
107                         u_c = self.u_cs[day][c, :]
108                         p_array = np.ones(len(self.b_h))
109                         for t in self.b_h:
110                             if u_c[t] > 0:
111                                 p_array[t] = max(sys.float_info.min,
112                                 (np.exp(-lambda_t[t]) *
113                                   np.power(lambda_t[t], x[t])) /
114                                   scipy.special.factorial(x[t]))
115                         return_value += -np.sum(np.log(p_array))
116                     return return_value
117
118                 return log_likelihood_i
119
120             log_likelihood_array.append(create_log_likelihood_i(c))
121
122         alpha_iteration = np.zeros(len(self.c_s))
123         beta_iteration = np.zeros(len(self.c_s))
124
125         # M-step:
126         for c in np.arange(0, len(self.c_s)):
127             # Execute optimization
128             result = optimize.minimize(log_likelihood_array[c],
129                     np.array([self.alpha[c], self.beta[c]]),
130                             bounds=[(0.001, np.inf), (0.001, np.inf)])
131
132             # Save results
133             alpha_iteration[c] = result.x[0]
```

```
134                 beta_iteration[c] = result.x[1]
135
136             # Check stopping criteria
137             if all(abs(np.divide(alpha_iteration - self.alpha, self.alpha)) <
138                            self.diff_tol):
139                 self.stopping_criteria_reached = i
140                 break
141             else:
142                 self.alpha = alpha_iteration
143                 self.beta = beta_iteration
```

Source Code 1: Expectation Maximization in Python

References

1. Belobaba, P. P., Farkas, A.: Yield management impacts on airline spill estimation, Transportation Science, 1999.
2. Cooper, W. L., Homem-de-Mello, T., Kleywegt, A. J.: Models of the Spiral-Down Effect in Revenue Management, Operations Research, 2006.
3. Dempster, A. P., Laird, N. M., Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society B„ 1997.
4. Enz, C. A., Canina, L.: An examination of revenue management in relation to hotels' pricing strategies, Cornell Hospitality Report, 2005.
5. Gökşen, S.: Implementing Revenue Management, Vrije Universiteit Amsterdam 2011.
6. Guo, P., Xiao, B., Li1, J.: Unconstraining Methods in Revenue Management Systems: Research Overview and Prospects, Advances in Operations Research, 2012.
7. Haensel, A., Koole, G.: Estimating unconstrained demand rate functions using customer choice sets, Journal of Revenue and Pricing Management, 2011b.
8. L'Heureux, E: A new twist in forecasting short-term passenger pickup, In Proceedings of the 26th Annual AGIFORS Symposium, 1986.
9. James, G.: An Introduction to Statistical Learning: with Applications in R, Springer, 2003.
10. Kohavi, R.: A Study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection, International Joint Conference on Articial Intelligence, 1995.
11. Lee, A. O.: Airline reservations forecasting : probabilistic and statistical models of the booking process, Flight Transportation Laboratory Report R90-5, 1990.
12. McGill, J. I.: Censored regression analysis of multiclass passenger demand data subject to joint capacity constraints, Annals of Operations Research, 1995.
13. McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions, 1997.
14. Price, I., Fowkesa, J., Hopman, D.: Gaussian Processes for Demand Unconstraining, 2017.
15. Pölt, S.: Forecasting is difficult - especially if it refers to the future. Reservations and Yield Management Study Group Annual Meeting Proceedings, 1998.
16. Pölt, S.: From bookings to demand: the process of unconstraining, Proceedings of the AGIFORS Reservations and Yield Management Study Group, 2000.
17. Queenan, C. C., Ferguson, M., Higbie, J., Kapoor, R.: A comparison of unconstraining methods to improve revenue management systems, Production and Operations Management, 2007.
18. Radack, S.: The system development life cycle (SDLC), National Institute of Standards and Technology, 2009
19. Ratanamahatana, C. A., Rostamizadeh, A., Talwalkar, A.: Three Myths about Dynamic Time Warping Data Mining, International Conference on Data Mining, 2005.
20. Rajopadhye, M., Ghalia, M. B., Wang, P. P.: Forecasting uncertain hotel room demand. Information Sciences, 2001.
21. Schwartz, Z., Uysal, M., Webb, T., Altin, M.: Hotel daily occupancy forecasting with competitive sets: a recursive algorithm. International Journal of Contemporary Hospitality Management, 2016.
22. Silva, D. F., Batista, G. E. A. P. A., Keogh, E.: On the effect of endpoints on dynamic time warping. 2016
23. Song, H., Li, G.: Tourism Demand Modelling and Forecasting âĂŤ A Review of Recent Research, Tourism Management, 2008.

24. Song, H., Wong, K.: Tourism and hotel competitiveness research, Journal of Travel & Tourism Marketing, 2009.
25. Talluri, K. T., van Ryzin, G. J.: The Theory and Practice of Revenue Management, Kluwer Academic Publishers, 2004.
26. Weatherford, L. R. : Unconstraining methods, Proceedings of the AGIFORS Reservations and Yield Management Study Group, 2000.
27. Weatherford, L.R., Belobaba, P. P.: Revenue Impacts of Fare Input and Demand Forecast Accuracy in Airline Yield Management, Journal of the Operational Research Society, 2002.
28. Weatherford, L.R., Pölt, S.: Better Unconstraining of Airline Demand Data in Revenue Management Systems for Improved Forecast Accuracy and Greater Revenues, Journal of Revenue and Pricing Management, 2002.
29. Weatherford, L.R., Kimes, S. E.: A Comparison of Forecasting Methods for Hotel Revenue Management, Tourism Management, 2003.
30. Wickham, R. R.: Evaluation of forecasting techniques for short-term demand of air transportation, MIT Thesis: Flight Transportation Lab, 1995.
31. Zeni, R. H.: Improved forecast accuracy in revenue management by unconstraining demand estimates from censored data, Rutgers University, 2001.